PUBLICATIONS OF
THE UNIVERSITY OF EASTERN FINLAND

Dissertations in Forestry and Natural Sciences

UNIVERSITY OF EASTERN FINLAND

FRIDAY JOSEPH AGBO

Co-designing a smart learning environment to facilitate computational thinking education in the Nigerian context

Co-designing a smart learning environment to facilitate computational thinking education in the Nigerian context

Friday Joseph Agbo

Co-designing a smart learning environment to facilitate computational thinking education in the Nigerian context

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences
No 464

University of Eastern Finland
Joensuu
2022

Academic dissertation

To be presented by permission of the Faculty of Science and Forestry for public examination in the Auditorium M100 in the Metria Building at the University of Eastern Finland, Joensuu, on May 10, 2022, at 13:00.

PunaMusta oy
Joensuu, 2022
Editors: Pertti Pasanen, Matti Tedre, Jukka Tuomela, and Matti Vornanen
Sales: University of Eastern Finland Library
ISBN: 978-952-61-4468-9 (print)
ISBN: 978-952-61-4469-6 (PDF)
ISSNL: 1798-5668
ISSN: 1798-5668
ISSN: 1798-5676 (PDF)

Author's address: Friday Joseph Agbo
University of Eastern Finland
School of Computing
P.O. Box 111
80101 JOENSUU, FINLAND
email: friday.agbo@uef.fi

Supervisors: Adjunct Professor Solomon Sunday Oyelere, Ph.D.
Luleå University of Technology
Depart. of Computer Science, Elect. & Space Eng.
Forskargatan 1
SE-93187 Skellefteå, SWEDEN
email: solomon.oyelere@ltu.se

Research Manager Jarkko Suhonen, Docent, Ph.D.
University of Eastern Finland
School of Computing
P.O. Box 111
80101 JOENSUU, FINLAND
email: jarkko.suhonen@uef.fi

Professor Markku Tukiainen, Ph.D.
University of Eastern Finland
School of Computing
P.O. Box 111
80101 JOENSUU, FINLAND
email: markku.tukiainen@uef.fi

Reviewers: Professor Francisco José García-Peñalvo, Ph.D
University of Salamanca
Depart. of Computer Science and Automatics
Paseo de Canajelas 169 37008 Salamanca, SPAIN
email: fgarcia@usal.es

Professor Leila Goosen, Ph.D
University of South Africa
Depart. for Science and Technology Education
Pretoria 0003, SOUTH AFRICA
email: goosel@unisa.ac.za

Opponent: Professor Erik Barendsen
Radboud University
Institute for Science Education
Postbus 9010
6500 GL NIJMEGEN, The Netherlands
email: erik.barendsen@ru.nl

Agbo, Friday Joseph
Co-designing a smart learning environment to facilitate computational thinking education in the Nigerian context.
Joensuu: University of Eastern Finland, 2022
Publications of the University of Eastern Finland
Dissertation in Forestry and Natural Sciences; 464
ISBN: 978-952-61-4468-9 (print)
ISSNL: 1798-5668
ISSN: 1798-5668
ISBN: 978-952-61-4469-6 (PDF)
ISSN: 1798-5676 (PDF)

ABSTRACT

This study examined how to co-design a student-centered smart learning environment to facilitate computational thinking education in the context of a Nigerian higher education institution. Smart learning is a new learning approach where learner-centered pedagogy and advanced technology play major roles in education. Computational thinking is a requisite for students in the digital age, not only because it provides the necessary skills for solving many contextual problems daily but also because it is crucial for preparing young learners for future career challenges. In particular, for computer science novices in Nigerian universities who have not been previously exposed to programming, computational thinking knowledge such as problem decomposition, abstraction, algorithmic thinking, recursive thinking, and pattern recognition could help them to gain programming knowledge.

This study employed a pragmatic design science research methodology and strengthened the rigor of the framework by infusing an online co-design process. The online co-design process was formulated to accommodate the prevalent situation of the COVID-19 pandemic, where a face-to-face co-design meeting with stakeholders was infeasible for creating interventions.

This research produced three main outputs: (i) an online co-design process as a pedagogical approach for conducting a user-centered study; (ii) an artefact called iThinkSmart, developed to immerse learners in a virtual world while learning computational thinking concepts by playing mini-games; and (iii) proposed guiding principles for mainstreaming computational thinking education in Nigerian higher education institutions.

The study results delineated that co-designing smart learning environments to facilitate students' understanding of computational thinking is a pragmatic pedagogical approach in the Nigerian context. Students who participated in the co-design process gained 21st-century skills such as creative thinking, problem solving, and collaboration to design mini-games. The outcomes of the demonstration and evaluation of the virtual reality-based smart learning environment – iThinkSmart - suggested that students gained computational thinking competence and cognitive benefits as well as enhanced interest and attitudes toward computational thinking education. Therefore, this study offers insights into how contemporary educational interventions can be developed by connecting contextual scenarios through a co-design process to foster students' learning experience. Furthermore, the study contributes to knowledge by associating relevant theories and pedagogical approaches that support the design and development of virtual reality-based smart learning environments to facilitate computational thinking education.

***Universal Decimal Classification**: 004.946, 004.96, 37.018.43:077, 004.4 `23,* 004.4`236

***Library of Congress Subject Headings**: Shared virtual environments; Educational games; Problem solving; Computer-assisted instruction—Programming: Student-centered learning; Computer games -Design; Distance education; Co-design; Smart learning environment; Computational thinking, game-based learning, virtual reality*

***Yleinen suomalainen ontologia**: oppimisalustat; tietokoneavusteinen oppiminen; tietokonepelit; ongelmanratkaisu; opetusohjelmat; pelisuunnittelu; oppilaslähtöisyys*

Acknowledgment

First, I give my earnest gratitude to God Almighty for helping me throughout my academic journey, and for providing all that I need, including wisdom, knowledge, and understanding up to this moment of completing my PhD.

My special appreciation to my supervisors – Dr Solomon Oyelere, Dr Jarkko Suhonen, and Professor Markku Tukiainen – for their unrelenting and incessant support in terms of mentorship, extensive supervision, and guidance throughout my PhD. They have made this academic journey smooth by being kind, open, and accessible, always providing creative and insightful instructions towards improving my research. I must particularly thank Dr Solomon for his wise counsel and other personal support rendered to me since I met him until now. 'Solomon's wisdom' has birthed the output of this study and I'm forever grateful to him. I'm very grateful to Dr Jarkko for his unrelenting support, providing deep insights and timely response during this study. His enormous experience has sharpened me to become a better researcher. I want to specially thank Professor Markku Tukiainen, the head of the department, School of Computing for his excellent supervision and for creating time to read through my manuscripts and provide insightful comments. Besides, Professor Markku has relentlessly supported my research through funding to successfully complete this study as scheduled.

Furthermore, I would like to thank Professor Eric Sunday Adewumi for facilitating my research, co-authoring, and providing guidance where necessary. In addition, I'm grateful to the administrators, staff, and students at Federal University Lokoja for providing the platform for my research. Similarly, my appreciation to Professor Teemu H. Laine for his constructive comments and co-authoring an article in this study.

I'm grateful to the School of Computing and the University of Eastern Finland for awarding me grants and research facilities to support my study. My special thanks to Professor Matti Tedre, the leader of the technology for learning and development group for his inspiring comments and professional counsel. I especially thank Professor Erkki Sutinen for opening my eye towards contextualizing my study. Also, my special thanks to all faculty, and colleagues of the edTech research group for providing help in different ways during my doctoral study. I appreciate the professional comments about my work from Dr Ilkka, Dr Calkin, Dr Saqr, and Dr Tapani; I enjoyed the warm moment I shared with Ismaila, Samuel, Dr Joni, Jani, and Alem in our office. Also, a special thanks to Nacir for providing inspiration during the study; thanks to our senior colleagues - Dr Alsen, Dr Nasibu, Dr Ezra, and all those who contributed to this study in one way or the other.

My sincere appreciation to the pre-examiners, Professor Francisco José García-Peñalvo and Professor Leila Goosen for painstakingly reading this dissertation and providing valuable comments regarding it. In addition, I am particularly thankful to Professor Erik Barendsen for accepting to serve as my opponent. Special thanks to my spiritual leaders and mentors among whom are pastor James Ayan (PhD), pastor Sunday Olaleye (PhD), pastor Brain Zyambo, and other pastors and leaders of the Deeper Life Bible Church Finland whose names space could not permit me to mention.

My hearty appreciation to my parents Mr. Joseph Agbo and Mrs. Mary Joseph for their love and care that has brought me thus far. I wish they could celebrate this moment with me. Furthermore, I'm deeply grateful to my in-laws Dr. & Mrs. Akuh, and to my siblings and their families: Mr. Chris Agbo, Mr. Monday Agbo, Mrs. Ladi Ameh, and Mr. David Agbo for standing by me throughout this study.

I couldn't have reached this feat without the unrelenting support of my wife, the one and only 'Sweetheart' - Barr. Naomi Agbo-Akuh - for her unflinching love, prayers, and support showered on me and our kids. I'm grateful to you LOVE, for taking the time to read through my manuscripts including this dissertation and correcting the English Language. A special thanks to our lovely kids – Perfect, Praise, & Peace for bearing with me when I'm away doing research. Your company is always a source of motivation.

Joensuu, April 2022
Friday Joseph Agbo

LIST OF ORIGINAL PUBLICATIONS

ARTICLE I

Agbo F. J., Oyelere, S. S., Suhonen, J., & Tukiainen, M. (2019). Smart learning environment for computing education: readiness for implementation in Nigeria. In *EdMedia+ Innovate Learning* (pp. 1382–1391). Amsterdam, Association for the Advancement of Computing in Education (AACE).
https://www.learntechlib.org/p/210149/

ARTICLE II

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Adewumi, S. (2019). A systematic review of computational thinking approach for programming education in higher education institutions. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1–10) ACM.
https://dl.acm.org/doi/abs/10.1145/3364510.3364521

ARTICLE III

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Tukiainen, M. (2019). Identifying potential design features of a smart learning environment for programming education in Nigeria. *International Journal of Learning Technology, 14*(4), 331–354. https://www.inderscienceonline.com/doi/abs/10.1504/IJLT.2019.106551

ARTICLE IV

Agbo, F. J., Sanusi, I. T., Oyelere, S. S., & Suhonen, J. (2021). Application of virtual reality in computer science education: A systemic review based on bibliometric and content analysis methods. *Education Sciences, 11*(3), 1–23.
https://www.mdpi.com/2227-7102/11/3/142

ARTICLE V

Agbo, F. J., Oyelere, S. S., & Bouali, N. (2020). A UML approach for designing a VR-based smart learning environment for programming education. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1–5). IEEE.
https://ieeexplore.ieee.org/abstract/document/9273956

ARTICLE VI

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Laine T. H. (2021). Co-design of mini games for learning computational thinking in an online environment. *Education and Information Technologies, 26*(5), 5815–5849.
https://link.springer.com/content/pdf/10.1007/s10639-021-10515-1.pdf

ARTICLE VII

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Tukiainen, M. (2021). Design, development, and evaluation of a virtual reality game-based application to support computational thinking. Submitted to *Educational Technology Research and Development*.

AUTHOR'S CONTRIBUTION

The articles included in this dissertation are original research outputs that focused on the design and implementation of a smart learning environment to facilitate computational thinking education in the Nigerian context. The author was the main contributor to all of the articles, which are denoted as Article I - VII. A detailed description of the author's contributions in each of the articles is presented as follows:

Article I: I am the main author of this article, which explored the possibility of implementing a smart learning environment in the context of Nigerian higher education institutions. While the manuscript was written by me, the other authors contributed to refining the research focus, provided comments on the first draft, and even guided me in how to respond to the reviewers' comments during the resubmission of the manuscript to a conference.

Article II: I am the main author of this article, which reviewed articles that have discussed the use of computational thinking as an approach for teaching and learning programming at higher education institutions. While I gathered the data and formulated the methodology, Oyelere and I synthesized the data and presented the results. Suhonen provided technical comments and insights into the data analysis. Furthermore, Suhonen and Adewumi proofread the manuscript and made corrections where necessary to improve the final draft.

Article III: I am the main author of this article, which mainly focused on complementing the output from the Article I by further engaging stakeholders to identify the potential features of a smart learning environment tailored to the facilitation of programming education among Nigerian university students. All of the authors jointly conceptualized the study and developed the study procedure, including the instruments. I conducted the field data collection, analysis, presentation, and drafting of the manuscript. Oyelere, Suhonen, and Tukiainen proofread the manuscript several times and provided constructive comments for improving the study before submission to a journal.

Article IV: I am the main author of this article, which investigated the role of virtual reality in computer science education to discover the state of the art for supporting our choice of technology for implementing the proposed smart learning environment. I was responsible for conceptualizing the study, collecting the data, conducting the analysis, and drafting the first version of the manuscript. Sanusi and Oyelere discussed the findings of the study and improved other sections of the manuscript after proofreading it. Suhonen also proofread and provided comments for improving the study before the final submission.

Article V: I am the main author of this article, which established the requirement specifications for a virtual reality-based smart learning environment using a formal notation of universal modeling language (UML). Oyelere and I conceived the study. I drafted the manuscript, designed the models, presented the system architecture, and discussed the study with pointers to our future work. Furthermore, Oyelere and Bouali proofread the manuscript and provided comments for improving the final draft of the manuscript.

Article VI: I am the main author of this article, which demonstrated how to co-design contextual mini-games with students to support their understanding of computational thinking concepts. The co-design process, which took place in an online environment, was designed by Oyelere and I. I developed the manuscript while Oyelere reviewed and provided comments for improving the study. Suhonen and Laine reviewed the manuscript and provided critical comments and contributions, which improved all of the sections of the manuscript. Laine facilitated the professional language editing, which improved the final version of the manuscript before it was submitted to a journal.

Article VII: I am the main author of this article, which presented the design, development, and initial evaluation of a virtual reality game-based smart learning environment called iThinkSmart. Said learning environment was aimed at facilitating students' understanding of computational thinking concepts and improving their problem-solving skills. I was the developer of the prototype tool iThinkSmart. I curated the entire study by demonstrating the tool to students, collecting data on the initial evaluation, and analyzing the results. Oyelere provided support during the demonstration of the artefact to students and provided critical comments on the manuscript for improving it. Suhonen and Tukiainen reviewed the manuscript, provided comments, and approved the final version before it was submitted to a journal.

Table of contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

OOP	Object-oriented programming
NECO	National Examination Council
JAMB	Joint Admission and Matriculation Board
VR	Virtual reality
WAEC	West Africa Examination Council
STEM	Science, technology, engineering, and mathematics
NERDC	Nigeria Educational Research and Development Council
NPE	National Policy on Education
OCD	Online co-design
RQ	Research question
DSR	Design science research
IBM	International Business Machine
ICT	Information and communication technology
GDP	Gross domestic growth
SLE	Smart learning environments
CT	Computational thinking
HEIs	Higher education institutions
4G	Fourth generation
5G	Fifth generation
TLBO	Teaching learning-based optimization
GBL	Game-based learning
IDEF0	ICAM definition for function modeling
RM	Research method
IT	Information technology
SRS	Software requirement specification
OD	Objective distance
HMD	Head-mounted display
MoPaTH	Mount Patti Treasure Hunt
BFS	Breadth-first search
CAS	Computing at School

1 Introduction

1.1 Study background and motivation

In Nigeria, the majority of students at higher education institutions lack the understanding of programming concepts required to solve problems, which is the main motivation behind the present study. Problem solving entails the skills of problem decomposition, algorithm formulation and design, and recursive thinking, which are essentially concepts of computational thinking (Lyon & Magana, 2020). A study reported that the majority of computer science students in Nigerian higher education institutions, especially novices, found it difficult to pass the introductory programming course (Dasuki & Quaye, 2016). Furthermore, a recent study reported that over 60% of students who enrolled for an Introduction to Programming course at a Nigerian university failed for several reasons, including novices lacking an understanding of the course (Sunday, et al., 2020). Understanding computational thinking concepts can facilitate students' understanding of introductory programming courses taught at the university level (Amber, 2014).

In the Nigerian context, all students enrolling for a university degree, including one in computer science, must have obtained a senior school certificate, as presented in Figure 1. To do so, they must have passed the final examination, which is conducted mainly by two constituted bodies, namely the West African Examinations Council (WAEC) and the National Examination Council (NECO) of Nigeria (Patriot, 2019). These students are further subjected to a university entry-level examination conducted by the Joint Admissions and Matriculation Board (JAMB). In all of these examinations, students' knowledge of science, technology, engineering, and mathematics (STEM)-related subjects such as mathematics and physics is tested. Unfortunately, the majority of these students do not have a background in programming or even problem-solving skills from secondary school (Dasuki & Quaye, 2016). However, the Nigerian Educational Research and Development Council (NERDC) introduced a new junior secondary school education curriculum in 2011 that included computer studies (Igbokwe, 2015). Figure 1 presents an overview of Nigerian higher education.

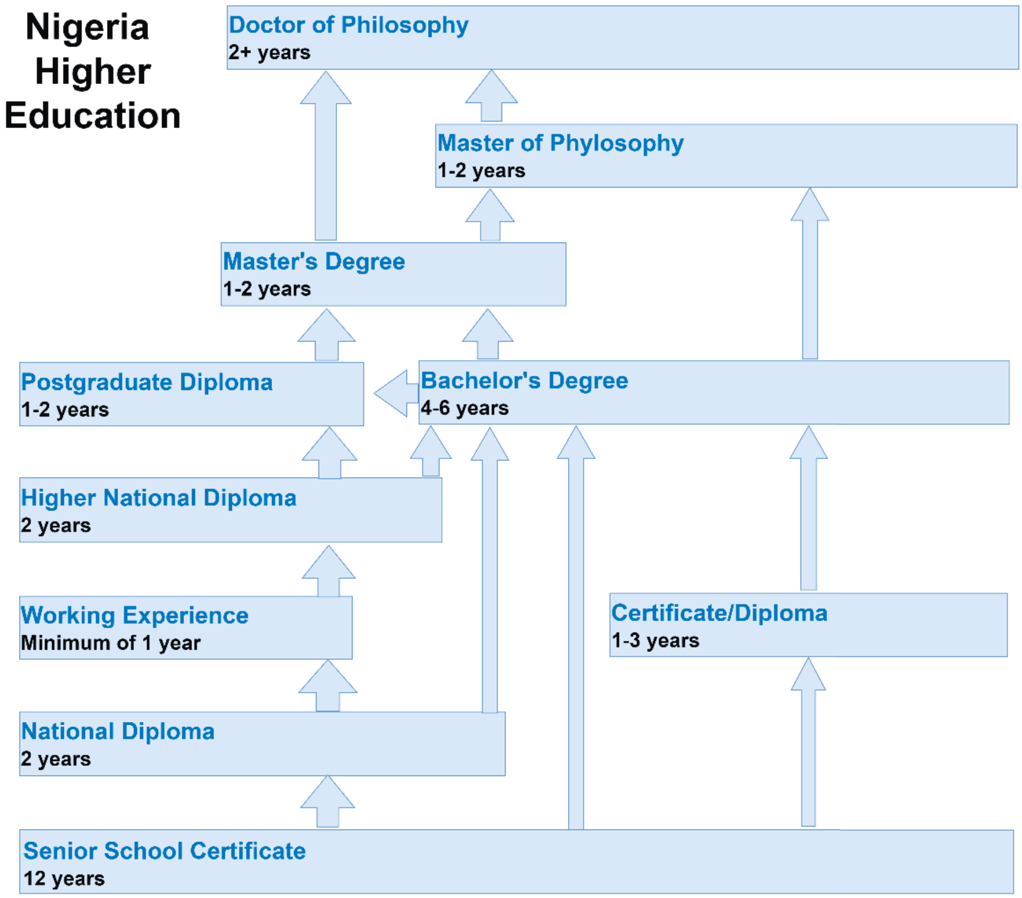


Figure 1. Overview of Nigerian Higher Education.

On the other hand, the senior secondary school curriculum was upgraded, with compulsory topics such as Computer and Information Technology (IT) and Civic Education added to the existing list of subsects such as Mathematics and English (Ahmadi & Lukman, 2015). According to Ahmadi and Lukman (2015), this restructuring of the senior secondary school curriculum in Nigeria was aimed at preparing students for 21stcentury higher education and future technological challenges. Unfortunately, based on my personal experience, only a few of the students whose parents can afford to sponsor them in private schools have the privilege of attending computer and IT classes at the secondary school level. This scenario is not only applicable to Nigeria alone but also to other African countries (Mufeti & Sverdlik, 2017).

The overwhelming demand for university education in Nigeria is growing every year, placing intense pressure on the available universities to admit a huge number of students beyond their teaching and learning capacity (Okoroma, 2008). As presented in Figure 2, the contextual challenges facing computing education in Nigerian higher education include a high student-to-teacher ratio, insufficient number of laboratories, limited learning materials, and poor physical infrastructure (e.g., classrooms) that is unsuitable for numerous students to learn (Oyelere, et al., 2016).

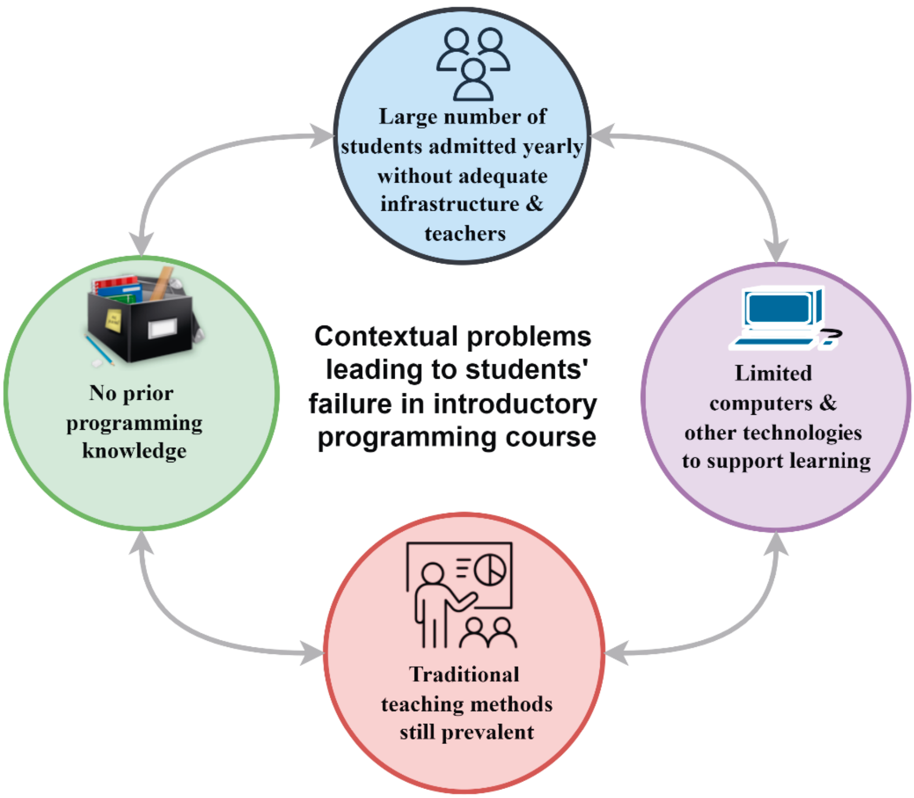


Figure 2. Contextual problems that motivated the study.

Consequently, many novices find learning introductory programming challenging. For some students, this is due to anxiety caused by exposure to a new topic and learning situation (Dasuki & Quaye, 2016; Oribhabor, 2020); for others, it is caused by the poor learning conditions and limited resources accessible to them (Oyelere, et al., 2016).

On top of all the aforementioned challenges, the traditional methods of teaching and learning introductory programming in Nigerian university classrooms can make the course boring, uninteresting, and cumbersome to students (Akinola & Nosiru, 2014). In a typical Nigerian university, lecturers use notes in the form of electronic or hard copy handouts, whiteboard and markers, and in some cases PowerPoint slides (Oyelere, 2017). With such traditional methods of teaching and learning, students can only write algorithms, pseudocodes, and even short computer programs on paper and whiteboards; hence, they are unable to practice on computers to gain knowledge on how computer programs behave or whether an error has occurred, or even to understand the logic behind the execution of a computer program. Consequently, most students grow frustrated and either drop out or change to another discipline (Akinola & Nosiru, 2014).

One approach for facilitating novices' understanding of introductory programming courses in Nigerian higher education is to demonstrate computational thinking knowledge. According to scholars (Aho, 2012; Wing, 2008), computational thinking is a thought process formulated to provide a solution to a problem in a finite number of steps. Although a consensus is lacking on the definition of computational thinking among experts, one critical theme that most scholars' definition points to is problem-solving skills. Such skills may remain elusive to Nigerian students if there is no change in teaching and learning from traditional methods to technology-enhanced learning. Today, higher education institutions in the Western world are fostering the teaching and learning of computational thinking concepts in their classes to provide a 21st-century learning experience (Lai, et al., 2019; Hooshyar, et al., 2020; Hooshyar, et al., 2021). Indeed, a recent study revealed that research conducted to promote computational thinking in schools is heavily concentrated in the United States and Europe, suggesting that Africa is far behind in the quest for mainstream computational thinking in schools, including higher education institutions (Saqr, et al., 2021). This creates a greater need for the adoption of computational thinking in Nigeria's higher education institutions to support students' problem-solving skills. Otherwise, the majority of students will continue to suffer failure and frustration (Balogun, et al., 2019), which may lead to total withdrawal from study. Even if some of the students manage to graduate, the majority may not have the skills required for future job demands.

1.2 Research objectives and questions

This dissertation addresses contextual issues related to the lack of comprehension of programming concepts by students and the unmotivating classical mode of teaching in Nigerian higher education institutions as depicted in Figure 2. The aim of this dissertation is to develop teaching approach and tools to facilitate computational thinking education as well as to suggest guiding principles for mainstreaming computational thinking in higher education institutions in the Nigerian context. The research is divided into three main parts. First, the study explicates the problems confronting students' comprehension of computational thinking concepts in a Nigerian university in order to understand how to facilitate their learning experience. Second, the study defines users' requirements necessary for the design and development of a smart learning environment for computational thinking education in the Nigerian context. Third, the study aims to co-design and develop a student-centered virtual reality (VR)-based smart learning environment to facilitate computational thinking education in the Nigerian context. This dissertation is grounded in studies that were conducted for Articles I–VII, which specifically investigated four main aspects of this research work. First, the study explicated the level of preparedness and contextual factors that can impede or aid the implementation of a smart learning environment to support computational thinking. Second, the study reported in Article III investigated the technical and pedagogical features necessary for the implementation of such a learning environment. Third, the study reported in Article VI focused on the design and implementation of a student-centered learning environment through a participatory co-design process. Fourth, an evaluation of the learning environment was conducted to ascertain the extent to which students gain support from the proposed intervention. Therefore, this dissertation seeks to answer the following research questions (RQs):

RQ1. What is the readiness of students and teachers in a Nigerian higher education institution for embracing smart learning environments to facilitate computational thinking education?

While smart learning environments promise a 21st-century learning ecosystem and aim to provide an enhanced learning experience, it is crucial to investigate the Nigerian higher education institution context to explicate the current status quo in terms of teaching and learning (Oyelere, et al., 2016). Therefore, RQ1 engaged Nigerian university students and teachers, who are the main users of smart learning environments, to unravel whether they are familiar with or have used smart learning environments, as well as their experience and readiness for embracing the use of smart learning solutions. Moreover, an overview of the technological, social, and economical situation of the context needed to be investigated to gain an understanding of whether the proposed smart learning approach is suitable and sustainable (Article I).

RQ2. What are the key requirements to implement in a smart learning environment to support students' computational thinking education in the context of a Nigerian higher education institution?

According to research (Zhu, et al., 2016; Liu, et al., 2017), the features of smart learning environments are often technology- or pedagogy-related. However, the study reported in Article III attempted to create a bridge between these two aspects by identifying five components of smart learning environments, namely user, device, technology, context, and pedagogy. This dissertation classifies features related to the technology, device, context, and user as technical features. Therefore, developing a solution to support computational thinking that conforms to technical and pedagogical features of a smart learning environment would require a diligent investigation into the existing technologies, devices, contexts, learner models, and learning approaches that can provide an enhanced learning experience. In the case of this dissertation, building students' problem-solving skills through a highly immersive, personalized, interactive, motivating, and engaging (Chao, 2020; Bolivar, et al., 2019) solution is the goal. Hence, a need existed to conduct a comprehensive review of the state of the art to uncover relevant features that should be developed in the smart learning environment to support students' computational thinking and problem-solving skills.

RQ3. How can student-centered VR-based smart learning environment be co-designed and implemented to facilitate computational thinking education in the context of a Nigerian higher education institution?

To implement a more meaningful computer-mediated learning environment to support students' understanding of computational thinking concepts and build their problem-solving skills, it is critical to involve users in all phases starting from the conceptual design to the evaluation stage (Havukainen, et al., 2020). Engaging users to develop an educational intervention through a participatory process can facilitate inclusive and student-centered learning (Karumbaiah, et al., 2019; Wu, et al., 2020). Unfortunately, the recent COVID-19 pandemic caused a huge challenge on nearly all sectors of life, including teaching and learning, where most academic institutions were constrained to move all education to online mode (Fidalgo-Blanco, et al., 2020; Daniel, 2020). While university institutions were left to navigate how to teach and assess students in online mode during the COVID-19 (Cabero-Almenara & Llorente-Cejudo, 2020; García-Peñalvo, et al., 2021a; García-Peñalvo, et al., 2021b), researchers must be innovative to conduct studies that are student-centered during the pandemic era (Béjat & Vera, 2022).

Therefore, the third RQ sought to demonstrate how to co-design a smart learning environment with students through a participatory OCD process. The OCD process is an alternative method for conducting a participatory study with users where face-to-face meetings are not practical (Béjat & Vera, 2022).

Aside from addressing these RQs, this dissertation attempts to contribute to knowledge by introducing guiding principles for mainstreaming computational thinking education in the context of Nigerian higher education institutions. The guiding principles are the outcomes of the whole dissertation study.

1.3 Research methodology and contributions of the dissertation articles to the study

In providing answers to these RQs, different research strategies were adopted at different stages of the design, development, and evaluation. Similarly, several research methods were applied, including mixed methods. The research conducted for this dissertation was aimed at facilitating computational thinking knowledge through a VR-based smart learning environment assembled with contextual mini-games. For achieving this goal, a pragmatic research methodology was adopted, namely design science research (DSR). Today, DSR has become one of the most popular research methods in information systems for development (Pereira, 2017; Kapinga, 2020) as well as in computer science research (Kolog, 2017; Oyelere, 2017). For example, Kapinga (2020) adopted DSR to develop a mobile application to support women entrepreneurs in the Tanzanian context. Similarly, Oyelere (2017) implemented a mobile learning system for computer science education in the Nigerian higher education context.

Moreover, the relevance of the DSR methodology to this dissertation research was strengthened by another recent study that used it to develop a framework for teaching computational thinking in the field of computing education (Apiola & Sutinen, 2021). One of the key characteristics of DSR is its methodological approach to designing and implementing artefacts, which is focused on addressing contextual problems emanating from real-life situations (Oyelere, et al., 2018; Apiola & Sutinen, 2021; Oyelere, 2017). The problem for which this research aims to provide a solution is related to how to teach computational thinking in Nigerian higher education institutions to facilitate students' problem-solving skills, which are necessary for them to comprehend programming concepts. According to Tommelein (2020, p. 360), DSR is about "designing and making artefacts to fulfil a purpose, and then testing and validating that they indeed are fit-for-purpose."

In particular, this research adopted the DSR framework proposed by Johannesson and Perjons (2014), in which five main activities are followed to arrive at a concrete solution to an identified problem. These activities are problem explication, requirement definition, design and development of the artefact, demonstration of the artefact, and evaluation of the artefact. These activities outlined in the DSR framework can be executed iteratively, incrementally, or even linearly depending on how the outcome of the process from the DRS method is verified as meeting the initial expectations (vom Brocke & Maedche, 2019); thus, they allow the researcher to continuously refine the approach to achieve an improved outcome. In addition, DSR is a flexible method that allows researchers to include other approaches to solving a problem depending on the nature of the specific problem being addressed.

Moreover, in the DSR framework, two main outcomes are expected. The first is a developed artefact aimed at providing a solution to a contextual problem, and the second is original findings that contribute to the existing knowledge base (both scientific and practical knowledge) of the field. The research contributions to the knowledge base may include theories or models that can be communicated through academic publications in journals and conferences (Johannesson & Perjons, 2014). In this dissertation, the two main expected outcomes of the DSR framework emerged by rigorously following all five of the activities demonstrated at the different stages of the study. The outputs are (i) an artifact co-designed with students and (ii) guiding principles for mainstreaming computational thinking education in the context of Nigerian higher education institutions.

Table 1. Connection of the research questions with the DSR framework.

Research question	**DSR activities**	**Dissertation articles**	**Research method and strategy**	**Research goal**
RQ 1	Problem explication	Articles I and III	Mixed methods	To explicate the problem, possibilities, and users' expectations regarding the teaching and learning of computational thinking concept in the Nigerian context.
RQ 2	Requirement definition	Articles II, III, IV, and V	Literature survey, mixed methods	To define user-centered and functional requirements for designing a smart learning environment to facilitate computational thinking education.
RQ 3	Design and development of the artefact	Articles V, VI, and VII	Prototyping, co-design, mixed methods	To showcase how to co-design student-centered smart learning environment to facilitate computational thinking skills.
RQ 3	Demonstration of the artefact	Articles VI and VII	Co-design, experimental, mixed methods	To understand students' learning experience and how they gained computational thinking knowledge from the co-design process.

RQ 3	Evaluation of the artefact	Article VII	Experimental, mixed methods	To investigate the efficacy of the developed VR prototype and obtain users' feedback regarding what aspects need improvement.

The activities in the problem explication stage of this research were covered in Articles I and III, for which we investigated the level of awareness and use of smart learning environments by students and teachers in a Nigerian university; examined their expectations of a technology-mediated solution for supporting the teaching and learning of computational thinking and programming concepts; and explored what tools and technology are available or that users possess that can enable the deployment of smart learning environments in that context. In addition, the problem explication stage of the DSR framework provided answers to RQ1 by unravelling the readiness of users and the preparedness of the Nigerian context to adopt a smart learning environment. While Article I used a quantitative research strategy, Article III employed a mixed research approach consisting of questionnaires and focus-group interviews. The outcomes of this stage were users' expectations of features and requirements for the proposed smart learning environment to foster computational thinking and programming knowledge, and a confirmation of the availability of infrastructure and tools to support the implementation of a smart learning environment in the context of Nigerian higher education institutions.

In the second stage of the DSR framework, a comprehensive study was conducted to concretely define the requirements for developing the proposed smart learning environment to support computational thinking. Adding to the users' features and expectations already uncovered in the problem explication stage, a systematic review of previous studies was conducted, focusing on the use of the computational thinking approach to teach programming in higher education institutions. The goal of the activities in this stage was to provide answers to RQ2, covered by Articles II, III, and IV, mainly to determine what technology and pedagogy could be utilized for implementing a smart learning environment in the context of Nigeria's higher education institutions. While Article II reported on computational thinking as an approach for programming education at higher education institutions, Article IV dwelt on the use of VR technology to support computer science education. The outcome of this stage was a roadmap toward designing and developing a low-fidelity prototype of the artefact.

The third stage of the DSR framework allowed for the design and development of the artefact based on the requirements defined in the previous stage. The activities in the design and development stage provided answers to RQ3. We first demonstrated how to co-design a low-fidelity prototype of a smart learning environment with students through a participatory process; second, we showcased the implementation of said prototype using a formal notation of universal modeling language (UML); and lastly, we developed a high-fidelity prototype of the VR-based smart learning environment called iThinkSmart, which consisted of mini-games to support students' computational thinking and problem-solving skills. This stage was covered in Articles V, VI, and VII. Article V utilized UML as a methodology to model the system; Article VI adopted a participatory research method with stakeholders to co-design contextual mini-games to provide computational thinking knowledge; and Article VII employed the software development process to design and implement iThinkSmart.

The fourth and fifth stages of the DSR framework dwelt on the demonstration and evaluation of the low- and high-fidelity prototypes of the VR-based smart learning environment with computer science students at a Nigerian university. The demonstration focused on obtaining initial feedback from the users regarding whether these first prototypes conformed with their initial expectations, whereas the evaluation sought to ascertain whether the initial prototype of iThinkSmart truly supported users' computational thinking competence, provided cognitive benefits, and motivated learners. As presented in Articles VI and VII, users were able to experience both the low- and high-fidelity prototypes of mini-games co-designed with them and then to provide useful feedback for improving the further development of the iThinkSmart prototype.

Moreover, to arrive at the objectives of this dissertation, the seven articles published in journals and conferences are presented to demonstrate their main contributions to this research. Table 2 presents the main purpose of the articles in the dissertation along with their outcomes.

Table 2. Contributions of the dissertation articles to the study.

Article	Main purpose	Study outcome
Article I	This paper investigated the expectations regarding the technical and pedagogical features of a smart learning environment from students and teachers, and also the potential challenges that may confront the implementation of the smart learning environment for programming education in the Nigerian context.	The outcomes of this study indicated that i. learners placed high expectations on the proposed smart learning environment to support programming education; ii. they were eager to have an experience that would impact their learning; iii. the majority of students and teachers in a Nigerian higher education institution own smartphones and are active mobile Internet subscribers, which are good indicators for implementing the new learning solution.
Article II	This paper focused on reviewing articles that have discussed computational thinking as an approach for teaching and learning programming at higher education institutions. The main purpose was to investigate how the use of the computational thinking approach has been explored for teaching programming in	The study revealed that the use of computational thinking as an approach for programming education i. has been practiced for approximately a decade in different contexts; ii. this finding provided evidence that supports our adoption of the computational thinking approach as the pedagogy for implementing a smart learning environment in the context of Nigeria's higher education institutions.

	higher education institutions.	
Article III	The main purpose of this paper was to complement the earlier studies by further engaging the stakeholders through mixed-methods research to identify potential features of a smart learning environment that are particularly relevant to programming education.	The findings of the study suggested that the user-centred potential features of a smart learning environment in this context include i. learning guides; ii. a personalized learning environment; iii. quick feedback mechanisms; iv. automatic task scheduling.
Article IV	This paper systematically investigated the role of VR in computer science education. The paper intended to unravel research evidence that may support our choice of VR technology as the enabler for implementing the proposed smart learning environment.	The outcomes of this study demonstrated i. how previous studies on VR for computer science education have been conducted in the recent past and what approaches these studies have adopted; ii. insightful information regarding research hotspots such as games, gamification, collaborative learning, and immersive learning dominating pedagogies to facilitate computer science education.
Article V	This paper established the requirement specification for a VR-based smart learning environment by designing and modeling the learning scenarios using a formal notation of the UML approach.	The outcomes of this study were i. activities that aroused the design and prototyping of the smart learning environment based on the DSR method; ii. the system architecture for the VR-based smart learning environment; iii. a scenario-based model for teaching and learning computational thinking concepts.
Article VI	This paper demonstrated how to co-design contextual mini-games with students to	The findings of this study indicated that the participants

	support the teaching and learning of computational thinking concepts in an online environment. The study innovatively introduced an OCD process, which is one of its main contributions to this dissertation.	i. gained relevant knowledge that supported their creative thinking in contextual game scenarios and game elements related to computational thinking; ii. learned how to collaboratively co-design contextual mini-game prototypes of low fidelity, and were motivated to design more educational mini-games in their future studies.
Article VII	This paper presented the design and implementation of a high fidelity prototype of a VR-based smart learning environment (called iThinkSmart) aimed at supporting learners in gaining computational thinking knowledge and problem-solving skills.	This study revealed that students who played the iThinkSmart VR mini-games i. gained computational thinking competency and that the tool has the potential to foster computational thinking knowledge in higher education institutions in Nigeria; ii. gained cognitive benefits and developed a greater interest in and better attitude toward learning computational thinking concepts.

To summarize, Figure 3 graphically demonstrates the role and contributions of these seven articles within the dissertation. It highlights the objectives of each article toward the development of a pedagogical co-design process that engendered the implementation of a VR-based smart learning environment to facilitate computational thinking.

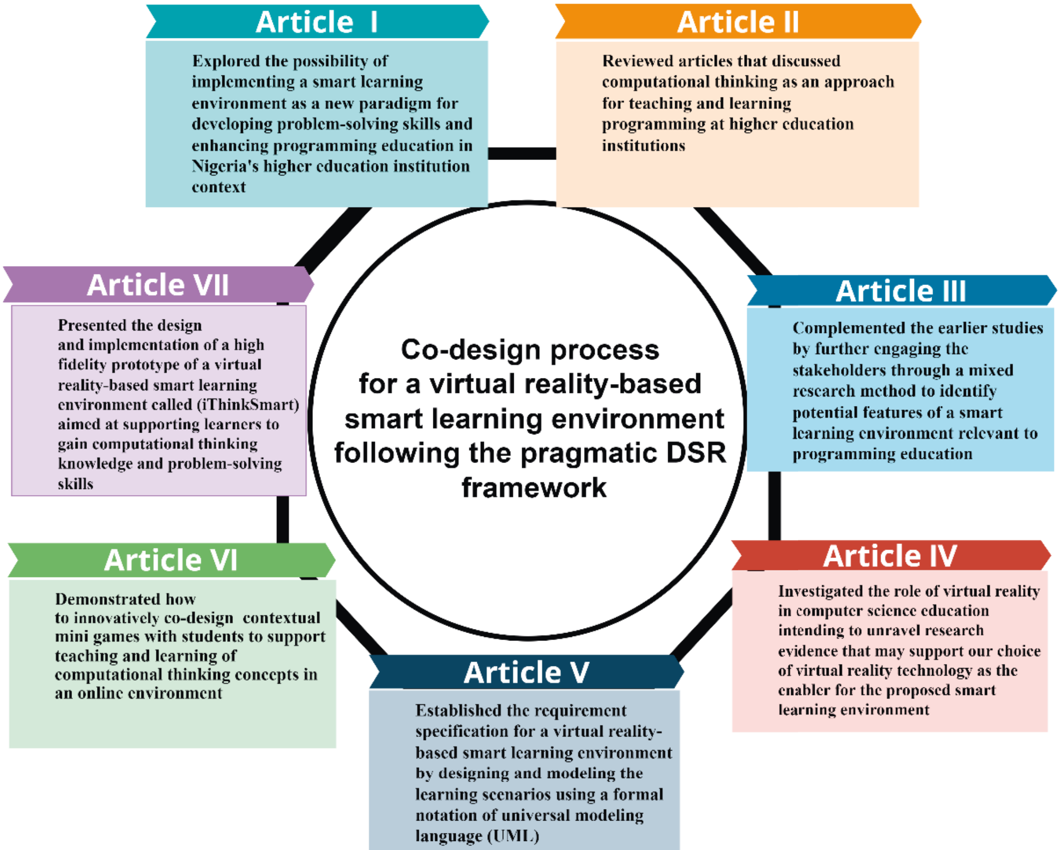


Figure 3. Summary of publications contributing to this dissertation study.

In general, the studies (Articles I to VII) provided significant contributions in positioning computational thinking in the higher education institution context as well as empirical evidence regarding how to design a student-centered smart learning environment. Further findings and discussions from these studies are presented in later chapters.

1.4 Structure of the dissertation

This dissertation contains five chapters and seven publications, and the remainder is structured as follows. Chapter 2 presents the literature review of key concepts in this dissertation, including the relevance of smart learning environments to computational thinking as well as the theories, technologies, and pedagogies connected to this research. Chapter 3 focuses on the research design and methodology, including the entire process of the DSR framework and how it was strengthened through the integration of a participatory co-design process to implement the VR-based smart learning environment artefact. Chapter 4 presents the demonstration, evaluation, and results of the pilot study to determine what worked and what did not. This chapter also presents the user feedback that guided the next iterative improvement of the tool. Chapter 5 discusses the main findings of the research as well as the contributions in terms of guiding principles that can support mainstream computational thinking in the context of Nigeria's higher education institutions. Finally, the chapter concludes by reflecting on the limitations and providing recommendations for future research.

2 Literature review

This chapter consists of state-of-the-art in computational thinking, computing education in Nigeria, and theoretical and pedagogical foundations for this research. The author begins by presenting an overview of computational thinking and what constitute its concepts from literature. The chapter also showcased how computational thinking concepts are fundamentals of programming education. Further, the author unravels the development of computing education in Nigeria and provided the operational definition of a smart learning environment in this dissertation. Finally, the chapter concludes by demonstrating relevant theories, pedagogies, and design approaches assembled and utilized in this research (Agbo et al., 2021d). This chapter mainly provides the background for the dissertation in terms of technology, theory, pedagogy, design, and other relevant methods applied in different stages of the study.

2.1 Computational thinking

As defined in the UK Computing at School (CAS) technical guide for educators (Csizmadia et al., 2015, p. 6), computational thinking is a “thought process involving logical reasoning by which problems are solved and artefacts, procedures, and systems are better understood.” This definition of computational thinking conforms to earlier scholars’ viewpoints (Wing, 2008; Aho, 2012; Tedre & Denning, 2016). From a historical perspective, the term computational thinking was first used by Papert and his colleagues in the 1980s. Their research provided a mathematical foundation for problem-solving through ‘procedural thinking,’ which was useful for mastering how to think like computers in that age (Papert, 1980). Two decades later, Wing (2006) used the term during her talk at the *Communications of the ACM* anniversary celebration. Since then, computational thinking has gained scholars’ interest with more research tilted toward promoting its education in K-12 (Wing, 2008). Noteworthily, Papert’s ‘procedural thinking’ has remained one of the key concepts associated with the design and facilitation of computational thinking for contemporary learners by educators in recent times (Barbero et al., 2020).

Computational thinking is crucial for ensuring that learners gain problem-solving skills (Mohaghegh & McCauley, 2016). In fact, problem-solving skills are not only critical for learners alone but also essential for all humans. According to the European Commission's digital competence framework for citizens (DigComp 2.1), every citizen is required to possess problem-solving skills (Carretero, et al., 2017). Similarly, the UK's national education system introduced a new curriculum for schools in 2014 with a focus on promoting the teaching and learning of computational thinking in classrooms (Humphreys, 2015). Thus, computational thinking is a popular topic among educators and scholars since problem-solving skills are required for a future job in this world of increasing technological advancements (Tedre & Denning, 2016). While computational thinking education is gaining greater ground in K-12 and high schools (Lye & Koh, 2014; García-Peñalvo & Mendes, 2018; Sanusi et al., 2022), the diffusion of computational thinking in higher education institutions is also beginning to gain momentum as studies have identified its potential for improving students' programming knowledge (Article II, Lyon & Magana, 2020). Therefore, one strategy for reducing the failure rate in introductory programming courses as well as the resulting withdrawal by students from pursuing a computer science degree is to introduce a computational thinking approach. Such an approach allows students to first build their problem solving, algorithmic thinking, and recursive thinking skills, which are necessary for understanding advanced concepts in introductory programming (Article II; Rojas-López & García-Peñalvo, 2018; Rojas-López, et al. 2019).

Educators have outlined the concepts that form the foundation for computational thinking knowledge, namely algorithmic thinking, decomposition, generalization (patterns), abstractions, and recursion (Csizmadia et al., 2015). Algorithmic thinking refers to the process of solving a problem through clearly defining a finite number of steps. Decomposition refers to the process of breaking down a complex problem into component parts, where each component can be easily understood and solved separately. Generalization or pattern recognition refers to the process of identifying patterns, similarities, and connections that exist in a given problem so that it can be exploited for solving new problems based on the previous solutions. Abstraction refers to the process of making a complex problem more understandable by isolating irrelevant and unnecessary details. Recursive thinking is a technique for solving complex computer science problems, such as graphs and artificial intelligence problems (Vilner et al., 2008), and encapsulates backward reasoning and reverse thinking (Ginat, 2004). Outside of computer science, recursive thinking has a wider application in other subjects such as mathematics, making it and other computational thinking concepts introduced in this section critical knowledge for contemporary learners.

While the aforementioned computational thinking concepts are regarded as pillars that provide a strong framework for studying computing in different contexts, especially in the K-12 setting (Csizmadia et al., 2015; Brackmann et al., 2016), other scholars' viewpoint of what concepts constitute computational thinking is broader in scope (Tedre & Denning, 2016). For example, concepts of computational thinking for professionals could also include program design, software engineering, computing theory, and artificial intelligence. Therefore, this section attempts to illustrate how computational thinking concepts can be harnessed for computing education with a focus on facilitating programming education among novices. Figure 4 depicts the relationships between computing education, programming education, and computational thinking concepts following the classification of Brackmann and his colleagues (Brackmann et al., 2016).

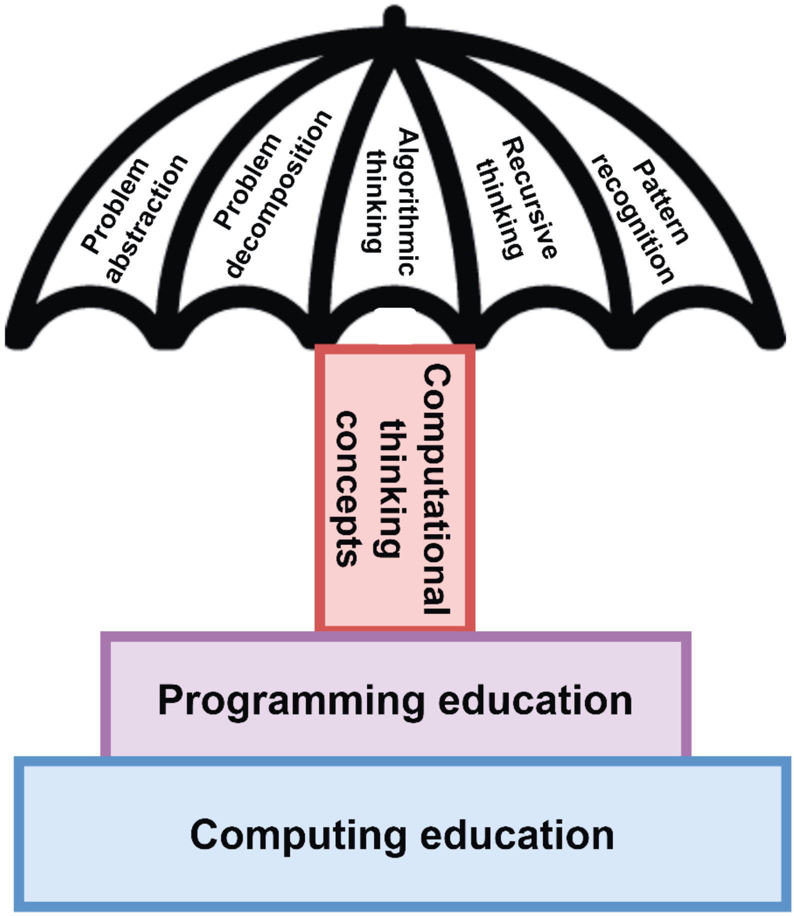


Figure 4. Foundational pillars of computational thinking education.

According to educators (Csizmadia et al., 2015; Brackmann et al., 2016), the integration of the aforementioned five concepts of computational thinking can facilitate programming education. In this dissertation, I focus on demonstrating how the concepts of computational thinking can be developed as mini-games to facilitate students' computational thinking knowledge, which can consequently support novices in developing programming knowledge (Rojas-López & García-Peñalvo, 2018; Rojas-López, et al. 2019; Figueiredo & García-Peñalvo, 2021)

2.2 Development of computing education in the Nigerian context

The history of computers in Nigeria, especially how and when they first appeared in the country, may be difficult to unravel. However, early experts and historians from Nigeria perceive computer science education to have begun when the IBM (International Business Machines) World Trade Corporation founded the IBM African Education Center in 1963, located at the University of Ibadan's campus (Anyanwu, 1978; Ogugua, 2016). According to Anyanwu (1978), the first computer for educational purposes was the IBM 1401. Then, universities in the west of Nigeria began to establish computer institutes that facilitated computing education across Nigeria's higher education institutions. According to Nwankwo and Njoku (2020), the National Policy on Education (NPE) document (NERDC, 2014) emphasized the development of computer education at secondary and tertiary institutions to allow students to gain the knowledge, skills, and competence required to manipulate and interpret computer languages. This goal suggests that the development of students' knowledge through formal education systems remains fundamental in the context of the Nigerian education system. However, the current provision in terms of human and infrastructural capacity to attain this goal remains a challenge (Mejabi, et al., 2017). For example, the number of institutions that train teachers for computing education is insufficient compared with the huge demand for such degrees (Nwankwo, 2018).

Moreover, Nigeria's population is estimated to be 200 million with 766 higher education institutions, including universities, polytechnics, colleges of education, and vocational schools, which is an inadequate number for meeting the country's huge educational demands (Nwankwo, 2018; Nwankwo & Njoku, 2020). This situation has placed the country's higher education institutions under intense pressure to admit more students than their capacity can handle (Okoroma, 2008).

The development of computing education in the Nigerian context seems to have progressed in recent years. According to one Nigerian agency, the information and communication technology (ICT) industry, manned by a majority of homegrown computer scientists, contributes over 13% to the nations' gross domestic product (GDP, Nigerian Communications Commission (NCC), 2019). These computer scientists have emerged from the country's higher education institutions and studied computing education to obtain a computer science or engineering degree. Researchers argued that the impact of computing education on producing more ICT-driven innovations in Nigeria could even outweigh the contributions of the oil and gas industry to the nation's GDP within two years (Nwankwo & Njoku, 2020). By contrast, other researchers from Nigeria, specifically in the field of computing education, argued that more effort is required to make computing education a more rewardable discipline (Obienu & Amadin, 2018). Obienu and Amadin (2018) investigated the influence of Nigerian university learning environments on computing students' academic performance. According to the authors, the learning environment not only consists of a well-equipped lecture room but also places computing education as one of the priority courses in the curriculum to create maximum interest and recruit qualified staff (Obienu & Amadin, 2018).

Unfortunately, Nigeria's higher education institutions are still facing several challenges that may limit the adoption and implementation of 21st-century learning environments. Some obvious challenges include the high cost of ICT infrastructure and Internet bandwidth, an unstable electricity supply, a lack of a strong implementation of ICT policy regarding cloud data, and insufficient government budget allocation or funding to cater for such an innovation (Ekundayo & Ajayi, 2009; Odediran, et al., 2015). Studies conducted by teachers and researchers of computing education in Nigeria's higher education institutions have demonstrated that one strategy for mitigating some of these challenges is to embrace cloud computing to deploy teaching and learning (Gital & Zambuk, 2011; Dogo, et al., 2013; Akin, et al., 2014).

Regarding the performance of students in computing education, studies have revealed that students from Nigeria find the courses, especially programming topics, difficult to understand (Oyelere et al., 2016; Oyelere et al., 2017). In particular, Idemudia et al. (2017) reported that factors such as hard work, dedication, interest, and self-motivation, which are required to study and pass programming topics, are lacking in most novices pursuing computer science degrees at Nigerian universities. Hence, their study investigated these factors and provided practical recommendations for developing programming courses and curricula to address some of the issues facing students. Similarly, Dasuki and Quaye (2016) investigated the causes of undergraduate students' failure in programming courses in Nigerian higher education institutions. Their study indicated that a "lack of intrinsic motivation, lack of future expectation, anxiety, peer influences, and poor lecturer skills and behavior" (Dasuki & Quaye, 2016, p. 1) were some of the factors causing students' failure in introductory programming courses. As a recent study reported, only 32.6% of students who participated in an introduction to programming course taught at a Nigerian university passed, suggesting a 67.4% failure rate for the course (Sunday, et al., 2020). The aforementioned findings generally reveal fundamental issues with computing education in Nigeria as a more pragmatic approach is required to facilitate students' understanding, retention, and performance in computing topics.

However, efforts are being made by educational technologists and researchers to provide interventions to support students in computing education, particularly in computational thinking and programming concepts. For example, Oyelere et al. (2017) developed a mobile application that integrated puzzle-based programming exercises and a contextual board game (Ayo) to facilitate the teaching and learning of programming concepts. According to the authors, the combination of these two pedagogical approaches (the Ayo board game and puzzle-based exercises) provided a new perspective on mobile learning (Oyelere, 2017). Furthermore, Oladipo et al. (2017) developed a scripting language called FulangS to facilitate computer science students' understanding of basic syntax and semantics of scripting. In addition, Agbo et al. (2020 & 2021c) conducted a study with Nigerian computer science students to demonstrate how social media platforms such as WhatsApp can enhance collaborative learning in topics such as programming basics, data structure, and databases. These efforts are noteworthy; however, there is still a huge gap in terms of innovations to facilitate computing education, which all stakeholders in this context must fill.

2.3 Smart learning environment for facilitating computational thinking in higher education institutions

The term "smart" is excessively used in everyday language. This term can often be ambiguous unless it is connected to a context. For example, some contextual uses of the term include smart people, smart city, smart agriculture, smartphone, and smart education. Therefore, the definition of "smart" is relative to the context. According to Spector (2016), the use of the term "smart person" could refer to someone who knows how to do something difficult and unusual. In the context of education, smart learning is a system that enhances the traditional method of learning (Gros, 2016). Hence, smart learning environments aim to supplement teaching and learning through state-of-the-art technology as an enabler of the provision of an enhanced learning experience. According to Gros (2016), the concept of smart learning environments emerged from two different types of technology, namely smart devices and intelligent technologies. However, smart learning environments are not necessarily focused on technology alone but also on the processes and approaches adopted to foster teaching and learning.

Intelligent learning environments have existed ever since technology began to enhance learning through the classical standalone desktop applications and later e-learning services (Kim, et al., 2011). Thus, the journey toward a smart learning environment may have started with e-learning platforms where web 2.0 was leveraged to deploy learning contents (Lwoga, 2012). Moreover, the advancement of technology has caused a transition in educational learning environments from the electronic environment to the mobile environment, and now to the smart environment (Taisiya, et al., 2013). On the other hand, learning approaches have also seen a transition from web-based learning to wireless mobile-based learning and context-aware ubiquitous learning (Agbo & Oyelere, 2019), and recently to socially aware learning (Liu & Hwang, 2010). These transitions have been characterized by different features, technologies, designs, and architectures. According to Yeonjeong's (2011) classification of learning environments, the transition started with electronic learning and stopped at ubiquitous context-aware learning, which was perhaps the latest learning environment at the time of their study.

However, learning environments have continued to evolve to provide a 21st-century learning experience (Gros & García-Peñalvo, 2016; Abtar & Hassan, 2017). Therefore, this dissertation presents a transition of technology-enhanced learning environments from electronic learning to smart learning, as depicted in Figure 5. This transition builds upon previous studies (Yeonjeong, 2011; Abtar & Hassan, 2017) to classify learning environments into e-learning (electronic learning), m-learning (mobile learning), u-learning (ubiquitous learning), and s-learning (smart learning).

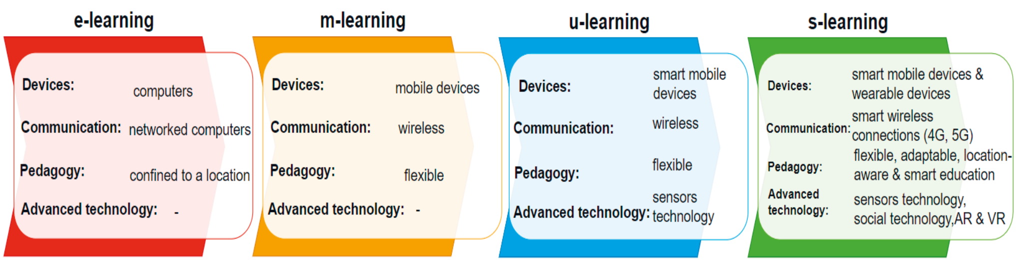


Figure 5. Transition of technology-enhanced learning environments.

In the transition flow, each learning environment has four components, classified as devices, communication, pedagogy, and advanced technology. For example, in the s-learning era, the use of wearable devices constitutes an improvement of the device component. Similarly, smart wireless communication such as the fourth- and fifth-generation (4G and 5G) Internet infrastructure remains an advancement over the previous communication infrastructure. In addition, the integration of smart education, adaptive learning, and location-aware learning has been the new pedagogical approach deployed in the smart learning era (Hwang & Fu, 2020; García-Peñalvo, et al., 2020). Finally, for the advanced technology component, augmented reality (AR) and VR are state-of-the-art social technologies that are currently deployed for smart learning environments.

Agbo et al. (2021a) demonstrated that the field of smart learning began in early 2000; however, it appears that the field started to witness more scientific output from 2012, when Huang et al. (2012) proposed the concept of the smart learning environment as the highest level of the digital environment for learning systems. However, other learning systems whose features are related to the smart learning environment existed before the 2000s. For example, intelligent learning systems (Brusilovsky, et al., 1996; Cheung, et al., 2003), adaptive learning environments (Brusilovsky, et al., 1995), and artificial intelligence in education (Lawler & Yazdani, 1987) were developed to foster learning in different contexts. Unfortunately, these systems lack recent smart features that can provide a high-level learning experience since they were mainly deployed on the World Wide Web (www) and are based on desktop computers (Brusilovsky, et al., 1996).

Over the last two decades, scholars such as Hwang (2014), Kinshuk et al. (2016), and Spector (2014) have provided tremendous contributions to the field of smart learning environments, as reflected in the annual article production figures presented in Figure 6. Moreover, several research topics on smart learning have emerged recently (Molina-Carmona, et al., 2020; Agbo, et al., 2021a).

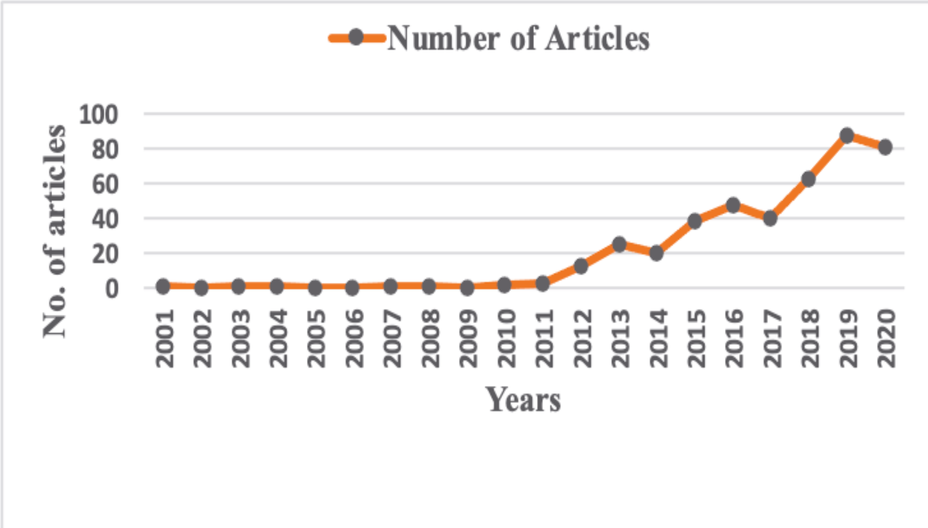


Figure 6. Annual scientific production of articles on smart learning environments from 2001 to 2020 (Source: Agbo et al., 2021a).

What is a smart learning environment? The field of smart learning environments is still maturing, and researchers are yet to arrive at a consensus on their definition (Abtar & Hassan, 2017). As already established at the beginning of this section, the term "smart" can have an expansive application, which could create difficulty in framing a definition that can be unanimously accepted by scholars. However, several studies (Hwang, 2014; Spector, 2014; Abtar & Hassan, 2017; Yassine, et al., 2016) have attempted to define a smart learning environment by connecting the application of technology to enhance teaching and learning experience, create flexible environments, and deliver efficient content. According to Spector (2014), a smart learning environment is an adaptive technology designed to include innovative features and capabilities that improve understanding and performance. The innovations, as stressed by Spector (2014), include features that make a smart learning environment adaptive, context-aware, and motivating for learners.

Similarly, Yassine et al. (2016) defined a smart learning environment as a technology-enhanced learning environment that integrates the criteria and functions of intelligent learning systems and context-aware ubiquitous learning. The intelligent feature of the smart learning environment includes learning analytics and learners' performance evaluation functionalities. Hwang (2014) defined a smart learning environment as a technology-supported learning environment that adapts and provides appropriate support. The supporting features include guidance, feedback, hints, and tools in the right places and at the right time based on individual learners' needs. These needs might be determined by analyzing their learning behaviors and performance as well as the online and real-world contexts in which they are situated. Thus, a smart learning environment is a form of an intelligent, personalized learning system that can be integrated with a variety of interactive devices (Ronghuai, et al., 2017).

From the array of definitions by different authors, the present author attempted to define a smart learning environment by bringing together all of the components that comprise a smart learning ecosystem (Figure 7). This was done because they all play a significant role in the development of a learning environment for the 21st-century learning experience. In sum, a smart learning environment is expected to be context-aware and ubiquitous, to leverage social technologies, to be highly interactive through sensor-enabled and wireless communication, and to be learner-centered by rendering supports based on users' learning needs. Hence, this dissertation defined a smart learning environment as

> *a technology-enabled, learner-centered, pedagogy-flexible, and context-aware learning environment that provides high levels of immersion, interaction, personalization, and engagement through intelligent feedback based on learners' characteristics and learning needs.*

This definition suggests that technology is an enabler that mediates between the learner and the learning content, whereas pedagogy remains the channel that provides high-level interaction for an immersive learning situation. Based on this definition, the dissertation sought to ensure that all the components that form the smart learning ecosystem (García-Holgado & García-Peñalvo, 2019) were explored to design and develop a solution that facilitates computational thinking knowledge. In an attempt to develop students' computational thinking knowledge, problem-solving skills, and cognitive engagement, a smart learning environment is integrated with multiple intelligences, approaches, and learning styles (Cheung, et al., 2020; Vakaloudi, 2020).

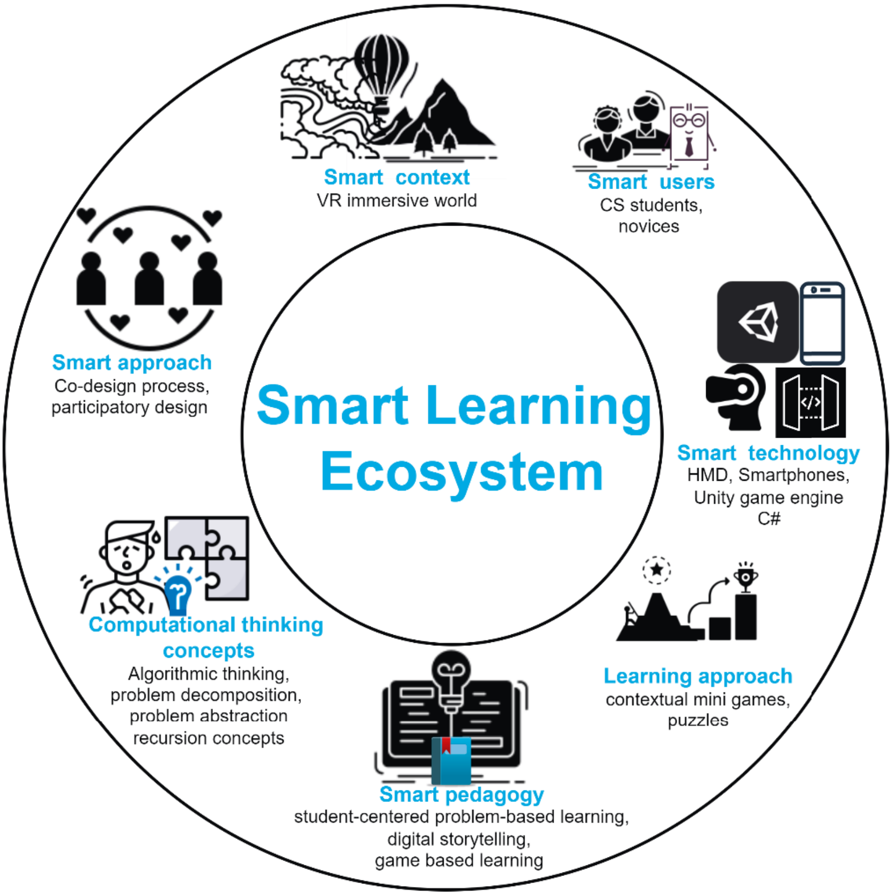


Figure 7. Conceptualized definition of a smart learning ecosystem.

From the curriculum perspective, one study demonstrated that relevant frameworks are being proposed for designing student-centered smart learning environments to foster the understanding of complex problems and innovativeness, thus preparing undergraduates for Industry 4.0 workplaces (Borg, et al., 2019).

Nowadays, there is a paradigm shift of technology-mediated education to a smart learning environment in order to provide relevant knowledge for learners’ job needs of Industry 4.0 (Han & Kim, 2017). Computational thinking in smart learning environments could provide relevant skills that students need for their future jobs. A recent study examined the relationship and effectiveness of computational thinking in a smart learning environment (Han & Kim, 2018). According to Han and Kim (2018), the result of educational effectiveness with students who experienced computational thinking in a smart learning environment was positive compared with the traditional learning environment. Against this backdrop, several studies have developed smart learning interventions to support computational thinking in higher education settings.

Gunn and Raven (2017) introduced a smart learning environment in an engineering school to foster technology-enhanced active learning in the classroom. According to the authors, the introduction of smart learning technologies in a university engineering classroom for teaching engineering courses related to problem-solving was a great success. Therefore, their intervention was mainly focused on the technology component of the smart learning ecosystem for problem-solving skills. Similarly, Srivastava (2016) proposed two prototypes of smart learning environments: (i) a mobile-based augmented reality application to support students’ understanding in lab works, and (ii) an intelligent breadboard that was capable of sensing errors made by students during lab work. According to the author, the proposed smart technologies mimic the physical laboratories in the electronic engineering curriculum to allow students to gain a "hands-on" learning experience and to build their problem-solving skills. In addition, Seo and Kim (2016) proposed an instructional model for collaborative problem-solving based on smart learning concepts. The authors demonstrated how to co-design a smart learning environment with users as one of the key components of the smart learning ecosystem. Furthermore, Sooncharoen et al. (2020) proposed a learning tool to support the teaching and learning of topics such as scheduling. The authors embedded a computational intelligence algorithm called Teaching–Learning-Based Optimization (TLBO) to facilitate the solving of scheduling problems (Sooncharoen, et al., 2020).

While all of the aforementioned studies have developed smart learning environments focused more on one or two components of the smart learning ecosystem, this dissertation attempts to bring all of the components in Figure 7 together, harnessing them to develop a context-aware smart learning environment for introducing computational thinking in the Nigerian context.

2.4 Design methods and pedagogical perspectives relevant to the study

To design and develop a smart learning environment, this dissertation recognized the design methods, fundamental learning theories, and pedagogical principles that define the process. The design methods outline techniques and activities that engendered the creation and implementation of the artefact, while the learning theories and pedagogy facilitated the learning process. According to cognitive psychologists Jill and Carol (2004), learning involves the use of memory, motivation, and reflection. They considered learning an internal process such that the processing capacity of a learner, the amount of effort invested, and the existing knowledge structure determine the amount of learning. While many learning theories by philosophers exist, this research identified relevant theories and pedagogy to design a VR-based smart learning environment for facilitating computational thinking (Quevedo-Torrero, 2009; Article VI). This section begins by presenting the design methods explored in this dissertation and then demonstrates the pedagogical approach of this research. Later, it presents the relationships between the design methods, pedagogy, and learning theories considered in this study.

2.4.1 Co-design and participatory design principles for facilitating computational thinking

Co-design has recently become a popular approach for designing educational mini-games with stakeholders (De Jans, et al., 2017). According to Vaajakallio and Mattelmäki, co-design is regarded as one of the most powerful ways to elicit reactions from co-designers in a participatory study (Vaajakallio & Mattelmäki, 2014). Participatory design deals with a methodological approach that ensures that users of a technological intervention are involved in the entire design process of conceptualizing, designing, and implementing the artefact to create a more efficient and usable system (Gomez, et al., 2018; Simonsen & Robertson, 3012). According to Spinuzzi, participatory design started in Scandinavia and has spread globally to become one of the most widely used research approaches in computer science (Spinuzzi, 2005). Today, participatory design seems to be gaining more ground in the educational research domain, particularly in computer science. Indeed, one study referred to participatory design as theory (Devisch et al., 2019) instead of a design approach as is widely known. Moreover, several studies have demonstrated a participatory co-design approach for developing educational tools. Table 3 presents some studies on computing education that have applied a participatory co-design approach to develop interventions for teaching computational thinking concepts.

Table 3. Studies that have demonstrated a co-design approach for developing computational thinking interventions.

Authors	**Aspect of computing education**	**Research focus**
(Rich et al., 2020)	Computational thinking concepts	This study conducted an analysis of how eight elementary teachers created computational thinking lessons through participatory design to engage their students during unplugged mathematics and science activities.
(Tsortanidou et al., 2019)	Computational thinking concepts	This study demonstrated a collaborative design of a pedagogical model that promoted creativity, problem-solving, and general computational thinking skills among students.

(Basu et al., 2020)	Computer science-aligned curriculum	This study designed a curriculum-neutral assessment based on a data and analysis concept from a computer science framework using a participatory design approach.
(Dindler et al., 2020)	Digital literacy and computational thinking	This study focused on the development of computational thinking empowerment and digital literacy by using a participatory design process.
(Wu et al., 2020)	STEM education and computational thinking	This study conducted a workshop to demonstrate participatory co-design of a STEM curriculum with teachers to integrate computational thinking in their classes.
(Peel et al., 2020)	Computational thinking and science curriculum	This paper presented a case study of a science teacher who implemented design-based research over three years to integrate computational thinking into the science curriculum. This design process was mediated by a co-design approach to support teachers in collaborative curriculum design.
(Vartiainen et al., 2020)	Computational thinking and machine learning	This theoretical paper dwelt on tensions that educators could encounter when attempting to bridge participatory learning with machine learning and algorithmic thinking.
(Motschnig et al., 2018	Computational thinking and digital competencies	This study demonstrated a workshop-based design thinking method for children by adopting participatory action research to teach computational thinking to the children.
(Karumbaiah et al., 2019)	Computational thinking and teachers' professional development	This paper reported on how in-service teachers were developing their profession through a participatory co-design approach.
(Bonani et al., 2017)	Computational thinking	This paper designed a prototype for teaching graph algorithmic thinking using a

		participatory action-based research approach.
(Brady et al., 2016)	Computational thinking and computing education	This study demonstrated a series of activities such as participatory simulations and computing projects that foregrounded the social and collaborative aspects of computer science. The aim was to create interest in computational thinking among students.

Since the overarching aim of this study was to co-design a smart learning environment to facilitate student-centered learning, it was necessary to exploit the participatory co-design process with students in the Nigerian context to design and develop the artefact. Ideas from existing literature were integrated into the OCD process presented in Section 3.2.

This dissertation also explored VR technology as one of the design approaches to simulate a real-world situation of context and concepts that would allow learners to experience concrete learning objectives through visualization in a virtual environment (Pellas & Vosinakis, 2018).

2.4.2 Constructivism and experiential learning theories

Constructivism is a theory that states that learners create their own mental representation of learning objectives through an active and constructive process (Jong, et al., 2010; Zhu, 2008). In other words, constructivism postulates that learners interpret information according to their personal reality; they learn by observation, processing, and interpretation, and then personalize the information (Ben-Ari, 1998). Scholars have expressed positive support for the use of constructivism theory in developing technological interventions in education; however, such interventions can only be successful if they incorporate the roles of the teacher, curriculum, and society (Elkind, 2004).

In the context of computer science education research, constructivism was not a commonly used theory among scholars approximately two decades ago (Ben-Ari, 2001); however, plenty of studies have reported the use of constructivism in recent years (Guo, 2018; Bakar et al., 2019; Yakar et al., 2020). Furthermore, the use of constructivist philosophy in VR-based educational interventions has recently been gaining ground (Lee & Shea, 2020; Seo et al., 2021). Therefore, in a smart learning environment, learners can learn best when they can contextualize what they learn, both for immediate application and to acquire personal meaning. In this dissertation, constructivism learning theory is connected to game-based learning, which is the pedagogical approach explored in this research. Figure 8 presents the relationships between the theories, design methods, and pedagogy considered in this dissertation to design and develop a smart learning environment for facilitating computational thinking knowledge.

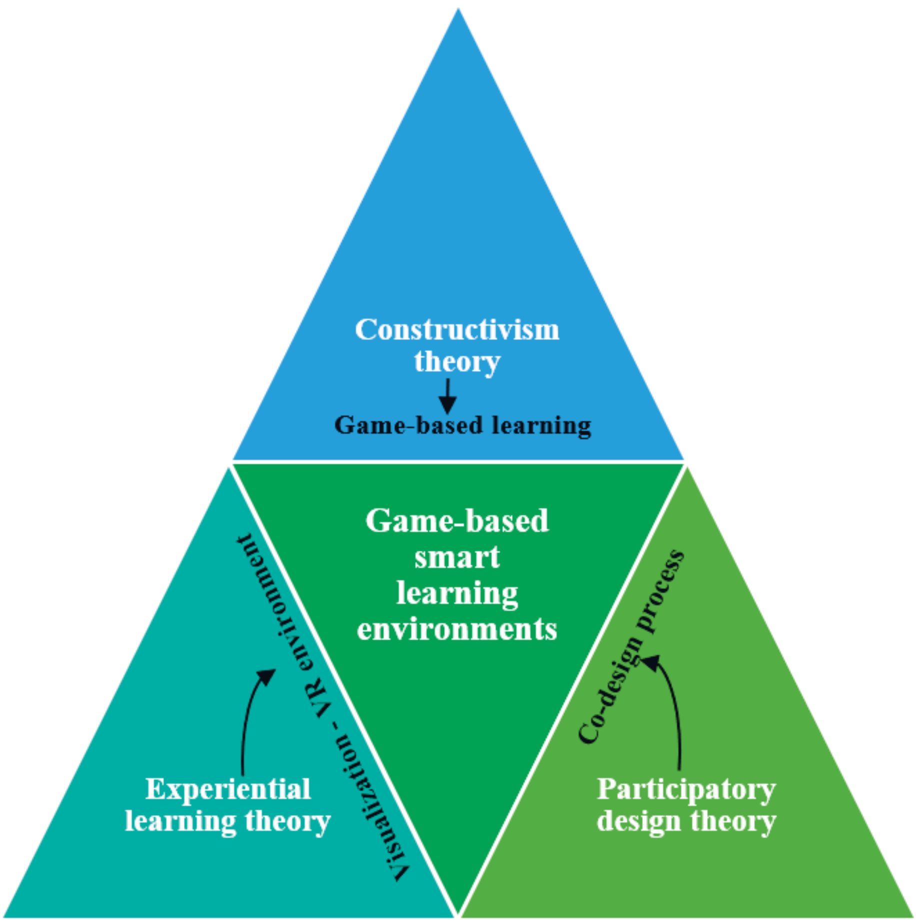


Figure 8. Interrelationships of relevant theories, pedagogy, and methods for designing a smart learning environment.

Experiential learning theory, on the other hand, holds the epistemological perspective that learning is a process whereby concepts are derived from and are continuously modified by experience; thus, experience plays a central role in the learning process (Kolb, 2014). According to Kolb, "ideas are not fixed and immutable elements of thoughts but are formed and re-formed through experience" (p. 26).

Similarly, Yardley et al. described experiential learning as "constructing knowledge and meaning from real-life experience" (Yardley, et al., 2012, p. 161). Recent studies have demonstrated how experiential learning can facilitate lab experiments (Pfeiffer et al., 2020), fields trips, and outdoor lessons (Cotic et al., 2020), as well as other learning contexts. In computing education, for example, El-Glaly et al. (2020) demonstrated how to promote accessibility education among students by using experiential learning theory. This dissertation also found experiential learning theory relevant because learners developed their understanding of computational thinking concepts by experiencing the process of creating mini-games through a collaborative co-design process. These mini-games were further modelled to be played in a virtual environment to provide rich visualization of the learning objectives. The relationship of learning theories explored in this research were reported in Article VI.

2.4.3 Educational game-based learning for computational thinking

Game-based learning (GBL) is one of the approaches that educators and educational technologists have applauded to provide motivation, engagement, and cognitive benefits to learners (Oyelere et al., 2019a; Seidel, et al., 2019; Hooshyar, et al., 2020). According to Plass et al., GBL "is a type of gameplay with defined learning outcomes" (Plass, et al., 2015, p. 259). Regarding computing education, some scholars have researched educational games as one way to present computational thinking and problem-solving skills to young learners (Seidel, et al., 2019). Using game techniques to supplement the traditional teaching method can facilitate students' understanding, enhance their learning experience, and improve their performance (Arrington Jr, et al., 2011). Types of educational games were presented in Article VI. Figure 9 presents the operational definition of an educational game and classifies games mainly into three types: video games, serious games, and mini-games (Devisch, et al., 2017). Mini-games are defined as small and simple games that can exist within a larger game and can be independently played (Van Borkulo et al., 2011).

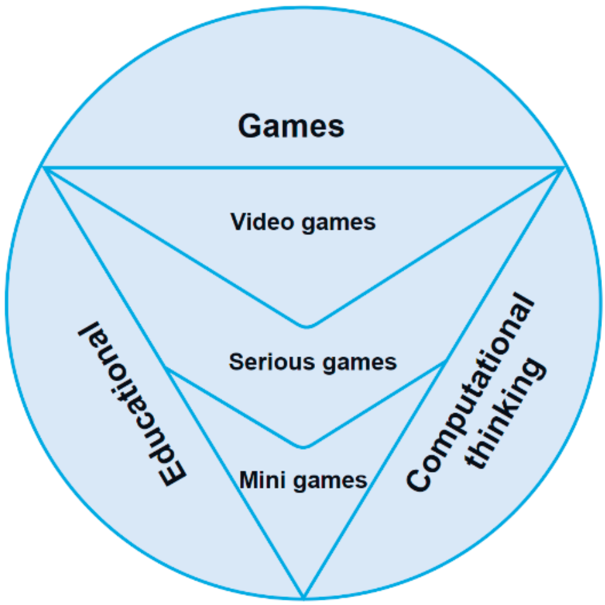


Figure 9. Classification of educational game-based interventions (Adapted from Article VI).

A recent study designed and developed an adaptive computer game named Autothinking to facilitate students' computational thinking skills (Hooshyar, et al., 2021). In Nigeria, GBL is also gaining grounds, such as the contextual board game Ayo developed by Oyelere et al. (2017) to facilitate students' understanding of programming concepts. Similarly, several studies have demonstrated virtual game-based applications to support teaching and learning in various educational disciplines (Kavanagh, et al., 2017; Radianti, et al., 2020). For example, Butt et al. (2010) developed a VR game-based application to aid in the training of nursing students. While some studies have argued that virtual game-based applications have great potential to support learning (Zhang, et al., 2020), their negative effects remain a topical issue (Pack, et al., 2020). Nevertheless, the huge benefits of deploying educational games in a virtual environment to enhance students' learning and engagement are evident (Article IV). Hence, this study attempted to leverage GBL by developing mini-games capable of enhancing students' computational thinking knowledge in an immersive virtual learning environment.

2.5 Application of virtual reality mini-games in computing education

This section presents an overview of how VR mini-games have been applied in computing education. The application of mini-games as a pedagogical component of learning approaches in classrooms has only recently been researched. Although not many studies on educational mini-games have been conducted, scholars are increasingly emphasizing the importance of using mini-games in classrooms (Savage, 2015; Horst, et al., 2019). For example, Li et al. (2021) developed a mini-game-based flipped classroom based on cognitive learning theory to demonstrate how the approach can improve the learning performance of students compared with the traditional video-oriented flipped classroom. Their results indicated that the mini-game-based flipped classroom facilitated students' learning more than the traditional classroom. Earlier, Van Borkulo et al. (2011) designed a mini-game to support students in learning mathematics basics. They described mini-games as consisting of "small, focused activities, which students play for fun and in which they are engaged for a short period of time" (p. 61) to learn basic skills.

Today, educational mini-games are developed as VR applications that provide full immersion, interaction, and engagement (Parong & Mayer, 2020; Article IV). As revealed by Radianti et al. (2020), a considerable number of VR studies have focused on educational mini-games for computing education. While some studies have deployed learning content that focused on enhancing computational thinking and problem-solving skills (Parmar, et al., 2016; Pellas & Vosinakis, 2018), others have centered on the learning of programming concepts such as object-oriented programming (OOP; Stigall & Sharma, 2018; Bouali, et al., 2019). For example, Parmar et al. (2016) developed a VR game-based application called VEnvI to support computer science students in learning about the basics of computer science concepts such as sequences, loops, variables, conditionals, functions, and parallel programming. Furthermore, Pellas and Vosinakis (2018) designed a 3D simulation mini-game to support students' problem-solving skills by using an open simulator with Scratch4SL. In addition, Stingall and Sharma (2018) developed VR instructional (VRI) modules to teach computer science undergraduate students the OOP concept. Moreover, Bouali et al. (2019) implemented a VR mini-game called Imikode to be played within and outside the classroom to teach students basic OOP concepts by creating objects such as houses, trees, and animals within a virtual environment. Recently, Segura et al. (2020) developed a VR application called VR-OCKS to support computer science and engineering students in learning basic programming control structures such as iteration and conditional selection. Table 4 presents several VR applications and interventions developed to support computing education, which motivated the development of the iThinkSmart VR prototype application to facilitate computational thinking in the present research.

Table 4. Virtual reality applications for computing education.

VR application	**Targeted users**	**Short description of the study goal**
VR mobile game (Maze Game; Srimadhaven et al., 2020)	Computer science and engineering students	This study focused on experimenting with a VR mobile game to assess the cognitive level of students on a Python course.
Imikode (Bouali et al., 2019)	Computer science novices	This study presented a VR game to teach novices about object-oriented programming concepts.

Alchemist Escape (Bolivar et al., 2019)	Computer science students	This study developed a VR video game called Alchemist Escape to provide students with the basics of computer science to increase their awareness.
Virtual environment interaction (VEnvI; Parmar et al., 2016)	Middle school students	The authors developed VEnvI to support students' knowledge of computer science concepts such as sequences, loops, variables, conditionals, functions, and parallel programming.
Dr. Chestr virtual human (Arrington Jr et al., 2011)	Middle school students	This study designed and implemented Dr. Chestr, a virtual human in a VR environment, to support the understanding and retention of introductory programming courses.
Virtual puppet (Vanderdonckt & Vatavu, 2020)	Computer science students	This study presented a VR application where a virtual puppet displayed human-like emotions for the benefit of children.
MYR: A web-based virtual reality application (Berns et al., 2019)	Middle school students	The VR educational platform MYR was built to spark students' interest in computer science by allowing them to write code that generates 3D animated scenes in a virtual world.
VRASP (Nguyen et al., 2020)	Middle school students	A VR programming environment (VRASP) was developed to allow students to produce an avatar (agent) in a virtual world that could answer questions in spoken natural language.
3D Virtual Programming Language (3D-VPL; Ortega et al., 2017)	Computer science students	The study developed a 3D-VPL to provide an interactive tool for beginners and intermediate computer science students to learn programming concepts.
FunPlogs (Horst et al., 2019)	Computer science students	The authors introduced FunPlogs, a serious puzzle mini-game for learning fundamental programming principles.

3D Simulation Game (Pellas & Vosinakis, 2018)	High school students	The authors explored a 3D simulation game to teach problem-solving among students using an open simulator with Scratch4SL.
VR Instructional (VRI) modules (Sharma & Ossuetta, 2017)	Computer science students	The authors developed a VRI module for teaching loops and arrays to provide a better understanding of the concepts.
CodeSpells (Esper et al., 2014)	Middle school students	This study introduced CodeSpells, a 3D immersive video game to teach students introductory computing concepts.
VR-OCKS (Segura et al., 2020)	Computer science and engineering students	This study presented the design and development of a VR application to teach the basic concepts of programming.
Cubely (Vincur et al., 2017)	Programmer novices	This study demonstrated an immersive VR environment where novice programmers solve puzzles to understand programming concepts.
GTI module (Stigall & Sharma, 2018)	Computer science majors	The theme-based instructional (GTI) module was a VR game developed to teach undergraduate computer science students about stacks and queues.

All of the tools and interventions presented in Table 4 have demonstrated the relevance of VR mini-games in computing education, which align with the report presented in Article IV. It can be deduced that these tools were particularly focused on supporting students in gaining computational thinking and problem-solving skills. Notably, their goal was to enhance computer science students' knowledge of programming concepts. For example, some of the outcomes of these studies indicated that students were motivated to learn programming concepts (Ortega, et al., 2017; Srimadhaven, et al., 2020), whereas other studies sparked learners' interest in computer science (Berns, et al., 2019; Bolivar, et al., 2019). However, specific concepts of computational thinking such as algorithmic thinking, problem decomposition, and recursive thinking have scarcely been covered as part of the learning objectives of these tools. For this dissertation, a study was conducted to develop a VR application to support students to think algorithmically, gain knowledge on problem decomposition, and think recursively to provide solutions to common and real-life problems. As an enabler, state-of-the-art VR technology was employed to provide immersive learning (Wu, Yu & Gu, 2020), a personalized learning experience (Horváth, 2020), a high level of interaction (Hudson, et al., 2019), motivation, and engagement (Chao, 2020).

2.6 Summary

This chapter has presented the literature review of the relevant concepts discussed in this dissertation. The chapter began by providing a contextual overview of computing education as the background for introducing other concepts. It magnified the concept of smart learning and how learning environments mediated by technology have evolved, and then provided the operational definition of a smart learning environment. Furthermore, the theoretical background for this dissertation was introduced. In other words, the learning theories that provide the foundation for the study were established (Radianti, et al., 2020). In addition, this chapter provided an overview of relevant components of smart learning, including design methods such as the participatory co-design process, pedagogical approaches such as GBL, and technology such as VR.

3 Research design and methodology

This dissertation employed several research methods, including mixed methods, qualitative and quantitative techniques, participatory co-design, and software prototyping. For the data collection, interviews, questionnaires, note-taking, and observation techniques were employed. These methods were deployed during the various stages of the DSR process. This chapter details the methodological process followed in this dissertation for co-designing and developing a VR-based smart learning environment with students to facilitate the understanding of computational thinking concepts. In particular, this chapter demonstrates how an OCD process was infused with the DSR framework to design and implement a smart learning environment for supporting computational thinking knowledge.

3.1 Design science research and co-design process

3.1.1 Design science research

DSR is a pragmatic research method that seeks to create new artefacts to address an identified human problem domain (Hevner et al., 2004). In a simple definition, Gregor and Hevner referred to DSR as a problem-solving process where knowledge of a design problem and its solution are acquired in the building of an artefact (Gregor & Hevner, 2013). Scholars have demonstrated other similar research approaches including action research (Lau, 1997), design experiments (Cobb et al., 2003), design thinking (Meinel & Leifer, 2012), and design-based research (Anderson & Shattuck, 2012) for finding solutions to address practical problems and advancing scientific knowledge. Indeed, DSR has also become a popular research approach in information systems research (Rossi, et al., 2013; Weber, 2010), engineering (Carstensen & Bernhard, 2016; Apiola & Sutinen, 2021), and computer science (Naidoo, 2012; Oyelere, 2017).

While all of the aforementioned design methods and more have similar characteristics that seek to create an intervention for addressing a contextual problem, DSR provides more flexibility in terms of the process and activities that lead to the development of the intervention. In fact, a recent study by Holopainen et al. (2020, p. 1) described the DSR methodology as the "most common and heavily referenced methodology in the context of innovation and development". While DSR has penetrated various disciplines as a pragmatic approach for conducting innovative research that seeks to solve practical problems, its application and relevance in computing research have become evident (Naidoo, 2012). In addition, DSR allows for advancing scientific knowledge through flexible approaches such as communicating research outputs in conferences and journals and formulating principles to contribute to theory or models. The goal of DSR is to create common grounds where academic researchers and industrial practitioners can design and develop interventions that provide a meaningful impact on society through a rigorous process (Holopainen, et al., 2020).

Recently, DSR has been used to design and develop information system interventions to address contextual problems facing users in developing countries. For example, Maphosa et al. (2021) designed a mobile application to enhance teaching and learning in a rural university in Zimbabwe following the DSR methodology. Their study revealed that the mobile application was capable of bridging the gap between the rural–urban dichotomy in terms of access to learning. Similarly, Paulmani (2020) developed an online educational platform for a Research Methods (RM) course taught at a higher education institution to facilitate the understanding of RM fundamentals among young researchers. The study followed the DSR methodology and was aimed at providing a solution for how the RM course is taught through visualization in the developing countries context. In another study, Montero and Kapinga (2019) implemented a mobile application to support women entrepreneurs from a rural area of Tanzania through the DSR method strengthened by a co-design process with stakeholders.

In the Nigerian context, Oyelere and Suhonen (2016) as well as Oyelere (2017) explored the DSR methodology to design and develop a mobile application for facilitating computer science education at higher education institutions. According to Oyelere, the contextual mobile application based on DSR demonstrated the potential of mobile learning to improve students' learning achievement in Nigeria. The outcomes of studies that have demonstrated the DSR methodology for developing artefacts for learning in the Nigerian context provide empirical evidence to support the rationale for adopting DSR in this dissertation. Furthermore, DSR is a legitimate and pragmatic approach suitable for contextual studies, as revealed by Apiola and Sutinen (2021), who used DSR to develop a framework for computer science students to gain computational thinking competency.

According to Baskerville et al. (2018), DSR projects provide two main contributions: the design artefact and the design theory or guiding principles. These two contributions suggest that the artefact addresses practical problems while the theory represents scientific rigor, which contributes to the research knowledge base. Other scholars argued that the theoretical contributions to the knowledge base can be in the form of models and guidelines that are communicated through academic publications in journals, books, or conferences (Johannesson & Perjons, 2014). As Baskerville et al. asserted (2018), the artefact takes the focal point of information technology projects that exploit DSR; however, the methodological theories, models, and guidelines that define the rigor of such a project must not be undermined. Therefore, Johannesson and Perjons (2012) demonstrated a practical framework that guides researchers in conducting and applying DSR in a project.

Several characteristics of DSR make it suitable in research that addresses contextual problems (Oyelere, 2017). For example, DSR provides a unique way of identifying specific problems that require a systematic approach to their solution through several strategies and methods. DSR allows for an incremental but also iterative process of planning, developing, and evaluating artefacts. Because of the iterative nature of the DSR method, it is possible to feed outputs from one DSR activity into another as the input. Furthermore, the possibility for iteration allows interactions to occur between DSR activities and makes it possible to conduct multiple circles of DSR, as demonstrated by Hevner and Chatterjee (2010). In addition, DSR allows the researcher to use any appropriate research methods deemed relevant to collect and analyse the data and present the findings.

Scholars have provided guidelines and frameworks that outline the process of how DSR should be applied in a contextual study (Van der Merwe et al., 2019). For example, Offermann et al. (2009) summarized several design processes found in the literature into three, namely (i) problem identification, (ii) solution design, and (iii) evaluation. Similarly, Peffers et al. (2007) developed a process model that outlined the DSR cycle through (i) problem identification and motivation, (ii) objectives of the solution, (iii) design and development, (iv) demonstration, and (v) evaluation and communication. Similar to Peffers et al. (2007)'s model, Johannesson and Perjons (2012) provided a framework that was broken down into five activities: (i) problem explication, (ii) requirement definition, (ii) design and development of an artefact, (iv) demonstration of the artefact, and (v) evaluation of the artefact. The author of this dissertation studied these different DSR frameworks and found Johannesson and Perjons' (2012) version to be more explicit and comprehensible to adopt. Their DSR framework does not emphasize the iteration that occurs in DSR compared with other frameworks such as that of Hevner and Chatterjee (2010); however, a crucial feature of DSR is the iterative nature and interaction between the activities. For the design and development process of the smart learning environment, the present study infused an iterative co-design process into the DSR framework of Johannesson and Perjons to strengthen the research rigor, as illustrated in Figure 10. From this point on, Johannesson and Perjons' DSR framework is referred to as the JPDSR framework for short. The JPDSR framework consists of five activities that are briefly explained in the following paragraphs.

Problem explication activity: The task of identifying and developing a relevant problem that can be researched using DSR can be challenging; hence, a structural problem formulation approach is important (Purao, 2021). The major activity during problem explication is to examine what practical problem the DSR method must solve (Gregor & Hevner, 2013), whether the problem is significant in terms of impact, and if it is well-formulated and motivated to address a real-world situation (Johannesson & Perjons 2014). A contextual explication of the problem would unravel the dependencies that can make a problem solvable through the DSR method.

Requirement definition activity: The requirement definition of a DSR project outlines a set of solutions in terms of what needs to be done to provide a solution to the explicated problem. In other words, the requirement definition activity primarily focuses on determining the functionalities of the artefact. As demonstrated by De Silva et al. (2014), the requirement definition activities of a DSR project that aims to design an artefact can involve multiple techniques, such as traditional surveys, interviews, identification scenarios, and the creation of functional prototypes with users.

Design and development activity: The design and development activity of a DSR project aims to create an artefact that is measured against the set of requirements in order to solve the explicated problem. This activity entails the creative construction of computer instructions using a programming language(s) (Holopainen, et al., 2020). Creativity is acknowledged as a crucial element in the design and development of an artefact using the DSR approach (Baskerville et al., 2019). The design and development activities attempt to demonstrate creative concepts that can be implemented to address the identified problem.

Demonstration of artefact activity: The demonstration activity entails testing the developed artefact to prove whether it serves the purpose or can solve an instance of the explicated problem (Johannesson & Perjons 2014; Tommelein, 2020). Demonstrating the artefact is critical as it showcases transparency in the implementation process of the DSR framework (Piirainen & Briggs, 2011).

Evaluation of artefact activity: The last DSR activity involves the evaluation of the artefact. Concrete testing with stakeholders can be conducted to determine the extent to which the developed artefact can solve the explicated problem. As acknowledged by scholars, evaluating an artefact designed based on DSR is a crucial step that contributes to the rigor of the DSR process (Venable et al., 2012).

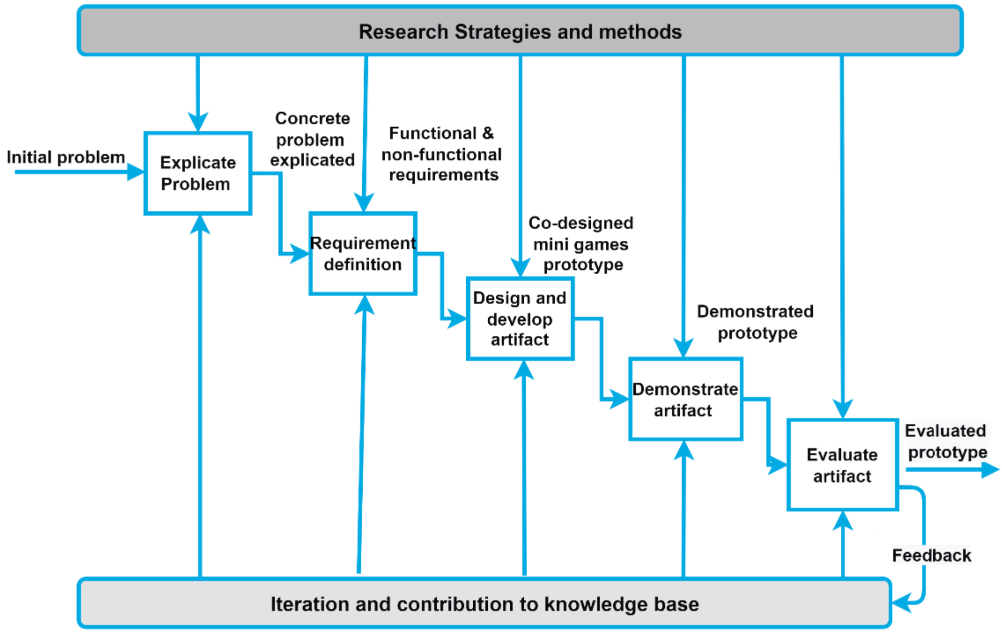


Figure 10. Diagram of the design science framework (adapted from Johannesson & Perjons, 2012).

In each of the JPDSR activity stages, any suitable research method can be used, such as qualitative, quantitative, or mixed-methods, which is why DSR is a pragmatic approach. For example, the interview method can be used to explicate the problem; a literature review and interview methods can be used to determine the requirements; while formative evaluation can be applied to evaluate the artefact (Johannesson & Perjons 2014).

These activities are iterative in nature, and even the intervention that emerges from the first circle of the DSR method can be fed back into any activities for another loop (Hevner & Chatterjee, 2010; Apiola & Sutinen, 2021). Sections 3.2–3.4 discuss in detail how this dissertation demonstrates each activity of the JPDSR framework and reports concrete findings based on my experience.

3.1.2 Online co-design process

OCD provides an opportunity to design a student-centered artefact with stakeholders in a participatory manner in an online environment. This practice can be synchronous participatory design (Lee et al., 2021). The practice of OCD is not common, especially prior to the recent COVID-19 pandemic. Previously, user-centered design studies that required a co-design process with subjects were often conducted face to face (Hjelmfors et al. 2018; Jessen et al. 2018; de los Ríos et al., 2019). Walsh et al. (2012) conducted one of the earliest studies that performed OCD activities with students using a desktop-based tool, which allowed subjects to collaborate from multiple locations. The authors adopted an asynchronous approach even though their co-designs were conducted in an online environment.

Since the COVID-19 pandemic had prevented the world from conducting face-to-face activities including educational institutions (Daniel, 2020), this research innovatively adopted an improvisation where an OCD process became an alternative approach used to facilitate the design phase of the research. Aside from our OCD study in 2020 (Article VI), researchers have recently conceptualized different approaches to adapt to the 'new normal' of co-designing in an online environment by demonstrating similar approaches to those used for this dissertation. For example, Grover et al. (2020) recently demonstrated how a computer science curriculum to support teachers' professional development can be co-designed in an online environment. Kennedy et al. (2021) conceptualized a framework for translating co-design from face-to-face to online mode. Their approach was demonstrated by co-designing a project that aimed to design an intervention to prevent poor mental health in the Australian context. Similarly, Lee et al. (2021) investigated factors to be considered when co-designing with subjects in an online environment by conducting several workshops to design new interventions. The OCD process demonstrated in this dissertation provided activities that helped to facilitate computational thinking knowledge by allowing students to demonstrate creative thinking and problem-solving skills.

3.2 Problem explication and requirements analysis for developing a smart learning environment

This section presents the activities conducted in this research to explicate the problem behind the development of a smart learning environment as well as how the system requirements analysis was conducted. In this research, different strategies were deployed to implement the activities of the JPDSR framework. For example, we engaged stakeholders who explicated the contextual problem through interviews and then reviewed literature to determine the requirements.

3.2.1 Explicating the problem behind the development of a smart learning environment

Explicating a contextual problem requires a series of activities. In this dissertation, the problem explication activity employed mixed-methods research. Data were collected from students and teachers of computer science at a federal university and a state polytechnic located in Lokoja, Nigeria. Questionnaire and interview techniques were used to collect the data. In particular, the interview questions (Q1–Q6) and questions on the awareness and use of smart learning environments in Appendix 1 were the instruments that elicited the data used to explicate the initial problem. Both data collection items focused on the kind of problems the students encounter while learning programming in higher education and whether they have experience of using their smartphones to learn programming concepts. For example, one of the interview questions asked the students to share their experience of how they are being taught programming and whether the method enhances their understanding. Furthermore, the questions elicited students' expectations from the smart learning environment to address their learning challenges. The questions were developed by the author and two senior researchers, who independently validated the items. To further validate the questions, a professor of computer science who teaches at the federal university where the research was conducted also scrutinized the questions to ensure their suitability and validity. The author developed and validated the questionnaires using this method rather than adapting them from existing literature because this study needed to analyze the contextual problems explicitly at the beginning of the research, and adapted instruments may not be sufficient for explicating a contextual problem. The study presented in Article I recruited 197 students to participate, whereas the study presented in Article III recruited 210 students and 15 teachers. In Article III, 165 (83.8%) of the students who participated in the study were from the university, whereas 32 (16.2%) were from the polytechnic in Lokoja. While the study in Article I utilized a quantitative approach for data analysis, the study in Article III adopted both qualitative and quantitative methods to analyse and present its findings.

3.2.2 Defining the requirements for developing a smart learning environment

During the stage of defining the requirements, the researcher concretely outlines what kind of artefact can provide a solution to the explicated problem (Johannesson & Perjons, 2012). The requirement for developing a smart learning environment to facilitate computational thinking in the Nigerian context was outlined using different approaches (De Silva et al., 2014). First, the author conducted a study to elicit the expectation of users regarding what features of a smart learning environment should be considered when implementing the system. The study reported in Article III adopted both quantitative and qualitative approaches. A total of 210 students from two higher education institutions (Federal University Lokoja and Kogi Polytechnic Lokoja) located in the North Central region of Nigeria responded to our questionnaires. These institutions were selected for the study because the researcher lived in the region and taught in the same university during the study period; hence, the study participants were easily accessible. In addition, six students were randomly selected from among the participants for a focus-group interview. The instruments for the requirement definition consisted of the interview questions, open-ended questions, and Likert scale questionnaires on users' expectation of a smart learning environment, as provided in Appendix 1. As mentioned earlier, the study in Article III was conducted to explicate the contextual problem and elicit users' requirements. Therefore, as already explained in Section 3.2.1, the data collection process ensured that contextual data analysis was conducted at the beginning of the research.

Next, the author conducted a study (Article II) to investigate suitable approaches for providing smart learning environments to facilitate 21st-century learning in the higher education context based on existing literature. The study followed a systematic literature review methodology to obtain relevant articles, which were analyzed to gain insights into several aspects of system requirements, including the pedagogical component. The study gathered 161 articles from four databases, namely ACM, IEEE Xplore, ScienceDirect, and SpringerLink. After applying the inclusion and exclusion criteria, the study analysed 33 articles in several aspects, such as the programming teaching approach, tools, and context. Furthermore, the study in Article IV was conducted to uncover how state-of-the art technology such as VR has been utilized in computing education. This study adopted a systematic approach to review literature, while the data analysis consisted of bibliometric and content analysis. A total of 971 data items were collected from the Web of Science and Scopus databases and quantitatively analyzed, whereas content analysis was performed on 39 articles.

With this initial understanding of the requirements based on users' expectations and the existing literature, the author demonstrated a software requirement specification (SRS) following the software development process as the methodology for modeling the activities that would engender the design and prototyping of a VR-based smart learning environment, as presented in Article V. To accomplish the SRS, the Unified Modelling Language (UML) was utilized. UML is one of the formal notations that is commonly used as a standard object-oriented modelling language, allowing one to visualize, specify, and document the SRS (Wiegers & Beatty, 2013).

3.3 Design and implementation of the iThinkSmart prototype using online co-design

This section presents how participatory OCD was infused into the DSR framework with the aim of strengthening the design process and rigor of the DSR methodology (Hevner et al., 2004). In addition, this section explains how the participatory co-design process was conducted to design and develop the contextual mini-game prototypes to facilitate computational thinking skills. Our participatory OCD process was embedded in the JPDSR framework, as depicted in Figure 11, to serve as the driver for the development of the artefact through the rigor of design and the development circle of DSR. The participatory OCD process has five main stages, namely the planning, exploration, design, discussion, and evaluation stages. This research designed and developed a prototype by leveraging the OCD process circle. A previous project had demonstrated a co-creation and co-design process sandwiched between the design, development, and demonstration activities of DSR to implement a mobile marketing application that supports Tanzanian women entrepreneurs (Kapinga, et al., 2019).

The author of this dissertation believed that the participatory co-design of artefacts would not only give users a sense of ownership but also empower them to provide useful inputs for improving the conceptualization and prototyping of the artefact. A recent study alluded to the significance of co-designing games with adults and children (Havukainen, et al., 2020). The authors affirmed that co-designing with stakeholders is significant because it achieves higher-quality artefacts that provide user-centered functionalities (Havukainen, et al., 2020).

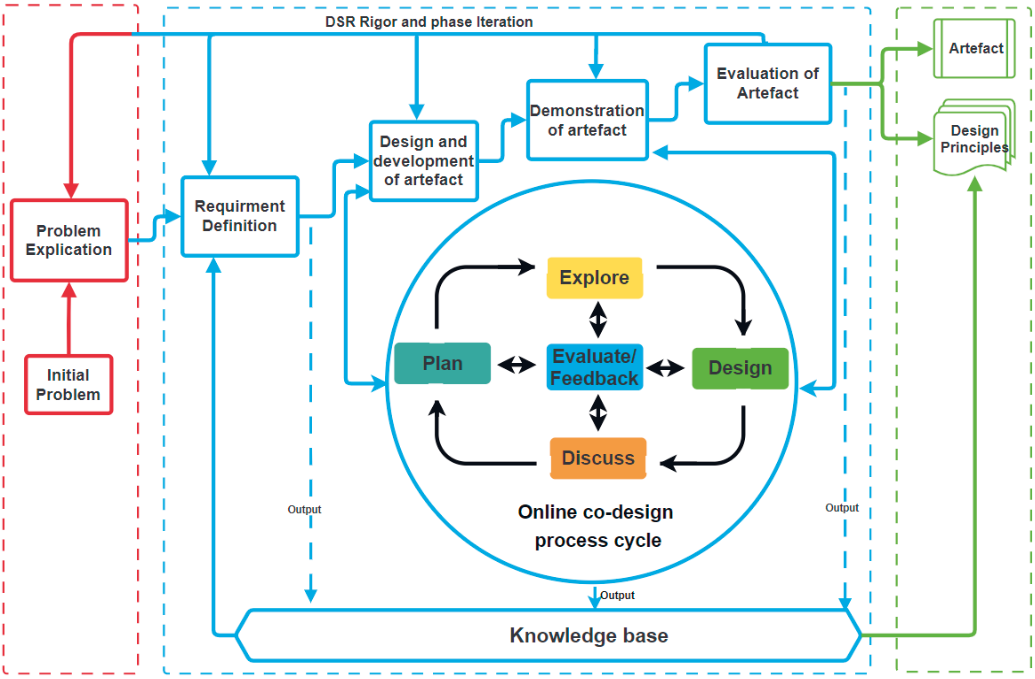


Figure 11. Design science research framework strengthened with the participatory co-design process.

Until recently, the co-design of user-centered educational artefacts was predominantly performed in a face-to-face mode. For example, Bonsignore et al. (2016) designed a game-based tool to assist young learners in learning the basics of scientific concepts by partnering with them through a face-to-face co-design process. Similarly, Hjelmfors et al. (2018) developed an intervention to improve communication about the heart failure trajectory and end-of-life care by engaging patients and family members in a face-to-face co-design process. Additionally, Havukainen et al. (2020) co-designed an educational game with older adults and children in a face-to-face method to demonstrate the intergenerational perspective of a user-centered study. While these co-designed studies in face-to-face mode reported positive outcomes, the contemporary situation during the research period of this dissertation did not allow suitable conditions for face-to-face meetings between students and researchers. This situation imposed some limitations on the research plan. However, it also provided a unique opportunity for the author to explore an alternative method of co-designing the artefact with students in an online environment.

The mini-games deployed in the smart learning environment prototype emerged from the co-design process between the author and the students to facilitate computational thinking knowledge. Although the requirements for the smart learning environment were obtained and even modeled using formal notation as explained in Section 3.2.2, the core learning objectives that concretely demonstrate students' engagement, motivation, and interaction for enhanced learning experience needed to be co-designed. This created the need for the OCD process, whose methodological flow is presented in Figure 12. The OCD process suggests that one way for students to develop computational thinking skills is to engage them in the design process of the computational thinking artefact (Article VI). In particular, during the COVID-19 pandemic, a user-centered study in university physical premises such as classrooms was unfeasible (García-Peñalvo, et al., 2021a; Béjat & Vera, 2022); hence, an innovative approach such as OCD that would allow students to create mini-game concepts and acquire computational thinking skills was a suitable option. As depicted in Figure 12, the OCD process systematically creates an opportunity for students to collaborate in groups for ideation on educational game concepts, elements, and scenarios. The first phase of the OCD process allows for familiarization between students and researchers.

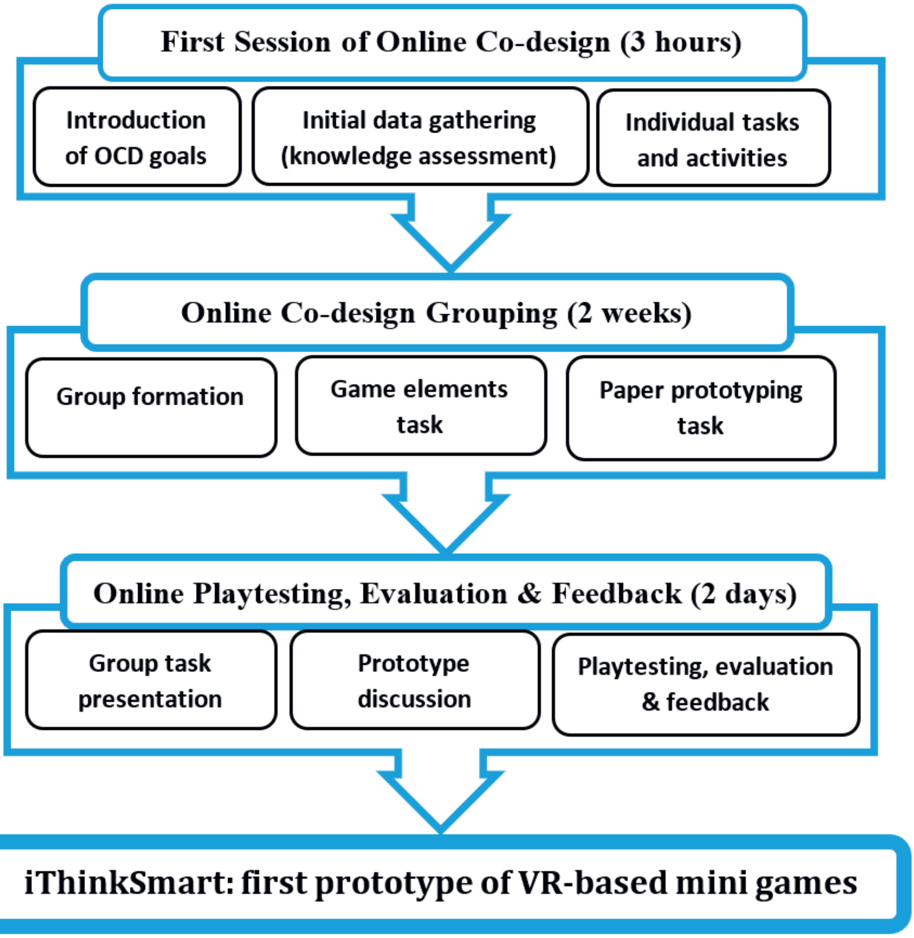


Figure 12. Innovative online co-design process for designing low-fidelity prototypes of mini-games for computational thinking.

The following text provides a brief introduction to the main objective of the study and the specific tasks to be accomplished during OCD. As illustrated in Figure 13, the researcher connected with the students through an online platform where the background of the study, main objectives, activities, and general lifecycle of the OCD process was provided.

Figure 13. Screenshot of the first session of online co-design with students highlighting the study goal.

After the objective of OCD was understood, the second phase of the OCD process allowed for students to be divided into groups of a manageable size to complete the task of creating low-fidelity mini-game prototypes over two weeks. The researcher provided adequate coordination and guidance to all of the groups while they independently completed their tasks. In addition, the third phase of the OCD allowed for playtesting and peer-review of the output of each group. The playtesting and peer-review were conducted at the group level to allow for constructive criticism that could further refine the concepts that emerged. Indeed, all of the activities demonstrated in the OCD process explored in this research provided means of integrating computational thinking into higher education; furthermore, they provided input for the implementation of the smart learning environment prototype.

The design activities conducted in this stage of the dissertation were detailed in Article VI. The study randomly recruited 12 computer science undergraduate students (eight male, four female) from Federal University Lokoja in Nigeria to participate in the co-design, which mainly occurred in an online environment. This OCD process also consisted of a researcher (the author); a postdoctoral researcher, who was also one of the doctoral researcher's supervisors; and a student coordinator, who was responsible for ensuring that students were able to participate in the study seamlessly online.

The study presented in Article VI adopted qualitative and quantitative data gathering and analysis techniques. At the beginning of the study, a presurvey was conducted to elicit participants' prior knowledge and experience about co-design, games, and game elements. The design activities employed several approaches, including note taking, observation, interviews, paper prototyping, and questionnaires to collect data. Items in the interviews and the questionnaire (see Appendix 2) were designed by the researchers based on the context of this study and were validated by three educational technology experts (Anyango & Suleman, 2018). The interviews were conducted through the Zoom platform, recorded, and later transcribed. The author coded the transcribed interviews and identified themes that discussed users' first-level reaction. Moreover, the paper prototypes were playtested by the researchers and participants, as demonstrated in Article VI.

3.4 Evaluation of the iThinkSmart prototype

3.4.1 Experimental design for evaluating the iThinkSmart prototype

This section presents the experimental procedure followed to conduct the demonstration and evaluation of the iThinkSmart prototype consisting of mini-games. The overall intention of evaluating the iThinkSmart mini-games was to examine their efficacy on students' learning achievement in terms of computational thinking competency, cognitive benefits, and their interest and attitude toward learning computational thinking concepts after playing. In other words, the initial evaluation sought to

i. determine to what extent the iThinkSmart mini-games improved students' computational thinking competency;
ii. investigate whether students who played iThinkSmart mini-games to facilitate computational thinking gained more cognitive benefits compared with students who played a different game;

iii. examine whether the iThinkSmart mini-games inspired students' interest and attitude in learning computational thinking concepts after completing the experiment.

Therefore, an experimental procedure consisting of two groups (an experimental group and a control group) was established to demonstrate the initial evaluation. The remainder part of this section presents the details of the experimental settings, participants, data collection, analysis, results, and discussions based on the initial evaluation of the artefact prototype.

3.4.2 Participants and research settings

The participants for this evaluation study were recruited from a public university (Federal University Lokoja) located in the North-Central region of Nigeria. The author obtained written permission from the university authority (the vice-chancellor through the academic registrar) to conduct the research with their students. The study adopted a purposive convenience sample of 47 computer science students, including six students who initially participated in the co-design process. An online invitation form was sent to the students and 60 of them registered their consent; however, only 47 turned up on the date of the experiment. This number of students was logistically manageable and considered sufficient for an initial evaluation of a VR application as demonstrated in a similar study (Butt, et al., 2010). Since the evaluation still remained at the initial stage, the inclusion of participants in the study was intentionally limited to computer science students. Another rationale behind choosing computer science students as participants was that they had previously completed courses that would have provided them with a sufficient background in computational thinking and problem solving. This background knowledge was useful for ensuring that all of the students participating in the study had equivalent experience. The procedure followed to conduct the experiment is presented in Figure 14.

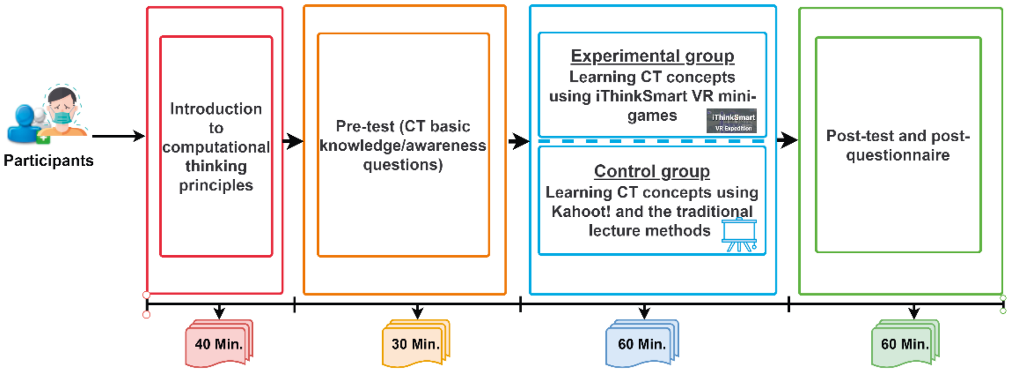


Figure 14. Procedure followed to conduct the initial evaluation of the iThinkSmart prototype.

Generally, the evaluation experiment engaged participants in pre–post tests and post-questionnaires. First, the evaluation study began with an introduction to computational thinking concepts and the goal of the developed artefact. This introduction lasted 40 minutes. Next, a short pretest consisting of yes/no and multiple-choice questions was administered to the students for 30 minutes (see Appendix 3A for the pretest questions). The pretest was aimed at evaluating whether all of the students had equivalent knowledge on the basics of computer science, problem solving, and algorithmic design and had all completed the first-year Introduction to Computer Science course. Subsequently, two groups were formed through random sampling, consisting of the experimental (n = 21) and control (n = 26) groups. Each group engaged in a learning session for 60 minutes using different learning approaches to acquire computational thinking knowledge.

The first learning approach was through the iThinkSmart mini-games. Students in the experimental group solved computational puzzles by playing iThinkSmart mini-games to gain computational thinking skills, as demonstrated in Figure 15. The second learning approach was through an online game platform named Kahoot!.[1] Kahoot! is an online GBL platform

[1] https://kahoot.com/

that allows players to participate in multiple-choice quizzes. The rationale for using Kahoot! as a second learning approach was that the platform is responsive (i.e., it automatically adjusts to different devices' screen resolutions), making it accessible via several devices including smartphones, so the students did not need to use a laptop or desktop computer, which would have created additional requirements for the experiment. Furthermore, Kahoot has been widely used as an educational game in recent years and can facilitate simple learning objectives.

Figure 15. Images showing how students were engaged in playing the iThinkSmart mini-games.

The control group solved similar puzzles by playing Kahoot! to gain computational thinking skills. The Kahoot! learning approach was blended with the traditional teaching method using a PowerPoint presentation with multimedia. The traditional approach was most familiar to the students. Importantly, the computational thinking learning contents deployed in both groups were identical. Therefore, the students were expected to achieve the same learning goal through both approaches. Computational thinking concepts such as algorithmic thinking, recursive thinking, problem decomposition, abstraction, and pattern recognition were part of the learning goals.

3.4.3 Instruments, data collection, and analysis

The instruments for the evaluation of the iThinkSmart prototype were adapted from several sources, mainly to assess the level of students' computational thinking competency as well as their perceived cognitive benefits, learning interest, and attitude. Some of the quizzes in the mini-games were based on the computational thinking competency test for beginners (BCTt v.1) developed by Zapata-Cáceres et al. (2020). The BCTt v.1 comprises 12 puzzles with multiple-choice questions. This test was recently adapted in a related study to assess students' computational thinking competency in an educational computer game (Hooshyar, et al., 2020). The test difficulty level increases as the player progresses from the first question to the last, as presented in Appendix 3. Each question is weighted and has a fixed time of 60 seconds for players to provide an answer. For the experimental group, data regarding the players' computational thinking competency score were computed using the objective distance model explained in Section 4.3 while they were engaged in the gameplay. Therefore, the calculation of players' computational thinking competency in the iThinkSmart prototype was performed on a real-time basis using the objective distance (OD) model. On the other hand, the control group played an online quiz using Kahoot! and had their computational thinking quiz weighted and computed manually.

Furthermore, instruments for measuring students' cognitive benefits were adapted from Lin et al. (2020), while the instruments for evaluating students' learning interests and attitudes were adapted from Hwang and Chang (2011). The data collection employed both quantitative and qualitative approaches. In particular, the study collected data from the pre-quiz, post-quiz, and post-questionnaires. All of the items in the questionnaire consisted of a 5-point Likert scale (from 1 = strongly disagree to 5 = strongly agree). Additionally, the author provided a free form containing a list of questions to prompt students who participated in the experimental group and played the iThinkSmart mini-games to provide feedback based on their experience. This free form was aimed at eliciting information that reflected students' experiences about the prototype and how it met their earlier expectations. Moreover, this information aimed to guide future improvements of the artefact to meet users’ specific needs.

For the data analysis, both quantitative and qualitative techniques were applied to analyze students' data. The SPSS 27 software package (IBM SPSS, 2020) was used to analyze the quantitative data, while a content analysis of participants’ first-level reaction after playing the iThinkSmart mini-games was used to analyze the qualitative data following Ge et al.’s (2021) approach. The validity and internal reliability of the instruments adapted for the post-questionnaire were computed. In terms of Cronbach’s alpha, the cognitive benefit items had an alpha value of 0.89, the learning interest items had one of 0.78, and the learning attitude items had one of 0.81. This indicated that the conditions for internal consistency and reliability of the instrument were satisfied.

3.5 Research ethics

Adherence to ethical conduct in scientific research is critical not only for maintaining the quality and integrity of the study but also for ensuring that the study outcomes are credible and reliable (Gomera, 2020). In the current era of digitalization, ethical issues have become increasingly important in computing research since most interventions are now pervasive in nature and almost all actions are connected to elements of computing (Oriogun and Ogunleye-Johnson, 2012). Therefore, there is a need to consider ethical issues when conducting a user-centered study from the perspective of the conceptualization, design, and development of an artifact. As highlighted by Kapinga (2020), conducting DSR creates the opportunity to make the value of such a study explicit enough that ethical consideration can easily be handled through the generation of new knowledge, leading to the creation of innovative artefacts.

In this study, ethical issues were considered throughout all research phases. This research conformed to the guidelines for conducting responsible research provided by the Finnish National Board on Research Integrity (Kohonen, et al., 2019). Similarly, the study obtained approval from the research ethics committee of Federal University Lokoja and adhered to all guiding principles of the university. Moreover, several steps were taken to ensure ethical compliance at every stage of the research; specifically, (i) the autonomy of all subjects recruited to participate in this research was respected; (ii) subjects were informed about the objectives of the research and that participation in the study was voluntary; (iii) participants were allowed to pull out of participating in the research at any stage; (iv) informed consent was always obtained from the participants to use data collected during the study for research purposes; (v) data collected during the experiment were anonymized before analysis and presentations were conducted; and (vi) images collected as part of the data were anonymized and blurred to conceal the identity of the participants.

3.6 Summary

This chapter presented an overview of the research methodology, including the research setting, data collection instruments, research subjects, and data analysis techniques. It also provided information about how this dissertation addressed ethical issues in conducting the user-centered study.

4 Results

4.1 Findings from the problem explication

The initial problem explication revealed that the majority of students at the federal university and state polytechnic in Lokoja, Nigeria generally lacks an understanding of introductory programming courses. Articles I, III, and IV reported that the students mainly learn programming through the traditional methods and are not exposed to a learning system that can enhance their understanding. Furthermore, the students do not have a sufficient number of computers to practice programming after class, while the majority cannot afford a laptop to support their learning (Articles I and III). These findings were not surprising since the author has experienced how students fail programming courses in previous years. Moreover, similar results have been reported in studies conducted in other universities in Nigeria. For example, Dasuki and Quaye (2016) revealed that the majority of computer science students at a Nigerian university found it difficult to pass the introductory programming course. Another study by Sunday et al. (2020) reported that over 60% of students who enrolled for an introduction to programming course at a Nigerian university failed due to a lack of understanding of programming concepts.

In this dissertation, the specific problem that students were encountering that limited their understanding of programming concepts became clearer to the author after iteratively conducting other DSR activities with the stakeholders. For example, this research was initially focused on addressing the problem of programming education in general; however, the iterative process of requirement elicitation and concept design revealed that students were finding specific programming topics difficult to comprehend due to how they were theoretically taught in class without demonstration and visualization. As demonstrated in Articles I and III, students and teachers mostly use the traditional method of learning and teaching rather than using technology. Table 5 reveals that mobile learning, or any form of smart learning were not utilized for programming education in the universities where the study was conducted. However, the students and teachers expressed an eagerness to adopt a smart learning system for teaching and learning.

Table 5. Users' experience and expectations regarding the use of a smart learning environment for programming education at the federal university and state polytechnic in Lokoja, Nigeria (adapted from Article III).

Users	Constructs	M	SD
Students (n = 210)	I have been taught a programming course with a mobile smart learning solution	3.10	1.11
	I prefer to learn to program with a smart learning solution rather than a white/blackboard	1.62	0.85
Teachers (n = 15)	I have been teaching programming courses with a mobile smart learning solution	3.67	1.11
	I prefer to teach programming with a smart learning solution rather than a white/blackboard	1.67	0.49

Note: The data analysis was performed according to a the Likert scale, where the options were coded as follows: strongly agree (SA = 1), agree (A = 2), neutral (N = 3), disagree (D = 4), and strongly disagree (SD = 5).

The findings from the qualitative interviews in Article III further explicated users' learning challenges and expectations for the proposed intervention. In terms of challenges, the students generally remarked that the traditional method of learning programming concepts rarely enhances their understanding and comprehension. One of the student interviewees stated the following: "We studied programming courses right from year two using the whiteboard and the marker; the lecturers come into the class and explain on the whiteboard; we then research on our own." A further investigation into how students practice programming after class revealed a lack of computers provided in the institutions' laboratories, which limited students' opportunities for hands-on practice and self-learning. For example, one student stated that, "those that have laptops, install the programming tool on their laptops and learn with the guide from an online tutorial." Unfortunately, only a few students in Nigerian higher education institutions can afford a personal computer (Article I).

Furthermore, participants of the interviews in Article III expected an intervention that could intelligently classify learners and learning contents into different levels to allow novices to learn progressively from the basics to complex topics. Furthermore, some respondents expressed an expectation for a smart learning environment that allows the automatic assessment of learning progress, which one student articulated as follows: "I prefer a smart learning environment that has a grading or credit rewarding mechanism to encourage learning." These findings guided the design and implementation of the artefact presented in this dissertation.

In addition, the author attempted to confirm which introductory programming concepts the students found most difficult by asking them to rate a list of topics, as reported in Article VI. This study revealed that one of the most problematic topics for students to comprehend is recursion (see Figure 16). Notably, recursion is one of the computational thinking concepts that this research sought to facilitate the knowledge of among students.

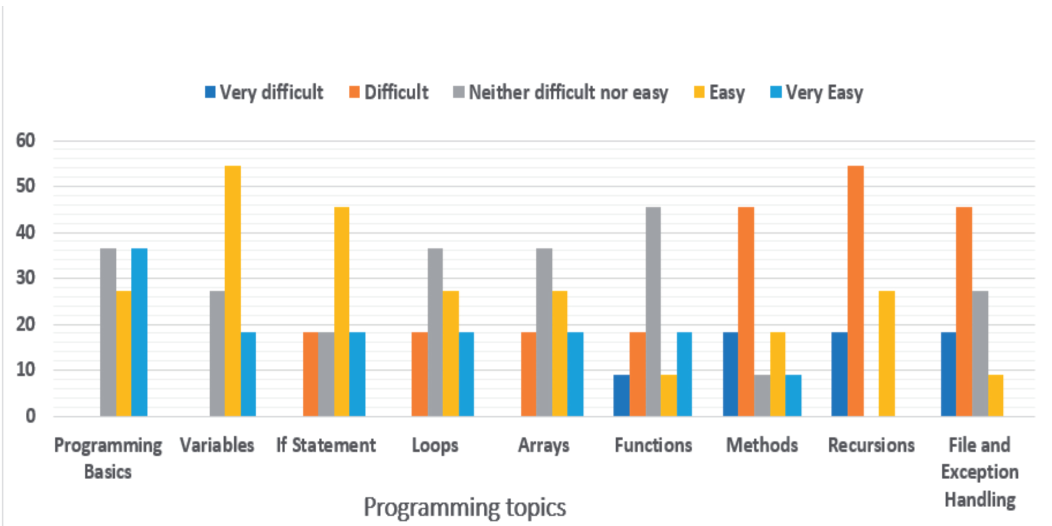


Figure 16. Perceived difficulty of programming topics by students at Federal University Lokoja.

4.2 Requirements for designing a smart learning environment for computational thinking education in the Nigerian context

Studies were conducted to elicit users' requirements in term of the features expected of a smart learning environment for facilitating teaching and learning in a federal university and state polytechnic in Lokoja, Nigeria (Article III). The findings revealed that the students and teachers were mostly interested in a system that provides learning guides and feedback. Other features of a smart learning environment mentioned were personalized learning, automatic task scheduling, and location awareness, as presented in Figure 17.

Figure 17. Users' requirements regarding features of a smart learning environment (adapted from Article III).

Furthermore, the author investigated the extant literature to gain an understanding of how computational thinking is taught at higher education institutions. The systematic literature review uncovered several approaches used to deploy the teaching and learning of computational thinking concepts in higher education institutions, including puzzle-based learning, game-based approaches, and other instructional design tools (Articles II). This dissertation adopted GBL as the pedagogical approach for teaching computational thinking concepts because the study in Article II demonstrated its relevance to programming education.

Regarding the relevant technology for supporting students' learning, immersive VR technology was found to be useful (Article IV). In computing education in particular, the use of VR applications is attracting huge scholarly interest owing to its ability to immerse learners, motivate students, and provide high-level interaction to enhance the learning experience. Evidence from recent studies also supports the affordances of VR technology for computing education through characteristics such as immersion, engagement, interaction, presence, and immediacy (Radianti et al., 2020; Seruga et al., 2020).

4.3 Design and development outcome: iThinkSmart prototype

This section presents the outcomes from the design and implementation of the iThinkSmart prototype. The section begins by demonstrating functional requirements for designing a VR-based smart learning environment. Then, it presents the outcomes from the OCD activities conducted between the researcher and students to co-design some of the mini-game concepts contained in the iThinkSmart prototype.

The functional requirements in Article V demonstrated the different components of the UML for designing a VR-based smart learning environment. For instance, the use case diagram, activity diagram, and sequence diagram were presented in Article V. An example of the use case diagram presenting the functional requirements of the smart learning environment is presented in Figure 18.

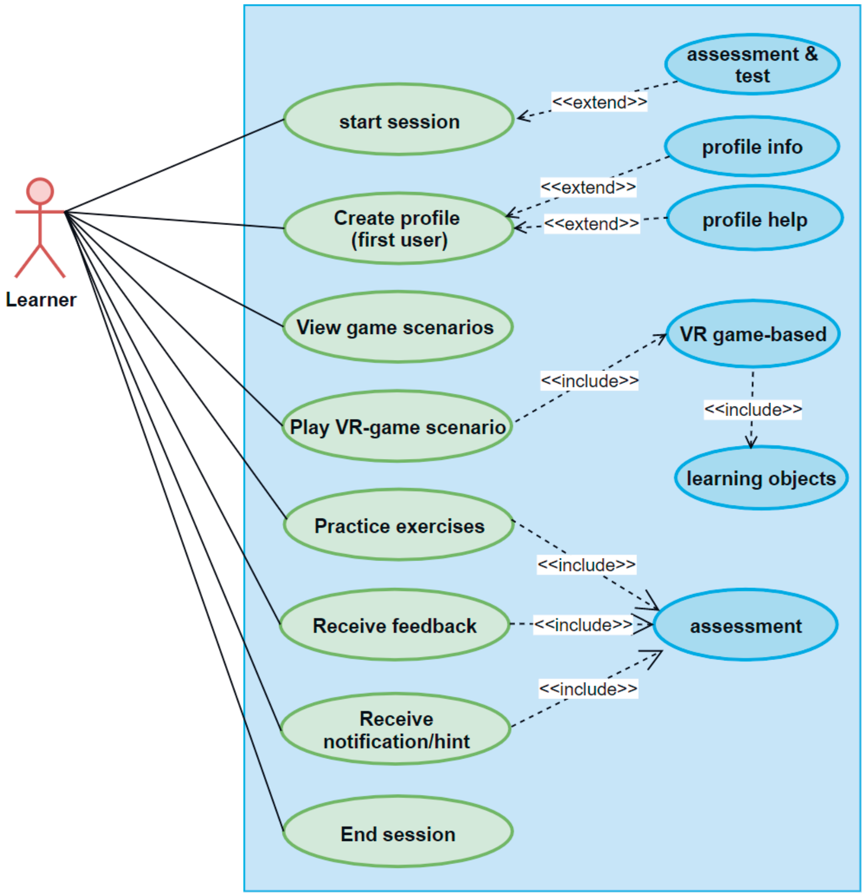
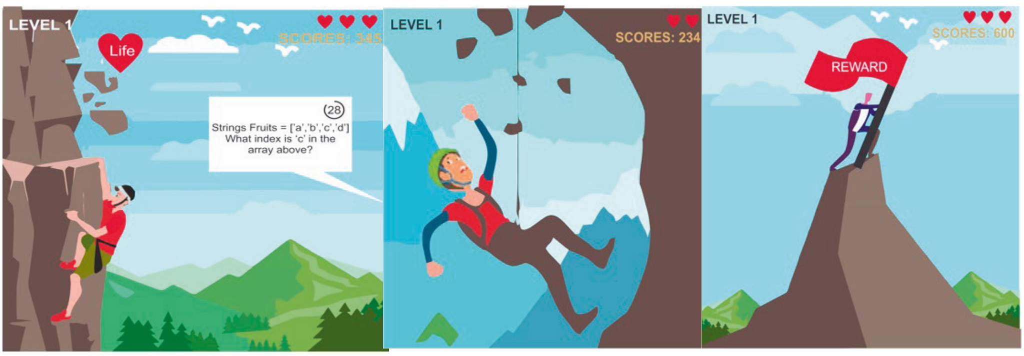


Figure 18. Use case diagram of the virtual reality-based smart learning environment (Article V).

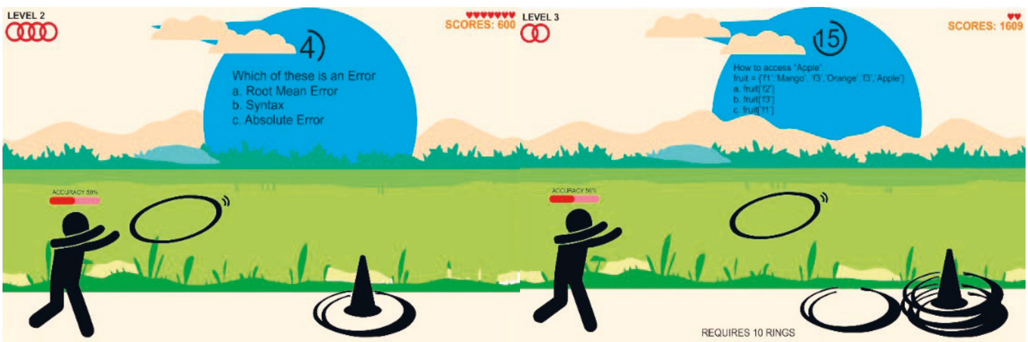
The use case diagram illustrates a scenario where a learner is playing mini-games within a virtual world to acquire computational thinking knowledge such as algorithmic thinking.

The findings from this OCD study indicated that the majority of the students were familiar with playing varied games and had experience with game elements but had not developed game concepts. Therefore, most of the students improved their practical knowledge of game creation through creative thinking demonstrated during the OCD activities. Article VI presented a formative evaluation of the contextual game prototypes of low fidelity (Oyelere et al., 2022), which were playtested by the students. Furthermore, the outcomes of the study in Article VI provided a selected wishlist of game elements that would support the development of the iThinkSmart mini-games. This wishlist included an interactive environment, character, challenge, competition, rules, rewards, graphic animations, and navigation.

During the co-design process, the author did not influence the ideation of the students while they were conducting their group tasks, which focused on providing a contextual game scenario. The author's only input was to motivate each group to collaboratively complete the two-week tasks and point out specific instructions outlined in the tasks. Once the rough paper ideation was completed, students developed a paper prototype of the mini-games and later refined them into mock-up prototypes, as presented in Figure 19a and b, which inspired the design and implementation of the high-fidelity prototype.



(a) Mount Patti Treasure Hunt mini-game



(b) Targeted Throws mini-game

Figure 19. Examples of conceptual mini-games.

Figure 19a showcases how students gamified the climbing of the famous Mount Patti hill (located in Lokoja city, Nigeria) to explore its potential. According to the students, the use of contextual scenarios such as Mount Patti in the Mount Patti Treasure Hunt (MoPaTH) mini-game can motivate players to discover its treasures while solving puzzles that can enhance their learning (Oyelere et al., 2019b). Indeed, MoPaTH was implemented in the iThinkSmart prototype to demonstrate how a contextual idea from students can enhance their learning. The students' ideas about puzzles were mainly quizzes that require computational thinking skills to provide short answers within a limited time. Figure 19b demonstrates a contextual game competition called Targeted Throws where young people test their level of measurement accuracy by throwing an object at a targeted distance. The learning objective behind this game is similar to that of MoPaTH; however, the contextual storyline is different. Moreover, the reward mechanism for Targeted Throws is the numeric score, whereas that of MoPaTH not only rewards with a numeric score but also provides the opportunity for the player to explore interesting sites after winning the mini-game. Because of this additional reward feature in MoPaTH, the author included it in the list of mini-games contained in the high-fidelity iThinkSmart prototype.

The development of the iThinkSmart prototype followed the model–view–controller service (MVCS) architecture in Unity, as illustrated in Figure 20. According to Ferrone (Ferrone, 2019), Unity is currently one of the most popular game engines used by both amateur and professional game developers worldwide.

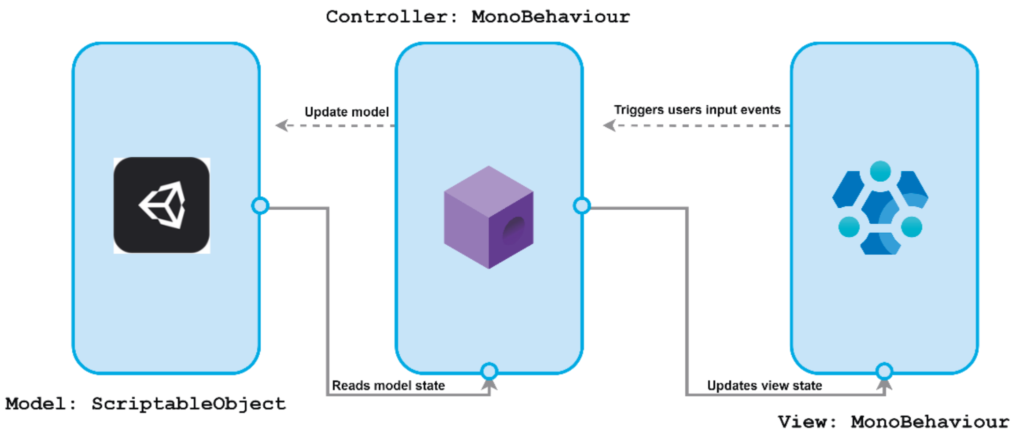


Figure 20. Model–view–controller service architecture of the iThinkSmart mini-games implemented in Unity.

Moreover, the Unity game engine was selected due to the following rationale. First, it is easy to use and provides all of the necessary tools for modeling interactive 3D environments. Second, Unity has clear documentation for its application programming interface that supports the user community, including amateur users. Third, Unity allows for the deployment of VR applications on different platforms such as PCs, consoles, and mobile devices (Segura, et al., 2020).

The high-level system architecture of the iThinkSmart VR prototype application has five components that interact with each other to deliver the goal of the smart learning environment, as depicted in Figure 21. These components are as follows:

i. mobile application interfaces
ii. learning logs
iii. a computational thinking competency model
iv. learning objects
v. database and repository

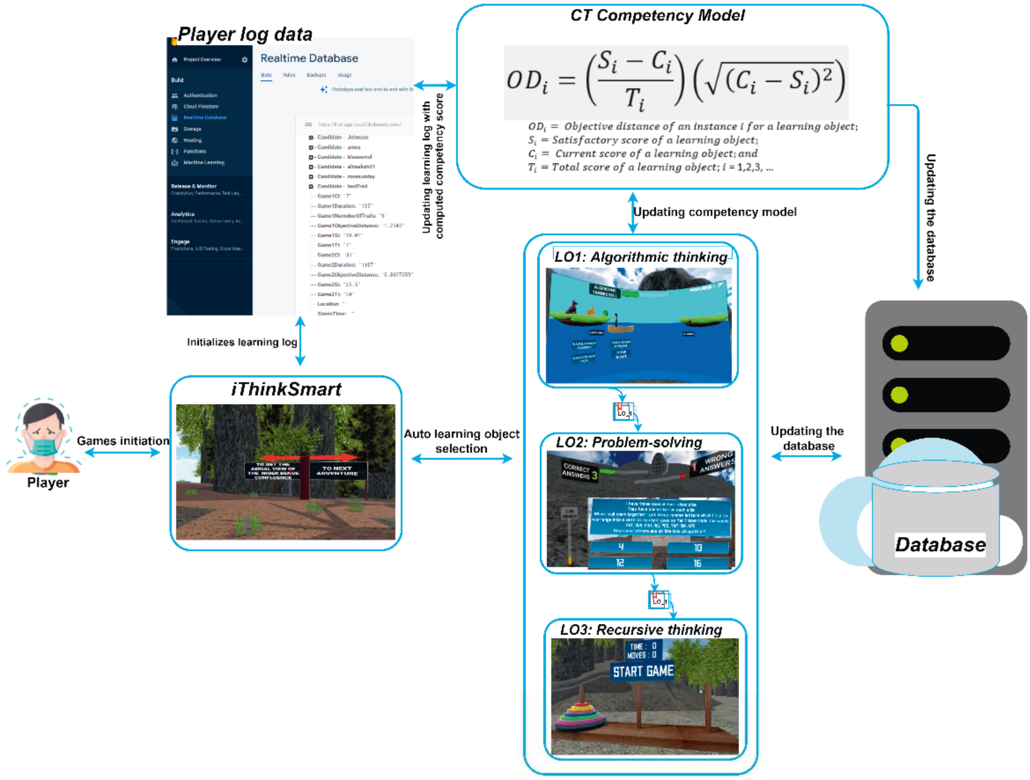


Figure 21. High-level system architecture of the iThinkSmart VR prototype application.

The mobile application interface allows players to launch the app from a smartphone and sign in with basic information such as their username and academic level. The learning log contains all of the data generated during gameplay. For example, the learning log is updated from the moment the player signs in until their session ends. The computational thinking competency model assesses the players' learning achievement and updates the learning log with such data. The learning objects are the mini-games aimed at delivering tangible knowledge of computational thinking concepts. The database warehouses all of the players' information and learning data generated during gameplay.

The iThinkSmart prototype was designed to run on Android smartphones, although it can also be built for other platforms such as desktop Windows and iOS. The first prototype of iThinkSmart was built to run on Android smartphones because the majority of students who are the targeted users of the application use Android smartphones (Articles I and III). The iThinkSmart prototype was developed to run on smartphones with minimal technical specifications such as a 1200 MHZ processor, 1 GB of RAM, and at least 250 MB of free memory storage. Users launch the application from their smartphone and then insert the smartphone into a VR head-mounted display (HMD). Bearing in mind that the iThinkSmart prototype is meant to support students' learning in a developing-country context, the author deliberately integrated the application to function with simple and affordable devices. For example, an unsophisticated HMD such as Google Cardboard or other relatively cheap HMDs and a hand controller (as depicted in Figure 22) can be used to play the iThinkSmart mini-games in a virtual world. Students at Federal University Lokoja can meet the technology requirements for using the iThinkSmart prototype. According to the study in Article VI, the cost of the HMD is approximately US$5, which is the only additional cost for students who already possess a smartphone to play the iThinkSmart mini-games.

Figure 22. Example of a HMD and a Bluetooth hand controller used for iThinkSmart.

Each mini-game in the iThinkSmart prototype is modeled to deliver a computational thinking concept as a learning object. Players can play these games progressively starting from basic challenges and progressing to more complex ones. During the gameplay, the player is guided through instructions to experience higher interaction and engagement. As the player progresses in the virtual world, computational thinking tasks are presented, which the player can solve within a stipulated time. Hence, the progress of the player is dependent on the number of problems solved.

Additionally, the player's computational thinking competency is measured while playing the iThinkSmart VR mini-games. This measurement is performed by using Chaichumpa and Temdee's OD model (2018). The OD model measures players' competence by applying the following formula:

$$OD_i = \left(\frac{S_i - C_i}{T_i}\right)\left(\sqrt{(C_i - S_i)^2}\right) \quad \dots (1)$$

$OD_i =$ *Objective distance for an instance i of a learning object*;
$S_i =$ *Satisfactory score for an instance i of a learning object*;
$C_i =$ *Current score for an instance i of a learning object*; *and*
$T_i =$ *Total score for an instance i of a learning object*;
$i = 1, 2, 3, \ldots$

The OD model was used to evaluate students' competency during the gameplay session in the virtual learning environment to ascertain their level of computational thinking skills and digital literacy (Temdee, 2019). The model measures the distance between the learner's expected/satisfactory score (S_i) and their current score (C_i), mainly to provide personalized support and intelligent feedback to the learner. Based on the player's learning outcomes computed using the OD model, iThinkSmart renders three types of personalized feedback, namely rewards for excellent or satisfactory performance, hints on how to improve learning when players perform poorly, and warnings when the player performs incorrect actions. This study adopted the OD model because it has been tested and found to perform well with high accuracy (Chaichumpa & Temdee, 2018; Temdee, 2019; Chaichumpa, et al., 2021). One of the characteristics of the OD model that attracted the author is that it measures the learner's computational thinking competency in concrete terms and provides a simple and straightforward representation.

The iThinkSmart VR-based prototype application consists of mini-games that aim to provide five basic computational thinking concepts, which are modeled as learning objects within the games. Three mini-games are integrated into the iThinkSmart prototype: (i) River Crossing, (ii) MoPaTH, and (iii) Tower of Hanoi. Table 6 provides information regarding each mini-game and what computational thinking knowledge they are targeted at delivering as a learning object for students' cognitive benefits. The author believed that the knowledge gained from these mini-games would aid the player to develop a concrete understanding of the computational thinking concepts necessary for programming education, as demonstrated in Figure 4.

Table 6. Computational thinking concepts connected to the iThinkSmart virtual reality mini-games.

Adapted/co-designed mini-games	**Targeted computational thinking concepts**				
	Algorithmic thinking	Recursive thinking	Problem decomposition	Problem abstraction	Pattern recognition
River Crossing	✓	✓	✓	✓	✓
MoPaTH	✓		✓	✓	
Tower of Hanoi	✓	✓	✓	✓	✓

Some of the puzzles deployed as mini-games in the iThinkSmart VR-based prototype have common applications in computing. They are used to demonstrate the teaching of computational thinking concepts such as algorithmic thinking. For example, basic algorithmic techniques such as the brute force algorithm and the divide-and-conquer algorithm can be taught to programming novices using puzzles such as the Easter Egg puzzle, the River Crossing puzzle, and the Santa Claus puzzle (Tsalapatas, et al., 2012).

Indeed, Ratnadewi et al. (2018) examined how the river crossing puzzle can be used to demonstrate an AI approach for solving the breadth-first search (BFS) algorithm. Unfortunately, game-based techniques such as iThinkSmart, which uses these puzzles to demonstrate the teaching and learning of computational thinking in Nigerian higher education institutions, are still lacking. Therefore, this study adapted some of these common puzzles and integrated them alongside one of the contextual mini-game co-designed with students to provide a mixed approach for facilitating computational thinking skills among computer science students in the Nigerian university context. Specifically, the River Crossing and Tower of Hanoi puzzles were adapted, while the MoPaTH mini-game was co-designed by the author and students.

MoPaTH

The MoPaTH mini-game was based on the real Mount Patti located near the study location, which has a historical bearing on the creation of the country called Nigeria. Mount Patti is a popular mountain that is over 1500 feet tall and located in the North-Central region of Nigeria. This mountain served as the government base of the colonial masters and from where the name "Nigeria" was given to the country (Abdullahi, et al., 2019). It is also a tourist and recreational center. The students gamified the climbing of this mountain based on their experience. The game engages the player to unlock the treasures located on top of the mountain (see Figure 23) by solving puzzles consisting of computational thinking problems. For instance, one of the MoPaTH puzzles requires the use of random outcomes from rolling dice to reverse-engineer what is on their faces. Players can only progress to climb to the top of the mountain by answering puzzles correctly and promptly, which requires critical thinking. The player is rewarded in points for any correct answer provided.

Figure 23. Screenshot of the top of Mount Patti from the Mount Patti Treasure Hunt mini-game.

River Crossing

As already established, the river crossing problem has been applied for teaching problem-solving concepts in the mathematics, computer science, and engineering fields (Ito, et al., 2015). The river crossing problem's integration into the iThinkSmart prototype was motivated by evidence that the game fosters learning when played in a highly interactive virtual environment (Valle-Tourangeau, et al., 2013). River Crossing (see Figure 24) tests students' competence in algorithmic thinking by requiring them to move items (a dog, rabbit, and melon) across a river through applying permutation and combination concepts in a thoughtful process of a finite number of steps while following the game constraints provided in the instructions. The lowest number of moves implies excellent algorithmic thinking skills.

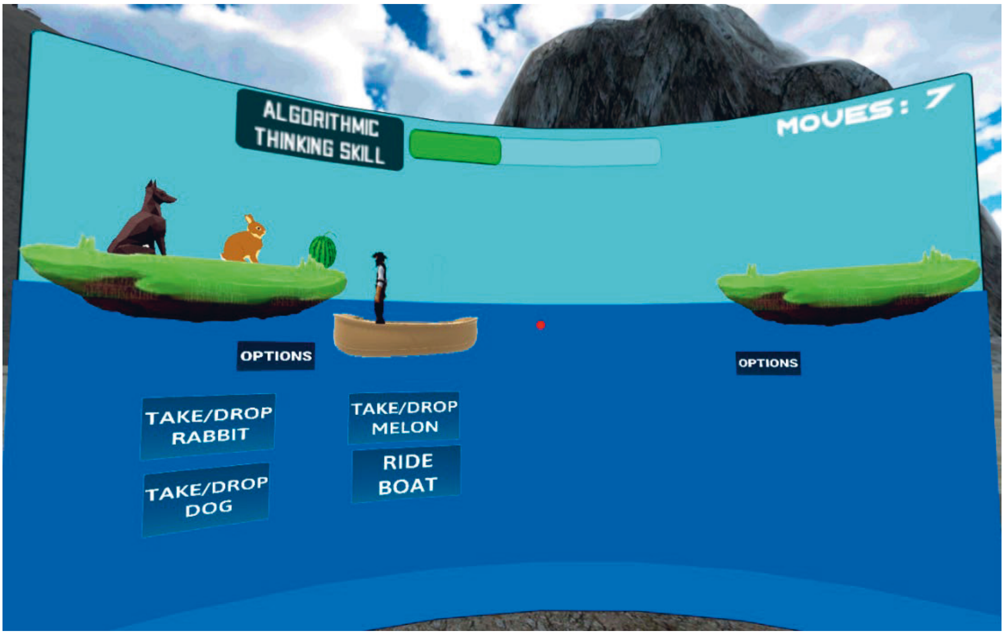


Figure 24. Screenshot of the River Crossing mini-game.

Tower of Hanoi

The Tower of Hanoi is a mathematical puzzle, also known as the Tower of Brahma (Awan, 2010). Teaching recursion to computer science students who are novices can often be difficult. Some of the mathematical approaches used to teach recursion include factorial and Fibonacci functions, which may not provide concrete knowledge due to their abstract nature. One way to demonstrate the concept of recursion is through visual simulation. According to Butgereit (2016), using games such as Tower of Hanoi can enhance students' comprehension of the recursion concept. The iThinkSmart prototype implemented the Tower of Hanoi, as depicted in Figure 25.

Figure 25. Screenshot of the Tower of Hanoi mini-game.

The Tower of Hanoi mini-game consists of three vertical pegs and an n-number of discs (where n ranges from 1 to infinity) piled on the first peg (see Figure 25). The discs are arranged in order of their size, with the smallest disc on the top and the largest disc on the bottom. The player must move all of the discs from the first peg to the last peg in the same order by observing the following rules: (i) only one disc can be moved at a time, and (ii) a disc of a larger size cannot be placed on top of smaller disc. Figure 26 presents an error message of an illegal move, namely placing a larger disc on top of a smaller disc.

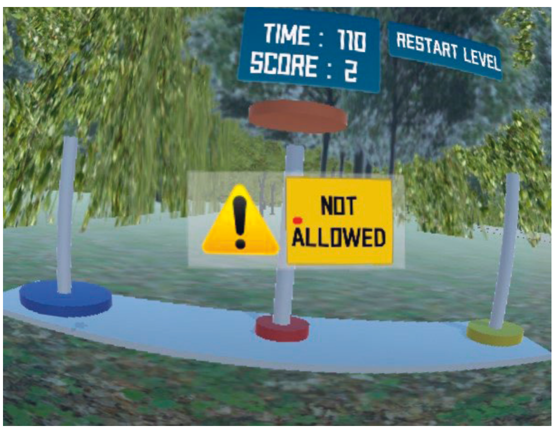


Figure 26. Screenshot showing an error from an illegal move.

4.4 iThinkSmart prototype demonstration and evaluation results

This section presents the results of the evaluation of the iThinkSmart prototype. The three main components that the artefact targeted were evaluated, namely students' computational thinking competency, cognitive benefits gained from playing the mini-games, and interest and attitude toward learning computational thinking concepts after playing the iThinkSmart mini-games. In addition, players' responses to the post-questionnaire of free forms, which elicited their perspective on the generality of the application based on their experience, were analyzed.
Moreover, during the evaluation of the iThinkSmart prototype, the author conducted a pretest that was aimed at evaluating whether all of the students recruited for the study possessed equivalent knowledge on the basics of computer science and had completed the first-year Introduction to Computer Science course. The results indicated that 66% of the students were familiar with the term computational thinking but only 44.7% understood what the term entails. In addition, 80.9% responded that they had not taken any course tagged with computational thinking; however, all of the students (100%) had completed the Introduction to Computer Science course.

4.4.1 Computational thinking competency

To evaluate the extent to which the iThinkSmart mini-games facilitated computer science students' computational thinking competency, this study computed the post-test scores of the experimental group ($n = 21$; who learned computational thinking concepts by playing the iThinkSmart mini-games) and the control group ($n = 26$; who learned computational thinking concepts by playing Kahoot!). Each test was weighted from 1 to 10, where 1 was the lowest score and 10 was the highest score. According to the results of the independent *t* test for both groups presented in Table 7, the mean score (μ) for the experimental group was 8.81 and the standard deviation (σ) was 2.70, whereas the mean score (μ) for the control group was 6.96 and the standard deviation (σ) was 2.36. These results suggested that the 21 participants in the experimental group demonstrated significantly superior computational thinking competency compared with the 26 participants in the control group (t (45) = 2.51, p = 0.016).

Table 7. Independent *t* test results of students' computational thinking competency based on post-test scores (adapted from Article VII).

Variable	Group	N	μ	σ	Std. Error	F-value	df	p value	t	d
CT competency Post-test	Experimental	21	8.81	2.70	0.59	0.05	45	0.02	2.51*	0.74
	Control	26	6.96	2.36	0.46	-	-	-	-	-

To further determine the extent to which the students who played the iThinkSmart mini-games gained competency in computational thinking, the effect size d was computed. According to Cohen (1988), the effect size statistic is defined as follows: d = 0.2 indicates a "small" effect size; d = 0.5 represents a "medium" effect size; and d = 0.8 means a "large" effect size. In this study, the d value was 0.74, which indicated an effect size bordering on large, implying that those who learned computational thinking concepts by playing the iThinkSmart mini-games gained more competency than those who learned computational thinking concepts by playing the traditional online game. This finding demonstrates the effectiveness of the developed iThinkSmart mini-games as an educational intervention for teaching and learning computational thinking concepts in the context of a Nigerian higher education institution.

4.4.2 Cognitive benefits

Further investigation of the efficacy of the iThinkSmart prototype was focused on whether the mini-games provide cognitive benefits to students and to what extent they support learners' memory. As demonstrated in Table 8, the experimental group gained slightly greater cognitive benefits than the control group. For example, the students from the experimental group were able to comprehend the concepts of computational thinking easily (μ = 4.48; σ = 0.51) compared with students from the control group (μ = 4.15; σ = 0.61). Similar results were witnessed across all of the items that measured students' cognitive benefits.

Table 8. Descriptive statistics of cognitive benefits of the iThinkSmart prototype.

Group	Group	N	μ	σ
The game approach made the comprehension of computational thinking topics easier.	Experimental	21	4.48	0.51
	Control	26	4.15	0.61
The game approach made the memorization of computational thinking topics easier.	Experimental	21	4.62	0.50
	Control	26	4.12	0.52
The game approach helped me to better apply what was learned.	Experimental	21	4.48	0.51
	Control	26	3.88	0.71
The game approach helped me to better analyze the problems.	Experimental	21	4.38	0.50
	Control	26	3.88	0.71
The game approach helped me to have a better overview of the content learned.	Experimental	21	4.43	0.60
	Control	26	3.96	0.66

Note: μ = Mean, μ = Standard deviation.

In addition, Table 9 presents the summary statistics of the cognitive benefits for the students based on mean scores. The mean score for the experimental group was 4.48, whereas the mean score for the control group was 4.00.

Table 9. Independent *t* test of cognitive benefits of the iThinkSmart prototype based on mean scores (adapted from Article VII).

Group	N	μ	σ	$\sigma_{\bar{x}}$	F-Value	df	t	d
Experimental	21	4.48	0.42	0.09	1.99	45	3.96	0.41
Control	26	4.00	0.40	0.08	-	-	-	-

p < 0.05

A comparison of the experimental and control groups revealed a slightly significant difference (t = 3.96, p < 0.05, d = 0.41). This implied that students in the experimental group who played the iThinkSmart mini-games gained moderately greater cognitive benefits than the students in the control group who used the alternative online game.

4.4.3 Learning interest

One of the components investigated in the evaluation of the iThinkSmart prototype was whether the mini-games influenced students' learning interest regarding computational thinking concepts. As shown in Table 10, the descriptive statistics delineated that students in the experimental group were more inspired to learn more about computational thinking concepts by playing the mini-games than students in the control group.

Table 10. Descriptive statistics of learning interest sparked by playing the iThinkSmart prototype mini-games.

Post-questionnaire items	Group	N	μ	σ
The computational thinking course is interesting.	Experimental	21	4.76	0.44
	Control	26	4.50	0.58
Learning more about computational thinking concepts is interesting.	Experimental	21	4.71	0.46
	Control	26	4.54	0.58
It is interesting to answer those computational thinking questions while learning through games.	Experimental	21	4.57	0.51
	Control	26	4.31	0.74

I always look forward to taking computational thinking topics and prepare for a programming class.	Experimental	21	4.71	0.46
	Control	26	3.88	0.95
The teacher's instructions on computational thinking concepts attracted my attention.	Experimental	21	4.48	0.51
	Control	26	4.23	0.43
Anything concerning computational thinking is always interesting to me.	Experimental	21	4.38	0.50
	Control	26	4.00	0.00
The computational thinking course is more interesting to me than other courses.	Experimental	21	4.38	0.67
	Control	26	3.38	0.64
Other courses do not attract me as much as the computational thinking course.	Experimental	21	3.76	1.00
	Control	26	3.08	0.80

As seen across the post-questionnaire items in Table 10, the mean for the experimental group was slightly higher than that for the control group. For example, the experimental group (μ = 4.76; σ = 0.44) found the computational thinking course slightly more interesting compared with the control group (μ = 4.50; σ = 0.58).

To investigate the extent to which the iThinkSmart mini-games sparked more learning interest in the students, the summary *t* test statistics were computed using the mean scores. The results are presented in Table 11.

Table 11. Independent *t* test of learning interest inspired by playing iThinkSmart mini-games (adapted from Article VII).

Group	**N**	**μ**	**σ**	**$\sigma\bar{x}$**	**F-Value**	**df**	**t**	**d**
Experimental	21	4.47	0.42	0.09	4.47	45	4.49	0.40
Control	26	3.99	0.31	0.06	-	-	-	-

$p < 0.05$

According to the results in Table 11, the mean score of the experimental group (μ = 4.47; σ = 0.42) was higher than that of the control group (μ = 3.99; σ = 0.31). This finding suggested a significant difference ($t = 4.49$, $p < 0.05$) in learning interest between the students who played the iThinkSmart mini-games and those who played an online game to acquire computational thinking skills. However, the resultant effect size of this difference was moderate ($d = 0.40$).

4.4.4 Learning attitude

Students' learning attitude toward computational thinking was the last evaluated component of the iThinkSmart prototype. As presented in Table 12, the students who played the iThinkSmart mini-games exhibited a more positive attitude toward learning computational thinking concepts compared with students who played an online game.

Table 12. Descriptive statistics of learning attitude orchestrated by playing the iThinkSmart mini-games.

Post-questionnaire Items	Group	N	μ	σ
The computational thinking course is worth studying.	Experimental	21	4.67	0.48
	Control	26	3.85	0.37
It is worth learning things about computational thinking.	Experimental	21	4.67	0.48
	Control	26	4.08	0.63
It is worth learning computational thinking concepts well.	Experimental	21	4.71	0.46
	Control	26	4.00	0.00
It is important to learn more about computational thinking concepts, such as algorithmic thinking, recursive thinking, and problem-solving.	Experimental	21	4.62	0.50
	Control	26	4.08	0.63
It is important to know and apply computational thinking concepts.	Experimental	21	4.67	0.48
	Control	26	4.12	0.33
	Experimental	21	4.48	0.60

I will actively search for more information and learn about computational thinking concepts.	Control	26	3.85	0.61
It is important for everyone to take the computational thinking course.	Experimental	21	4.52	0.51
	Control	26	3.58	0.70

For example, the majority of the students ($\mu = 4.48$; $\sigma = 0.60$) who played the iThinkSmart mini-games were more willing to actively learn more about computational thinking concepts compared with the students who played a different game ($\mu = 3.85$; $\sigma = 0.61$). Furthermore, the summary of the *t* test statistics in Table 13 indicates that the students in the experimental group demonstrated higher learning attitudes compared with the students in the control group.

Table 13. Independent *t* test of learning attitude caused by playing iThinkSmart mini-games (adapted from Article VII).

Group	**N**	**μ**	**σ**	**$\sigma_{\bar{x}}$**	**F-Value**	**df**	**t**	**d**
Experimental	21	4.62	0.40	0.09	2.97	45	6.34	0.40
Control	26	3.93	0.34	0.07	-	-	-	-

$p < 0.05$

However, when the effect size of learning attitude of both groups of students was computed, the results revealed a moderate effect size ($d = 0.40$). This implied that although both groups had a positive learning attitude, that of the experimental group was moderately higher than that of the control group.

4.4.5 Analysis of feedback from students who played the iThinkSmart mini-games

Table 14 presents the analysis of feedback received from students who played the iThinkSmart mini-games. The feedback was generally based on their experience of playing the games. The analysis followed Ge et al.'s (2021) approach for analyzing participants' first-level reaction to the gameplay. The first column in Table 14 provides the themes obtained from the participants' responses.

Table 14. Students' reactions and feedback after playing the iThinkSmart mini-games.

Themes	Analysis of responses and aspects of the game that require improvement
Graphics and esthetics	Participants provided feedback regarding the graphics. The specific aspect of graphics was not mentioned and the participants' expectations in some cases seemed vague; however, one participant specified how the graphics should be improved to a certain value of frames per second (fps). The following are examples of responses: "The graphics of the game should step up to 720 p and 60 fps." "The graphics should be enhanced." "The graphics of the game need an enhancement."
Navigation, instructions, demo, and hints	Some of the responses from participants suggested the need to improve the navigation, instructions, and tutorials provided within the game to guide players. Some example responses are provided as follows: "It will be better and easier to learn and navigate through the virtual environment if there was a kind of demo mode or detailed description/inscription on different objects and different locations on how to navigate and the available options to make." "An overview of how to play the game should be given so that anyone playing the game for the first time can easily play the game not necessarily attending a seminar on it or reaching out for a tutor." "Many games usually have a short demo, giving the steps and how the game should be played, iThinkSmart game should try and improve in this line..." "I want the arrow indicating the direction to be very visible...."
Movement, speed, and controller	Students anticipated an improvement in the aspect of speed, walking, and running behaviors of the game as well as easier navigation. For example, one of the participants asserted that the

	"…speed of the avatar only has pace and walking mode but not running mode, and it hinders players who would like to break records by making the fastest gameplay time." "…the ability for the movement to be faster." In addition, some of the participants wanted the controller to be friendlier to ensure high-level interaction.
Support for device	One participant reported how the device on which the VR application was installed could not properly support smooth gameplay: "…it seems the VR application doesn't run smoothly on all phones – some phones were not responding to movement."
Learning content	The participants wished to see more learning content deployed within the game. An example response was as follows: "…more topics to be added in the game."
Satisfaction with learning outcomes	A few participants praised the game and expressed satisfaction with how it provides computational thinking knowledge and problem-solving skills. One response was as follows: "…good job for developing such computational thinking system, I think it is very good in providing knowledge on problem-solving." Some of the participants were also positive about the game and even expected it to be hosted on the Google Play store to be easily downloaded by students from anywhere in the world. For example, one participant stated the following: "…let the game be launched to play store so that the world can have experience of iThinkSmart mini-games too."

Generally, the majority of feedback from participants was centered around the interaction and response of the game. Students seemed to find it difficult to move freely in the virtual world. Another related comment touched on the aspect of only allowing a walking mode in the virtual world and not a running mode. While these responses were highly useful for improving the iThinkSmart prototype, they basically revealed that this first prototype was able to truly engage the students to experience a high-fidelity VR application consisting of mini-games.

4.5 Summary and reflection on the design science process

This section presents a summary of the DSR process demonstrated in this dissertation and reflects on directions for further studies based on the iterative DSR process. Each DSR activity conducted for this dissertation provided concrete outputs in terms of articles, processes, or ideas that connected to the next activity as input or feedback into a previous activity for another iterative loop. Figure 27 follows the iterative design concept of Hevner and Chatterjee (2010) by presenting a cycle of research activities, beginning from the problem explication and continuing through the requirements analysis, design and implementation, evaluation, and reflection. The entire iterative process of DSR can be summarized in four steps, namely analysis, action, evaluation, and reflection. The analysis consisted of research planning, problem explication, and requirements elicitation. Outcomes from the analysis unravelled the contextual problems that computer science students faced in a federal university and a state polytechnic in Nigeria. In particular, students found programming concepts difficult to comprehend because the mode of teaching was classical, meaning that most of the teaching approaches were theoretical. Therefore, there was a clear need for a system to be developed that allows students to visualize programming concepts for an enhanced understanding. However, an understanding of programming concepts can be achieved through the demonstration of computational thinking skills. Hence, this dissertation opted to design a smart learning environment to allow students to visualize these concepts in a virtual world through a game-based approach.

Next, actions were planned and implemented to co-design paper and mock-up prototypes of mini-games with students. GBL and a computational thinking approach were found to be useful for demonstrating how concepts that aid programming knowledge can be visualized. Noteworthily, while the design and development of the mini-game concepts were ongoing, further explication of problems was equally conducted to narrow down and isolate specific requirements. The DSR method provides the opportunity for this kind of loop, where feedback interactions can occur within a circle of project implementation. In addition, the low-fidelity prototypes were refined and developed into VR mini-games that provide high levels of interaction, immersion, presence, and engagement.

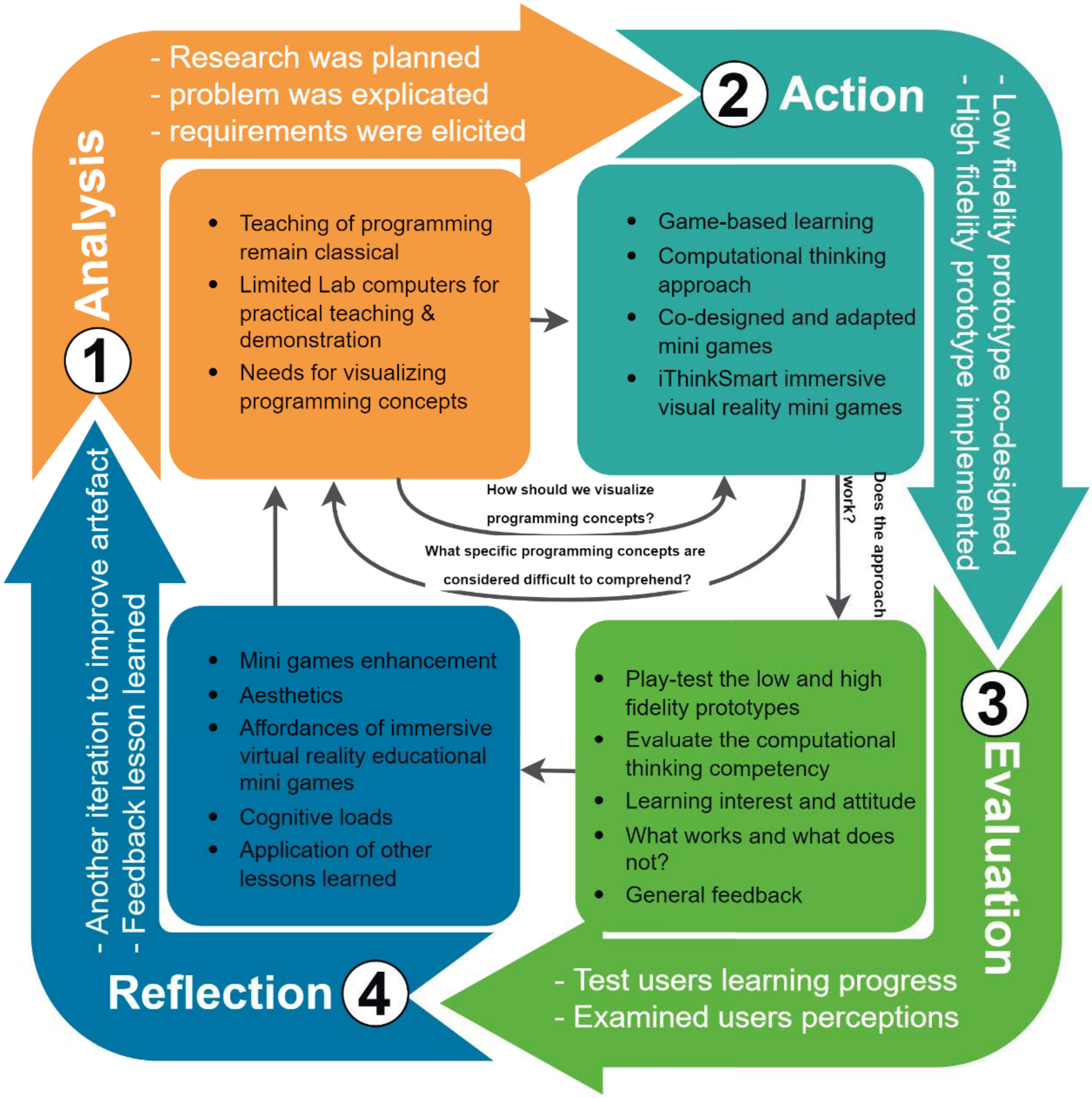


Figure 27. Circle of the design science research process demonstrated in this dissertation.

To examine whether the designed intervention addressed the specific problem explicated, a third stage of DSR – namely evaluation – was conducted. Usually, to evaluate an artefact developed using DSR, a formative approach could be adopted where the prototype can be quickly tested to see what works and what needs improving based on a functional perspective (Agbo, et al., 2021b). However, artefacts can also be evaluated using a summative approach where a detailed investigation can be conducted on, for example, the learning outcomes of an educational game (Venable et al., 2016). This study conducted an evaluation to playtest the VR game-based prototype with students from Federal University Lokoja and obtained their perceptions of the game experience. The main objective of the evaluation was to measure the efficacy of the developed iThinkSmart virtual mini-game prototype to ascertain whether it provided computational thinking competency to the players, improved their cognitive benefits, or inspired their interest and learning attitude toward learning computational thinking concepts.

The study revealed that students who played the iThinkSmart mini-games had higher computational thinking competency compared with the students who played an online game instead. This finding established the potential of using educational mini-games to teach computational thinking concepts, as demonstrated in previous studies (Devisch, et al., 2017; Hooshyar, et al., 2021). The findings of this study correspond with those of Hooshyar et al. (2020), who measured the computational thinking knowledge and skills of students who played a computer game compared with those who used the traditional learning approach to acquire the same skills. Similarly, this study’s outcomes were aligned with the findings of an experimental study that investigated students' computational thinking skills on concepts such as algorithmic thinking, pattern recognition, and simulation while playing an educational game (Hooshyar, et al., 2021).

Regarding the cognitive benefits that students who played the iThinkSmart mini-games gained, the results demonstrated a positive outcome. In other words, the findings of the experiment indicated that the educational tool supported students who played the mini-games in gaining moderately higher cognitive benefits compared with those who used the traditional approach. Although the statistical difference between the control group and the experimental group was moderate, the overall outcome suggested that the iThinkSmart prototype could support students in a Nigerian higher education institution in building their cognitive skills. Since computational thinking skills can improve students' programming knowledge (Rojas-López & García-Peñalvo, 2018; Rojas-López, et al. 2019), students in Nigeria who experience difficulty in passing their introductory programming courses could play iThinkSmart mini-games to gain higher cognitive benefits and to understand relevant concepts of programming. The link between cognitive benefits and the understanding of programming instruction has been studied for over three decades (Linn, 1985; Román-González, et al., 2017). Moreover, one study alluded to the potential benefits of using educational games to support computer science students in the Nigerian higher education context (Oyelere, et al., 2018). Our study agrees with the outcomes of Oyelere et al. (2018) as the iThinkSmart prototype was demonstrated to provide a better comprehension and understanding of computational thinking concepts.

Additionally, the evaluation of the iThinkSmart prototype revealed that the mini-games have the potential to stimulate students' learning interests. Through the iThinkSmart mini-games, students developed an interest in computational thinking concepts. In particular, the students who played iThinkSmart mini-games had a statistically higher interest in computational thinking concepts compared with those who played an online game. As the results indicated, the difference in learning interest among the two groups was moderately significant with an effect size of d = 4.0. As demonstrated in a previous study, students' interest in learning computational thinking concepts can be boosted by using educational games (Hooshyar, et al., 2020), which aligns with the current research findings.

Regarding learning attitude, the results of the evaluation of the iThinkSmart mini-games revealed that the majority of the students who played the mini-games exhibited a more positive attitude to learn computational thinking concepts. In fact, a similar study by Hooshyar et al.

(2021) reported an outcome that was consistent with this study's findings. According to Hooshyar et al., their educational game approach improved students' learning attitude toward learning computational thinking compared with the traditional approach.

In addition, users' experience was collected regarding their perception of the prototype and what in the system should be improved to further enhance their learning experience. Table 15 presents a summary of the aspects of the iThinkSmart prototype that required further development and improvement based on the students' feedback.

Table 15. Future development ideas that emanated from the evaluation.

Improvement aspects	Considerations for further development
User interface	The graphics and general esthetics of the game need to be improved. The experience of users suggested that the virtual world should almost be perceived as real since the users identified with the context being modeled in the virtual environment. In addition, the users seemed to like the sounds in the virtual environment as they immersed the player. However, some suggestions were made for how to improve the sounds, one of which was to consider developing the prototype to work with high-performance HMD (such as Oculus) instead of low-cost HMDs.
Interaction experience	Interaction within the virtual environment when playing the mini-games needs to be enhanced. Users suggested a more robust interaction with captivating navigation. A future improvement could leverage available sensors for hand and body tracking provided in sophisticated VR devices to enhance the interaction experience of users.
Gaming experience	The gaming experience also requires improvement through integrating more game elements. Currently, only one player can play the iThinkSmart mini-games. For future improvements, the introduction of more elements will be considered, such as avatars and multi-player games. Moreover, the game challenge needs to be enhanced with a motivating score and reward

	mechanism that is superior to what is currently implemented in the iThinkSmart mini-games.
Interoperability	The current prototype of iThinkSmart can be installed and played on an Android smartphone. One future improvement could be to make it platform-independent.

Overall, the students' feedback regarding what needs to be improved about the iThinkSmart prototype was useful for the further development of this research. After the evaluation, there was a need to reflect on the research outputs in terms of what worked and what did not to initiate another cycle of the iterative DSR process. In addition, lessons learned from the previous activities in the research were considered to ensure that the next iteration of the process produced better results to meet the explicated problem.

5 Discussion and conclusion

This dissertation focused on the design and development of a student-centered smart learning environment to facilitate computational thinking in the context of a Nigerian higher education institution. The overarching objective of this dissertation was primarily to investigate how to co-design an educational intervention to enhance students' learning in the context of a Nigerian higher education. In particular, this dissertation aimed to facilitate computational thinking education among students in a Nigerian University and to suggest guiding principles for mainstreaming computational thinking in higher education institutions in the Nigerian context.

Computational thinking skills are crucial for all learners in this digital age (Tedre & Denning, 2016; Carretero, et al., 2017). As part of the goal for every citizen to gain the digital literacy and creative thinking skills necessary for problem-solving, computational thinking education is being pursued in developed regions of the world such as Europe (Barendsen & Stoker, 2013; Carretero, et al., 2017). However, little effort is made to provide computational thinking skills in the context of Africa, particularly in Nigeria. As highlighted in Chapter 1, a lingering problem is the mass failure of students in introductory programming courses in Nigerian universities (Dasuki & Quaye, 2016; Sunday, et al., 2020) due to a lack of understanding of the topics.

Producing findings to facilitate students' understanding of programming concepts was the motivation for conducting this study. One approach for allowing students to gain an understanding of programming concepts is to facilitate their computational thinking skills through GBL (Devisch, et al., 2017; Chao, 2020). Therefore, this dissertation contributes to fostering computational thinking knowledge among students in the context of the developing region of Africa, particularly in Nigeria. A contextual mini-game (MoPaTH) with a Nigerian storyline was designed and implemented alongside the common mini-games (River Crossing and Tower of Hanoi), which are to be played in a virtual environment to provide an enhanced learning experience. The goal of this dissertation was to create an intervention that can engage students, motivate their learning, and immerse them in a smart learning environment to foster their understanding of the abstract concepts of computational thinking currently being taught through the traditional method. The smart learning environment was envisaged to deliver learning objects for micro learning using mini-games.

Therefore, this study designed and developed a VR-based smart learning environment named iThinkSmart to facilitate the understanding of computational thinking concepts through several mini-games. These mini-games were adapted or co-designed to deliver knowledge of computational thinking such as algorithmic thinking, problem abstraction, problem decomposition, pattern recognition, and recursive thinking. The developed smart learning environment supports students in gaining computational thinking competency through the immersion, interaction, immediacy, and engagement characteristics of VR technology (Andone, et al., 2018; Kamińska, et al., 2019; Radianti, et al., 2020). These characteristics of VR technology provide opportunities for learning affordances that are suitable for contemporary learners (Dalgarno & Lee, 2010). The relevance of VR technology to computing education was presented in Section 2.5, and Table 4 revealed several studies that demonstrated the use of VR mini-games in the educational context (Merchant et al., 2014), particularly in computing education.

This study employed the DSR methodology to design and develop the smart learning environment with two main outputs. The first was the iThinkSmart artefact, which enables users to learn computational thinking concepts through mini-games, and the second was the guiding principles for mainstreaming computational thinking in the context of a Nigerian higher education institution (see Section 5.3). In particular, the first prototype of iThinkSmart was designed and developed during the DSR process, whereas the guiding principles were developed by reflecting on the experience of the DSR process. As an additional output, this study enhanced the DSR framework by embedding the co-design process circle, which further strengthened the rigor of the DSR and provided an opportunity to develop a user-centered artefact, as described in Section 3.2.

5.1 Contributions of the dissertation

This section presents a summary of the dissertation's contributions to the knowledge base by highlighting key results that addressed the formulated RQs. RQ1 was as follows:

RQ1. What is the readiness of students and teachers in a Nigerian higher education institution for embracing smart learning environments to facilitate computational thinking education?

The problem with the teaching and learning method adopted in Nigerian universities for teaching computer science courses, such as introduction to computer programming courses, is well-known and was established in Section 1.1. The problem centers around a lack of understanding of programming courses, which has caused persistent failure among students as well as the attrition of the number of students pursuing computer science degrees. Based on the author's personal experience, this problem has lingered and affected many students negatively. To complicate matters, resources such as computers in university labs are grossly limited, which hinders students from learning on their own (Odediran, et al., 2015). Therefore, if these students could use their personal devices such as smartphones, this may provide a solution for addressing the identified problem.

However, addressing this contextual problem would not be possible without establishing the readiness of stakeholders to embrace a smart learning environment for teaching and learning. Furthermore, it was also crucial to investigate to what extent the technological, social, and economic situations in this context can facilitate the deployment of smart learning environments in higher education institutions. Therefore, providing an answer to RQ1 was crucial because otherwise the study's intention of developing a smart learning environment to facilitate computational thinking education might not be achieved or sustained (Oyelere, 2017).

This dissertation explicated the problem by following the JPDSR framework to provide answers to RQ1. Articles I and III were devoted to establishing the readiness and expectation of the stakeholders and the context in general for embracing smart learning environments. The findings from Article I suggested that both the teachers and students at the federal university and state polytechnic in Lokoja, Nigeria had no experience of using a smart learning environment to facilitate teaching or learning. However, they placed high expectations on the proposed educational intervention to support students' learning. In this digital age, the majority of young learners from Nigeria are technology savvy (Obienu & Amadin, 2018); therefore, they are eager to embrace a technology-mediated learning approach (Oyelere, et al., 2018).

In addition, the findings from Articles I and III revealed that the majority of students in the university and polytechnic owned a smartphone, which conformed with the finding of a previous study (Oyelere, et al., 2016). Smartphones are capable of facilitating ubiquitous learning (Hwang, 2014). In addition, the level of Internet facilities and number of mobile Internet subscribers in Nigeria (NCC, 2019) provide suitable conditions for the deployment of smart learning environments in Nigerian higher education institutions (Article III). The entire process of designing and developing the smart learning environment justifies the findings from Articles I and III as some of the research activities were conducted remotely between the author and students in an online environment (Article VI). The outcomes from this phase of the study are consistent with the study of Shonola et al., (2016) and Oyelere (2017), who studied, designed, and developed a mobile learning environment to support computing education in Nigerian higher education institutions.

RQ2. What are the key requirements to implement in a smart learning environment to support students' computational thinking education in the context of a Nigerian higher education institution?

To provide answers to RQ2, this dissertation conducted a series of studies by adopting several approaches. First, a user-centered study was conducted to understand the expectations of stakeholders regarding how they would want the proposed smart learning environment to address their learning problem (Articles I and III). Second, a systematic literature review was conducted to understand the pedagogical and technological requirements necessary for developing a 21st-century smart learning environment (Articles II and IV).

In the first approach, the author engaged the students and teachers to define user-specific requirements. For example, an outcome from Article III included a summary of users' expected features of a smart learning environment, consisting of an interactive user interface, learning guide, personalized learning environment, quick feedback mechanisms, and automated task scheduling. Experts have described the features of smart learning environments as including personalization, context-awareness, adaptivity, and autonomous learning (Spector, 2014; Kinshuk, et al., 2016; Abtar & Hassan, 2017), which align with the findings in Article III. The iThinkSmark prototype contains features of smart learning environments that offer the user high-level interaction with game elements in a virtual world, provide personalization through the HMD and hand controller, render learning guides and tutorials on how to play the game within the virtual world, and provide personalized feedback in terms of hints, warnings, and rewards.

In the second approach, a review of the literature focusing on computational thinking for programming education in higher education contexts revealed insights on how to develop a smart learning environment to facilitate computational thinking (Article II). Indeed, the literature review explained the potential of using the computational thinking approach to facilitate the understanding of programming concepts in the higher education context. Therefore, this dissertation opted to develop a smart learning environment to facilitate students' computational thinking skills instead of dealing with direct programming syntax. As demonstrated by Hurson and Sedigh (2010), teaching computational thinking concepts in introductory programming classes could provide more problem-solving skills than the traditional method of teaching programming. In addition, to make the smart learning environment engaging, immersive, and interactive for greater learning achievement, Article IV featured a systemic literature review of VR applications in computing education. This review revealed that deploying smart learning in a virtual world is the state of the art and can provide an enhanced learning experience. Therefore, the technological features of VR, including immersion, presence, interaction, engagement, and immediacy, were leveraged to develop the smart learning environment to facilitate computational thinking skills among students in the context of a Nigerian higher education institution.

After an extensive investigation of the literature, together with user engagement to gather the requirements for developing smart learning environments, a formal approach to system requirement specification was implemented (Wiegers & Beatty, 2013). During the requirements specification, a UML use case diagram, activity diagram, and sequence diagram were modeled to depict the scenario of a smart learning environment. This formal modeling of the smart learning environment was necessary to demonstrate how adapted mini-games can be concretely deployed within a virtual world, as reported in Article V. Therefore, the entire research on requirement elicitation and specification demonstrated how to arrive at the key features of a smart learning environment to facilitate computational thinking competency in the Nigerian context.

RQ3. How can student-centered VR-based smart learning environment be co-designed and implemented to facilitate computational thinking education in the context of a Nigerian higher education institution?

This dissertation addressed RQ3 by conducting a user-centered study where the author and the students co-designed a smart learning environment to facilitate their learning. The user-centered design of educational tools has been found to produce high-quality artefacts (Vaajakallio & Mattelmäki, 2014; De Jans, et al., 2017; Havukainen, et al., 2020). The author innovatively co-designed one of the mini-games deployed in the iThinkSmart prototype with computer science students in a Nigerian university. The OCD process introduced in Sections 3.1 and 3.2 was exploited to co-design with students in an online environment. As a proof of concept, the outcomes from the OCD process were demonstrated in Article VI. The intensive co-design process demonstrated vital contributions to the rigor of the DSR framework (Venable et al., 2012), where the OCD circle was embedded in the design, development, demonstration, and evaluation stages of the DSR to strengthen the entire methodology.

Furthermore, the study in Article VI established the implementation of a student-centered smart learning environment that allows students not only to gain knowledge through collaborative learning but also to peer-review themselves for an authentic assessment of their learning outcomes. This process of developing a smart learning environment provided an evidence-based study on how students contribute to the research, aimed at solving educational issues mediated with technology (Article VI). As presented by Kapinga (2020), co-designing an artefact with stakeholders from Sub-Saharan African countries can provide a great opportunity to understand the grassroots by uncovering indicators that may foster user acceptance and ease of use of the artefact.

In addition, the co-designed contextual mini-games were refined and modeled into a VR prototype application named iThinkSmart. The iThinkSmart VR prototype application immerses players into a contextual virtual world that mimics the expedition of an explorer through rivers or over land. During the expedition, players experience a famous river in Nigeria, namely River Niger, and a desert around the North-Central States. While the player enjoys the expedition and explores all of the interesting sites around Nigeria in the virtual world, they are confronted with computational thinking challenges that require them to provide a solution before advancing in the expedition. The author claimed that all of these contextual ideas harnessed together in the iThinkSmart prototype provide a suitable smart learning environment to facilitate computational thinking among students in the Nigerian context.

The findings from this study demonstrated the importance of developing smart learning environments to support students' learning in the Nigerian context. In particular, the evaluation experiment revealed that the iThinkSmart prototype can facilitate students' comprehension of computational thinking concepts, increase students' cognitive benefits, and spark their interest and attitude to learn more about computational thinking, which conforms with the findings from an earlier study (Merchant et al., 2014).

5.2 Study reflection

The global quest for computational thinking across all educational levels cannot be overemphasized (Tedre & Denning, 2016). As anticipated by Wing (2008), the pursuit for adopting computational thinking and mainstreaming its concepts into the educational curriculum has been advocated worldwide, as the concepts of computational thinking are known to be fundamental for preparing all citizens for the digital age (Carretero, et al., 2017). While developed countries have advanced their efforts to incorporate computational thinking concepts into the classroom to improve students' problem-solving skills (Barendsen & Stoker, 2013; Humphreys, 2015; Figueiredo & García-Peñalvo, 2017; Saqr, et al., 2021), it has become necessary for developing countries such as Nigeria to implant computational thinking into the curriculum to facilitate digital literacy and problem-solving skills as well as to prepare young students for future job challenges.

One may wonder why we must focus on computational thinking rather than programming education. As presented in Section 2.1, computational thinking is a foundational skill necessary to demonstrate programming knowledge. Indeed, experts have argued that students can display programming competence by applying computational thinking skills to solve problems (Csizmadia et al., 2015; Brackmann et al., 2016). In addition, the findings in Article VI (which showed that students found recursion more difficult to comprehend) confirmed our rationale for opting to design a smart learning environment to facilitate computational thinking. Moreover, the targeted users were mostly novices who had not been exposed to programming knowledge before enrolling at university.

In this section, the author reflects on how the smart learning environment was developed from the perspectives of pedagogy, technology, and the design process. In addition, the reflection presents how this dissertation connects to the existing knowledge and what contributions it makes.

For a learning environment to be meaningful (i.e., one that supports students' effective learning achievement), the pedagogy must be well-planned and adapted to suit the culture and context of the learners (Spector, 2014). According to the operational definition of a smart learning environment provided in Section 2.2, pedagogy is a component to consider when designing a smart learning environment. Therefore, this dissertation showcased the use of pedagogy such as GBL in the implementation of the iThinkSmart prototype. First, the computational thinking approach was discovered to be the foundational learning goal (Article II). Then, GBL was adopted as the driving pedagogy to deploy computational thinking concepts (Hooshyar, et al., 2021). Furthermore, contextual mini-games were identified as the micro learning approach for teaching computational thinking concepts (Awan, 2010; Devisch, et al., 2017). These interwoven approaches together formed a suitable pedagogy considered useful for the design and implementation of the smart learning environment.

Earlier experts have placed greater emphasis on the role of technology in smart learning environments (Hwang, 2014; Yassine, et al., 2016; Ronghuai, et al., 2017). As relevant as technology is to the development of a smart learning environment, Gros (2016) argued that technology itself cannot provide a suitable learning environment but must also connect with other components such as a relevant pedagogy, as demonstrated in this research. This dissertation leveraged VR as the main state-of-the-art technology to facilitate the deployment of the smart learning environment. Furthermore, VR in education has recently witnessed increasing growth in teaching and learning at different levels and disciplines (Radianti, et al., 2020; Article IV). VR applications can be installed on a smartphone and students can experience the virtual world by placing their smartphones in a simple HMD.

Because the students already possessed smartphones, this research developed a VR prototype application (iThinkSmart) that runs locally on the students' smartphones. The only additional device required to allow the students to experience full immersion in the virtual world is the HMD. With a very cheap HMD (approximately US$5), students can play iThinkSmart mini-games in a virtual world to gain computational thinking skills. This approach makes the educational intervention provided in this research a cost-effective technology for supporting the integration of a smart learning environment to facilitate computational thinking in higher education institutions in Nigeria.

Noteworthily, this dissertation contributes to the knowledge base through its creation of an innovative OCD sandwiched in a popular DSR framework, which allows a contemporary user study and the co-design of educational artefacts between researchers and learners in an online mode. This OCD process was also useful during the COVID-19 pandemic because face-to-face meetings for co-design between the researchers and students were not practical. Moreover, the OCD process itself created the opportunity for the students to gain computational thinking skills through collaborative learning (Article VI). In other words, the implementation of the OCD process facilitated the goal of this dissertation, which was essentially to support students in a Nigerian university in gaining computational thinking skills. As demonstrated in Article VI, the outcomes from the implementation of the OCD process for co-designing iThinkSmart mini-games were that the students who participated in the study

i. gained relevant knowledge that supported their creative thinking in contextual game scenarios and game elements related to computational thinking; and
ii. learned how to collaboratively co-design contextual mini-game prototypes of low fidelity and were also motivated to further design educational mini-games in their future studies.

As presented in Section 3.1, DSR is a pragmatic approach to be considered in user-centered research since it provides practical and straightforward activities to guide the project implementation (Johannesson & Perjons, 2012). However, to strengthen the rigor cycle required in the DRS project, a co-design process was introduced (Montero & Kapinga, 2019). Although the integration of the OCD process into the DSR framework still requires further validation, it impacted this dissertation and indeed the DSR framework in general. By implication, the OCD process may be useful for researchers using the DSR method in their project during a pandemic such as COVID-19. Specific recommendations on how to adapt OCD in a student-centered study can be found in Article VI, which may be helpful in terms of providing a reference that is evidential to stakeholders from Sub-Saharan Africa or those conducting research that touches this region.

5.3 Recommendations for practitioners and lessons learned

This dissertation demonstrated several approaches to mainstream computational thinking in Nigerian higher education institutions. Therefore, a list of guiding principles that emanate from this dissertation can be considered for mainstreaming computational thinking in a Nigerian higher education institution's curriculum to enhance students' learning experience, especially that of novices enrolled in programming education. These guiding principles are based on the DSR process, where two main outcomes emerged. According to Johannesson and Perjons (2014), two main outputs emanate from the DSR methodology, namely an artefact and guiding principles, models, or theory. The first output of this dissertation is an artefact called the iThinkSmart prototype, developed based on the DSR framework. The iThinkSmart prototype is a smart learning environment for supporting students' learning experience. The second output of this dissertation is the guiding principles, which contribute to the knowledge base regarding how to promote computational thinking in a higher education institution in the Nigerian context.

Consequently, the author discusses the guiding principles for mainstreaming computational thinking in a Nigerian higher education institution from four perspectives, namely (i) pedagogy, (ii) technology, (iii) learning theories, and (iv) design process. Furthermore, although these guiding principles are meant for designing smart learning environments and mainstreaming computational thinking in a Nigerian higher education institution, they can also be applied in different contexts. However, they are dynamic in nature and may be context-dependent, which might necessitate certain modifications and adaptations.

Guiding principle 1: Relevant pedagogy is required for mainstreaming computational thinking in the Nigerian context.

The need to harness relevant pedagogy in designing educational interventions was stressed in Section 5.2. Therefore, this research devised the pedagogical components of the smart learning environment to include GBL and mini-games to teach computational thinking concepts in the Nigerian higher education context. Hence, as a guiding principle, the author emphasizes the use of contextual mini-games and GBL as the core pedagogy for integrating computational thinking in the Nigerian higher education institution context.

Guiding principle 2: Readily available technology should be harnessed to facilitate computational thinking education in the Nigerian higher education context.

The use of technology to enhance teaching and learning process has gained huge research focus (Valtonen et al., 2022). Regarding the choice of technology, this research underscores a guiding principle that involves a thoughtful process of leveraging already existing technology so that students do not necessarily need to acquire expensive devices to use the smart learning environment. In other words, one must make the technology required to access the intervention as cheap as possible to motivate users' adoption. Such an approach can be a cost-effective way of integrating and mainstreaming computational thinking in the Nigerian context, where resources are limited, and most students may not be able to afford expensive devices.

Guiding principle 3: Relevant learning theories that support the development of computational thinking interventions should be considered.

The importance of embedding learning theories as foundations for building smart learning environments was emphasized in Section 2.4. According to Gros (2016), for a smart learning environment to be effective, researchers and educators must “develop new thoughts and pedagogy based on existing learning theories, such as constructivism, cognitive load theory, connectivism, and networked learning” (p. 6). Following the advice of Gros, this dissertation carefully associated theories to demonstrate the teaching and learning of computational thinking in the higher education context, and they were also relevant for designing a VR-based smart learning environment. Consequently, the present author asserts that one of the key guiding principles for designing a smart learning environment to mainstream computational thinking in a Nigerian higher education institution is to harness relevant learning theories to form the foundation for building the learning environment.

Guiding principle 4: A pragmatic research design process should be applied when developing an intervention to mainstream computational thinking education in the Nigerian context.

The design process involved in building a smart learning environment and integrating computational thinking in the Nigerian higher education context can essentially be considered a dynamic phenomenon. However, this dissertation rigorously developed a process that seemed to work even during a global pandemic. The OCD process was developed and embedded in the DSR, thereby strengthening the rigor of the DSR framework, as illustrated in Figure 11. As presented in Article VI, the OCD process underwent several rounds of contextual refinement and adaptation to allow successful user-centered research. Hence, the guiding principle aimed at designing a contextual smart learning environment for computational thinking in the Nigerian context is that educators and researchers must make the process student-centered by co-designing the artefact with them. In addition, researchers and designers should plan, explore, and evaluate how the contextual scenario fits the co-design process, whether in a face-to-face or online mode, to ensure that students acquire computational thinking skills from the co-design process.

In summary, the lessons learned from this dissertation are capable of providing a relevant guide that can support researchers and educators in Nigerian higher education institutions in designing and developing pedagogical processes, educational artefacts, and interventions to integrate computational thinking into their curriculum, thus facilitating critical thinking, creative thinking, and problem-solving skills (Mohaghegh & McCauley, 2016). Moreover, the guiding principles outlined in this section are based on author's experience of applying DSR with OCD, which are evidential. For policymakers and university administrators, this dissertation suggests that plans should be made regularly to train teachers in advanced knowledge on the application of technology in education and to improve their pedagogical skills, thereby improving their professional development to create smart learning environments. In addition, Internet infrastructure and other ICT tools should be improved in universities to create opportunities for hands-on practice and learning for students.

5.4 Research quality and limitations

This dissertation has demonstrated the design and development of a smart learning environment to foster computational thinking in the Nigerian higher education context following the DSR framework. The evaluation of the smart learning environment with computer science students in Nigeria revealed that the application can improve students' computational thinking competence in terms of algorithmic thinking, recursive thinking, problem decomposition, abstraction, and pattern recognition. Being a contextual and student-centered study, the goal was to solve an identified problem facing students in Nigerian universities. However, several limitations impacted the research quality and validity, which must be considered (Larsen et al., 2020).

First, a significant portion of the research was conducted from a distance, where the author was based in Europe while the students were studying in Nigeria. This distance research placed some limitations on the study because the physical contact the author required to conduct several research activities with the students was limited. This limitation was due to a lack of sufficient funding to facilitate the research in terms of travel costs and other expenses necessary for the author and students to meet regularly. Another reason why the research was somewhat conducted from a distance was the outbreak of COVID-19 pandemic in December 2019, where traveling was restricted, and face-to-face gatherings of people were not feasible or encouraged. These challenges also caused the research to narrow the stakeholders to students instead of the initial intention of including both students and teachers. Notwithstanding, the author innovatively designed an alternative approach to mitigate this limitation by creating an OCD process using several online platforms, where regular meetings and research activities were completed with the students.

Second, the study focused on computer science students in a Nigerian university. Even though all students are meant to acquire computational thinking skills, this study concentrated on students from a specific discipline and in a university due to a lack of funds for broadening the scope. Furthermore, all of the evaluations were limited to students from one field, a situation that created another limitation of the study. Therefore, the generalizability of the outcomes from this research is greatly limited. Indeed, only one university was the focus of the evaluation of the developed artefact; hence, one cannot draw a blanket conclusion about the efficacy of the application as further research is required to justify the claim. Notwithstanding, the outcome from this study, especially the guiding principles can be extended and applied in another context.

Third, the DSR framework adopted in this study requires an iterative process of evaluating and improving outputs from the research in order to refine and improve on them. Unfortunately, this aspect of iteration through the activities of DSR was not sufficiently done. For example, further evaluation of the iThinkSmart prototype and its improvement based on the outcome of a previous evaluation is required. Moreover, a rigorous formative evaluation of the prototype is required, the outcome of which can be feedback into another circle of the system development based on the iterative DSR process. In addition, several methods and processes developed in this research as theoretical contributions, such as OCD, require validation.

Fourth, although the developed iThinkSmart prototype does not require a large amount of technology or devices, students would still need a simple smartphone and an HMD to fully experience the immersion and presence of VR while playing the mini-games. Unfortunately, not all students can afford such devices. Therefore, the effective use of the application by the students to independently learn computational thinking concepts could be limited, unless the university purchases enough HMDs to serve the students.

However, these limitations also created several opportunities and lessons through which the author learned from this study. For instance, the author learned how to develop an online strategy for conducting a user-centered study, where all the activities conducted in an online mode recorded success, as demonstrated in Article VI.

5.5 Future work

Although the limitations encountered in the research are highlighted in Section 5.4, they, however, create huge opportunities for the author's future studies. For example, further evaluation (both formative and summative) with other stakeholders such as educators and teachers in the future would provide more evidence to create an impact in mainstreaming computational thinking in Nigerian higher education institutions.

This study presented several concepts and processes that must be improved, evaluated, and even standardized. Some of these concepts are merely based on associating well-known theories and pedagogies, or an entirely new process such as OCD, which has not been rigorously tested. The author will invest more time into conducting future studies on the validation of these concepts, especially the OCD process such that it hopefully becomes a standard method. This future work is necessary, particularly now that most studies have adopted an online mode due to the pandemic. Moreover, researchers would require a verified process for conducting a user-centered study in an online environment in such a pandemic era.

The need to demonstrate whether the iThinkSmart prototype would be accepted widely in Nigeria’s higher education institutions for learning computational thinking concepts will form part of the author’s future work. The demonstration of the iThinkSmart prototype in different universities and disciplines could create great opportunities to inculcate creative thinking and problem-solving skills among students in Nigeria. In addition, wider demonstrations of the tool could spark stakeholders' interest in integrating computational thinking as mainstream education in Nigeria’s higher education institutions. Therefore, the author anticipates conducting future research in this direction and widening the scope of stakeholders to include teachers.

In addition, more contextual mini-games will be developed in the future to foster computational thinking in different fields and topics. The author plans to conduct further research to demonstrate other abstract topics that students in Nigerian higher education institutions find difficult and to model them in a visualized 3D application, where students can gain understanding in an immersive virtual world. Moreover, the author plans to demonstrate an improved version of the VR application with students from another context such as Finland to gain insight into students' learning process and how they react differently or otherwise to learning computational thinking through VR-game-based application compared to Nigerian students.

Meanwhile, this study has demonstrated how VR mini-games can help students to understand abstract topics that proved difficult to comprehend when taught or learned through traditional methods.

6 Bibliography

Abdullahi, H. A., Enejoh, O. T., & Lola, J. (2019). The influence of monuments on the Socio-Economic development of Kogi state. *Epra International Journal of Multidisciplinary Research, 5*(12), 106-111.

Abtar, D. S., & Hassan, M. (2017). In Pursuit of smart learning environments for the 21st century. *In-Progress Reflection No. 12 on Current and Critical Issues in Curriculum*, *Learning and Assessment,* UNESCO Bureau of Education, https://unesdoc.unesco.org/ark:/48223/pf0000252335/PDF/252335eng.pdf.multi

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Tukiainen, M. (2021a). Scientific production and thematic breakthroughs in smart learning environments: a bibliometric analysis. *Smart Learning Environments, 8*(1), 1-25.

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Tukiainen, M. (2021b). iThinkSmart: Immersive virtual reality mini games to facilitate students' computational thinking skills. In *Koli Calling'21: 21st Koli Calling International Conference on Computing Education Research*, November 18-21, ACM. https://dl.acm.org/doi/abs/10.1145/3488042.3489963

Agbo, F. J., Olawumi, O., Balogun, O. S., Sanusi, I. T., Olaleye, S. A., Sunday, K., Kolog, E. A., Atsa'am, D. D., Adusei-Mensah, F., Adegbite, A. & Ipeayeda, F. W. (2021c). Investigating students' perception towards the use of social media for computing education in Nigeria. *Journal of Information Systems Education, 32*(3), 213-227.

Agbo, F. J., Yigzaw, S. T., Sanusi, I. T., Oyelere, S. S., & Mare, A. H. (2021d). Examining theoretical and pedagogical foundations of computational thinking in the context of higher education. In *2021 IEEE Frontiers in Education Conference (FIE)* (pp. 1-8). IEEE.

Agbo, F., Olawumi, O., Oyelere, S. S., Kolog, E., Olaleye, S., Agjei, R., Ukpabi, D. C., Yunusa, A. A., Gbadegeshin, S. A., Awoniyi, L., Erinle, K., Mogaji, E., Silas, A. D., Nwachukwu, D., Olawuni, A. (2020). Social media usage for computing education: The effect of tie strength and group communication on perceived learning outcome. *International Journal of Education and Development using Information and Communication Technology, 16*(1), 5-26.

Agbo, F. J., & Oyelere, S. S. (2019). Smart mobile learning environment for programming education in Nigeria: adaptivity and context-aware features. In *Intelligent Computing-Proceedings of the Computing Conference* (pp. 1061-1077). Springer, Cham.

Ahmadi, A. A., & Lukman, A. A. (2015). Issues and prospects of effective implementation of new. *Journal of Education and Practice Secondary School Curriculum in Nigeria, 6*(34), 29-39.

Aho, A. (2012). Computation and computational thinking. *The Computer Journal, 55*(7), 832-835.

Akin, O. C., Matthew, F., & Comfort, D. (2014). The impact and challenges of cloud computing adoption on public universities in Southwestern Nigeria. *International Journal of Advanced Computer Science and Applications (IJACSA), 5*(8), 13-19.

Akinola, O. S., & Nosiru, K. A. (2014). Factors influencing students' performance in computer programming: A fuzzy set operations approach. *International Journal of Advances in Engineering & Technology, 7*(4), 1141-1149.

Amber, S. (2014). What’s motivation got to do with it? A survey of recursion in the computing education literature. Chicago: College of Computing and Digital Media, DePaul University.

Anderson, T., & Shattuck, J. (2012). Design-based research: A decade of progress in education research? *Educational researcher, 41*(1), 16-25.

Andone, D., Vert, S., Frydenberg, M., & Vasiu, R. (2018). Open virtual reality project to improve students' skills. *IEEE 18th International Conference on Advanced Learning Technologies (ICALT)* (pp. 6-10). IEEE.

Anthony, E. (2003). Computing education in academia: Toward differentiating the disciplines. *Proceedings of the 4th Conference on Information Technology Curriculum* (pp. 1-8). USA: ACM.

Anyango, J. T., & Suleman, H. (2018). Teaching programming in Kenya and South Africa: What is difficult and is it universal? *Proceedings of the 18th Koli Calling International Conference on Computing Education Research* (pp. 1-2). Koli: ACM.

Anyanwu, J. A. (1978). Computer science education in a developing nation. *The SIGCSE/CSA Technical Symposium on Computer Science Education* (pp. 37-40). ACM.

Apiola, M., & Sutinen, E. (2021). Design science research for learning software engineering and computational thinking: Four cases. *Computer Applications in Engineering Education, 29*(1), 83-101.

Arrington Jr, C., Wilson, D. M., & Lehmann, L. (2011). Improving performance and retention in computer science courses using a virtual game show. *Proceedings of the 49th Annual Southeast Regional Conference* (pp. 320-321).

Awan, M. A. (2010). Thinking strategies used while engaged in solving the Tower of Hanoi, the River-crossing and Find the Pattern puzzles. Dubai: Doctoral dissertation, The British University in Dubai (BUiD).

Bakar, M. A., Mukhtar, M., & Khalid, F. (2019). The development of a visual output approach for programming via the application of cognitive load theory and constructivism. *Development, 10(*11), 305-312.

Balogun, O.S., Oyelere, S.S., & Refer'am, D.D. (2019). Data analytics is the performance of computing students. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1-2). ACM.

Barbero, G., Gómez-Maureira, M. A., & Hermans, F. F. (2020). Computational thinking through design patterns in video games. In *International Conference on the Foundations of Digital Games* (pp. 1-4).

Barendsen, E., & Stoker, I. (2013). Computational thinking in CS teaching materials: A pilot study. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research* (pp. 199-200).

Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design science research contributions: finding a balance between artifact and theory. *Journal of the Association for Information Systems, 19*(5), 358-376.

Basu, S., Disalvo, B., Rutstein, D., Xu, Y., Roschelle, J., & Holbert, N. (2020). The role of evidence centered design and participatory design in a playful assessment for computational thinking about data. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 985-991). ACM.

Béjat, M. D. C. & Vera, M. D. M. S. (2022). Change from face-to-face to virtual mode during confinement due to COVID-19: Perceptions of university students. *RIED. Revista Iberoamericana de Educación a Distancia, 25*(1), 243-260. https://doi.org/10.5944/ried.25.1.30623

Ben-Ari, M. (1998). Constructivism in Computer Science Education. USA: ACM.

Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching, 20*(1), 45-73.

Berns, C., Chin, G., Savitz, J., Kiesling, J., & Martin, F. (2019). Myr: A web-based platform for teaching coding using VR. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 77–83). Minneapolis, MN, USA: ACM.

Bolivar, S., Perez, D., Carrasquillo, A., Williams, A. S., Rishe, N. D., & Ortega, F. R. (2019). 3D interaction for computer science educational VR game. *International Conference on Human-Computer Interaction* (pp. 408-419). Springer, Cham.

Bonani, A., Del Fatto, V., Dodero, G., Gennari, R., & Raimato, G. (2017). Participatory design of tangibles for graphs: A small-scale field study with children. *Conference on Smart Learning Ecosystems and Regional Development* (pp. 161-168). Springer, Cham.

Bonsignore, E., Hansen, D., Pellicone, A., Ahn, J., Kraus, K., Shumway, S., Kaczmarek, K., Parkin, J., Cardon, J., Sheets, J., Holl-Jensen, C., & Koepfler, J. (2016). Traversing transmedia together: Co-designing an educational alternate reality game for teens, with teens. *Proceedings of the 15th International Conference on Interaction Design and Children*, (pp. 11-24).

Borg, J., Scott-Young, C. M., & Turner, M. (2019). Smarter education: Leveraging stakeholder inputs to develop work ready curricula. *Smart Education and e-Learning 2019* (pp. 51-61). Singapore: Springer.

Bouali, N., Nygren, E., Oyelere, S. S., Suhonen, J., & Cavalli-Sforza, V. (2019). Imikode: A VR game to introduce OOP concepts. *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1-2). Koli: ACM.

Brackmann, C., Barone, D., Casali, A., Boucinha, R., & Muñoz-Hernandez, S. (2016). Computational thinking: Panorama of the Americas. *In 2016 international symposium on Computers in Education (SIIE)* (pp. 1-6). IEEE.

Brady, C., Orton, K., Weintrop, D., Anton, G., Rodriguez, S., & Wilensky, U. (2016). All roads lead to computing: Making, participatory simulations, and social computing as pathways to computer science. *IEEE Transactions on Education, 60*(1), 59-66.

Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In *International conference on intelligent tutoring systems* (pp. 261-269). Springer, Berlin, Heidelberg.

Brusilovsky, P., Specht, M., & Weber, G. (1995). Towards adaptive learning environments. In *GISI 95* (pp. 322-329). Springer, Berlin, Heidelberg.

Butgereit, L. (2016). Teaching recursion through games, songs, stories, directories and hacking. *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)* (pp. 401-407). South Africa: IEEE.

Butt, A. L., Kardong-Edgren, S., & Ellertson, A. (2010). Using game-based virtual reality with haptics for skill acquisition. *Clinical Simulation in Nursing, 16*, 25-32.

Cabero-Almenara, J., & Llorente-Cejudo, C. (2020). COVID-19: Radical transformation of digitization in university institutions. *Campus Virtuales, 9*(2), 25-34.

Carretero, S., Vuorikari, R., & Punie, Y. (2017). DigComp 2.1: The digital competence framework for citizens with eight proficiency levels and examples of use. Seville site: Joint Research Centre. https://publications.jrc.ec.europa.eu/repository/handle/JRC106281

Carstensen, A. K., & Bernhard, J. (2016). Design science research as an approach for engineering education research. *The 12th International CDIO Conference* (pp. 1072-1081). Turku: Turku University of Applied Sciences.

Chaichumpa, S., & Temdee, P. (2018). Assessment of student competency for personalised online learning using objective distance. *International Journal of Innovation and Learning, 23*(1), 19-36.

Chaichumpa, S., Wicha, S., & Temdee, P. (2021). Personalized learning in a virtual learning environment using modification of objective distance. *Wireless Personal Communications*, 1-18. https://doi.org/10.1007/s11277-021-08126-7

Chao, G. C. (2020). Development of the motivation and engagement in virtual reality Chinese language learning questionnaire (MEVRCLQ). *International Symposium on Educational Technology (ISET)* (pp. 67-72). IEEE.

Cheung, B., Hui, L., Zhang, J., & Yiu, S. M. (2003). SmartTutor: An intelligent tutoring system in web-based adult education. *Journal of Systems and Software, 68*(1), 11-25.

Cheung, S. K., Li, R., Phusavat, K., & Paoprasert, N. (Eds.). (2020). Blended learning. Education in a smart learning environment: *13th International Conference, ICBL 2020*, Bangkok, Thailand, August 24–27, 2020, Proceedings (Vol. 12218). Springer Nature

Cobb, P., Confrey, J., DiSessa, A., Lehrer, R., & Schauble, L. (2003). Design experiments in educational research. *Educational researcher, 32*(1), 9-13.

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Hillsdale, New Jersey: Lawrence Earlbaum Associates.

Cotic, N., Plazar, J., Istenic Starcic, A., & Zuljan, D. (2020). The effect of outdoor lessons in natural sciences on students' knowledge, through tablets and experiential learning. *Journal of Baltic Science Education, 19*(5), 747-763.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking: A guide for teachers. UK: Dsigital Schoolhouse. https://community.computingatschool.org.uk/files/6695/original.pdf

Dalgarno, B., & Lee, M. J. (2010). What are the learning affordances of 3-D virtual environments? *British Journal of Educational Technology, 41*(1), 10-32.

Daniel, S. J. (2020). Education and the COVID-19 pandemic. *PROSPECTS*, 49, 91-96. https://doi.org/10.1007/s11125-020-09464-3

Dasuki, S., & Quaye, A. (2016). Undergraduate students' failure in programming courses in institutions of higher education in developing countries: A Nigerian perspective. *The Electronic Journal of Information Systems in Developing Countries, 76*(1), 1-18.

de los Ríos, S., García-Betances, R. I., Páramo, M., Cabrera-Umpiérrez, M. F., Vancells, M., Garolera, M., Kaźmierski, J. & Waldmeyer, M. T. A. (2019). Designing an ICT solution for the empowerment of functional independence of people with mild cognitive impairment: Findings from co-design sessions with older people. In *International Conference on Smart Homes and Health Telematics* (pp. 18-26). Springer, Cham.

De Jans, S., Van Geit, K., Cauberghe, V., Hudders, L., & De Veirman, M. (2017). Using games to raise awareness: How to co-design serious mini-games? *Computers & Education, 110*(1), 77-87.

De Silva, L., Goonetillake, J., Wikramanayake, G., Ginige, A., Ginige, T., Vitiello, G., Sebillo, M., Di Giovanni, P., Tortora, G. & Tucci, M. (2014). Design science research based blended approach for usability driven requirements gathering and application development. In *2014 IEEE 2nd International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE)* (pp. 17-24). IEEE.

Devisch, O., Gugerell, K., Diephuis, J., Constantinescu, T., Ampatzidou, C., & Jauschneg, M. (2017). Mini is beautiful. Playing serious mini-games to facilitate collective learning on complex urban processes. *Interaction Design and Architecture (s) Journal,* (35), 141-157.

Devisch, O., Huybrechts L., De Ridder, R. (2019). Participatory Design Theory: Using Technology and Social Media to Foster Civic Engagement (1st ed.). Routledge.

Dindler, C., Smith, R., & Iversen, O. S. (2020). Computational empowerment: Participatory design in education. *CoDesign, 16*(1), 66-80.

Dogo, E. M., Salami, A., & Salman, S. (2013). Feasibility analysis of critical factors affecting cloud computing in Nigeria. *International Journal of Cloud Computing and Services Science, 2*(4), 276-287.

Ekundayo, H. T., & Ajayi, I. A. (2009). Towards effective management of university education in Nigeria. *International NGO Journal, 4*(8), 342-347.

El-Glaly, Y., Shi, W., Malachowsky, S., Yu, Q., & Krutz, D. E. (2020). Presenting and evaluating the impact of experiential learning in computing accessibility education. In *2020 IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)* (pp. 49-60). IEEE.

Elkind, D. (2004,). The problem with constructivism. *The educational forum, 68(*4), 306-312.

Esper, S., Wood, S. R., Foster, S. R., Lerner, S., & Griswold, W. G. (2014). Codespells: How to design quests to teach java concepts. *Journal of Computing Sciences in Colleges, 29*(4), 114-122.

Ferrone, H. (2019). Learning C# by developing games with Unity 2019: Code in C# and build 3D games with Unity. (4, Ed.) Birmingham: Packt Publishing Ltd.

Fidalgo-Blanco, Á., Sein-Echaluce, M. L., & García-Peñalvo, F. J. (2020). Hybrid flipped classroom: adaptation to the COVID situation. In F. J. García-Peñalvo (Ed.), *Proceedings TEEM'20. Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*. ACM. https://doi.org/10.1145/3434780.3436691

Figueiredo, J., & García-Peñalvo, F. J. (2017). Improving computational thinking using follow and give instructions. In J. M. Dodero, M. S. Ibarra Sáiz, & I. Ruiz Rube (Eds.), *Fifth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'17)* (pp. Article 3). ACM. https://doi.org/10.1145/3144826.3145351

Figueiredo, J. A. Q., & García-Peñalvo, F. J. (2021). Teaching and learning strategies for introductory programming in university courses. In *proceedings TEEM'21. Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 746-751). ACM. https://doi.org/10.1145/3486011.3486540

Ge, J., Smyth, R. E., Searle, M., Kirkpatrick, L., Evans, R., Elder, A., & Brown, H. (2021). Perspectives from students: How to tame the chaos and harness the power of technology for learning. *Brock Education Journal, 1*(30), 74-94.

Ginat, D. (2004). Do senior CS students capitalize on recursion? *ACM SIGCSE Bulletin*, *36*(3), 82-86.

Gital, A. Y., & Zambuk, F. U. (2011). Cloud computing: Solution to ICT in higher education in Nigeria. *Advances in Applied Science Research, 2*(6), 364-369.

Gomez, K., Kyza, E. A., & Mancevice, N. (2018). Participatory design and the Learning Sciences. International Handbook of the Learning sciences (p. 9). New York: Taylor and Francis.

García-Peñalvo, F. J., & Mendes, J. A. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior, 80*, 407-411. https://doi.org/10.1016/j.chb.2017.12.005

García-Holgado, A., & García-Peñalvo, F. J. (2019). Validation of the learning ecosystem metamodel using transformation rules. *Future Generation Computer Systems, 91*, 300-310. https://doi.org/10.1016/j.future.2018.09.011

García-Peñalvo, F. J., Casado-Lumbreras, C., Colomo-Palacios, R., & Yadav, A. (2020). Smart learning. *Applied Sciences, 10*(9), 1-7. https://doi.org/10.3390/app10196964

García-Peñalvo, F. J., Corell, A., Abella-García, V., & Grande-de-Prado, M. (2021a). Recommendations for mandatory online assessment in higher education during the COVID-19 pandemic. In D. Burgos, A. Tlili, & A. Tabacco (Eds.), *Radical Solutions for Education in a Crisis Context. COVID-19 as an Opportunity for Global Learning* (pp. 85-98). Springer Nature. https://doi.org/10.1007/978-981-15-7869-4_6

García-Peñalvo, F. J., García-Holgado, A., Vázquez-Ingelmo, A., & Sánchez-Prieto, J. C. (2021b). Planning, communication and active methodologies: Online assessment of the software engineering subject during the COVID-19 crisis. *RIED. Revista Iberoamericana de Educación a Distancia, 24*(2). https://doi.org/10.5944/ried.24.2.27689

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly, 37*(2), 337-355.

Gros, B. (2016). The design of smart educational environments. *Smart Learning Environments, 3(*1), 1-11.

Gros, B., & García-Peñalvo, F. J. (2016). Future trends in the design strategies and technological affordances of e-learning. In M. Spector, B. B. Lockee, & M. D. Childress (Eds.), Learning, Design, and Technology. An International Compendium of Theory, Research, Practice, and Policy (pp. 1-23). Springer International Publishing. https://doi.org/10.1007/978-3-319-17727-4_67-1

Grover, S., Cateté, V., Barnes, T., Hill, M., Ledeczi, A., & Broll, B. (2020). First principles to design for online, synchronous high school CS teacher training and curriculum co-design. In Koli Calling'20: *Proceedings of the 20th Koli Calling International Conference on Computing Education Research* (pp. 1-5). ACM.

Guo, H. (2018). Application of a Computer-Assisted Instruction System Based on Constructivism. *International Journal of Emerging Technologies in Learning, 13*(4), 33-44.

Gunn, C. L., & Raven, J. (2017). Smart education: Introducing active learning engineering classrooms in the Middle East. *2017 Fourth HCT Information Technology Trends (ITT)* (pp. 1-4). IEEE.

Han, O., & Kim, J. (2017). A study on the improvement to the effectiveness of education in smart learning environment. *Proceedings of the 12th APIC-IST*, (pp. 25-28).

Han, O., & Kim, J. (2018). Examining the relationship between educational effectiveness and computational thinking in smart learning environment. *Journal of Internet Computing and Services (JICS), 19*(2), 57-67.

Havukainen, M., Laine, T. H., Martikainen, T., & Sutinen, E. (2020). A case study on co-designing digital games with older adults and children: Game elements, assets, and challenges. *The Computer Games Journal, 9*(2), 63-188.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.

Hevner, A., & Chatterjee, S. (2010). Design science research in information systems. In *Design research in information systems* (pp. 9-22). Springer, Boston, MA.

Hjelmfors, L., Strömberg, A., Friedrichsen, M., Sandgren, A., Mårtensson, J., & Jaarsma, T. (2018). Using co-design to develop an intervention to improve communication about the heart failure trajectory and end-of-life care. *BMC Palliative Care, 17*(1), 1–10.

Holopainen, J., Mattila, O., Pöyry, E., & Parvinen, P. (2020). Applying design science research methodology in the development of virtual reality forest management services. *Forest Policy and Economics*, 116, 1-9. https://doi.org/10.1016/j.forpol.2020.102190

Hooshyar, D., Malva, L., Yang, Y., Pedaste, M., Wang, M., & Lim, H. (2021). An adaptive educational computer game: Effects on students' knowledge and learning attitude in computational thinking. *Computers in Human Behavior, 114*(106575), 1-13.

Hooshyar, D., Pedaste, M., Yang, Y., Malva, L., Hwang, G., Wang, M., Lim, H., & Delev, D. (2020). From gaming to computational thinking: An adaptive educational computer game-based learning approach. *Journal of Educational Computing Research*, 1-27. https://doi.org/10.1177/0735633120965919.

Horst, R., Naraghi-Taghi-Off, R., Diez, S., Uhmann, T., Müller, A., & Dörner, R. (2019). FunPlogs - A serious puzzle mini-game for learning fundamental programming principles using visual scripting. *International Symposium on Visual Computing* (pp. 494–504). Berlin, Germany: Springer.

Horváth, I. (2020). Personalized learning opportunity in 3D VR. *11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)* (pp. 000425-000430). IEEE.

Huang, R., Yang, J., & Hu, Y. (2012). From digital to smart: The evolution and trends of learning environment. *Open Education Research, 1*(1), 75-84.

Hudson, S., Matson-Barkat, S., Pallamin, N., & Jegou, G. (2019). With or without you? Interaction and immersion in a virtual reality experience. *Journal of Business Research, 100*, 459-468.

Humphreys, S. (2015). Computing at school. Retrieved March 27, 2021, from https://www.computingatschool.org.uk/computationalthinking

Hurson, A. R., & Sedigh, S. (2010). Transforming the instruction of introductory computing to engineering students. *2010 IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*, (pp. 1-19). Ireland.

Hwang, G. J., & Chang, H. F. (2011). A formative assessment-based mobile learning approach to improving the learning attitudes and achievements of students. *Computers & Education, 56*(4), 1023-1031.

Hwang, G.-J. (2014). Definition, framework and research issues of smart learning environments - a context-aware ubiquitous learning perspective. Springer Open Journal.

Hwang, G.-J., & Fu, Q.-K. (2020). Advancement and research trends of smart learning environments in the mobile era. I*nternational Journal of Mobile Learning and Organisation, 14*(1), 114-129. https://doi.org/10.1504/IJMLO.2020.103911

IBM SPSS Statistics. (2020). https://www.ibm.com/support/pages/spss-statistics-27-now-available.

Idemudia, E. C., Dasuki, S. I., & Ogedebe, P. (2017). Factors that influence students' programming skills: a case study from a Nigerian university. *International Journal of Quantitative Research in Education, 3*(4), 277-291.

Igbokwe, C. O. (2015). Recent curriculum reforms at the basic education. *American Journal of Educational Research Level in Nigeria Aimed at Catching Them Young to Create Change, 3*(1), 31-37.

Ito, H., Langerman, S., & Yoshida, Y. (2015). Generalized river crossing problems. *Theory of Computing Systems, 56*(12), 418-435.

Jill, A., & Carol, S.-S. (2004). Mobile learning anytime everywhere. UK: Learning and Skills Development Agency. https://stu.westga.edu/~bthibau1/MEDT%208484-%20Baylen/mLearn04_papers.pdf

Johannesson, P., & Perjons, E. (2012). A design science primer. CreateSpace Independent Publishing Platform.

Johannesson, P., & Perjons, E. (2014). An introduction to design science. Springer.

Jong, M. S., Shang, J., Lee, F. L., & Lee, J. H. (2010). VISOLE: A constructivist pedagogical approach to game-based learning. *Collective Intelligence and E-learning 2.0: Implications of Web-Based Communities and Networking* (pp. 185-20). IGI Global.

Kamińska, D., Sapiński, T., Wiak, S., Tikk, T., Haamer, R., Avots, E., Helmi, A., Ozcinar, C., & Anbarjafari, G. (2019). Virtual reality and its applications in education: Survey. *Information, 10*(10), 1-20.

Kapinga, A. F. (2020). *Mobile technology for empowerment: a case of women entrepreneurs in the food processing sector in Tanzania* (Doctoral dissertation, University of Eastern Finland). https://erepo.uef.fi/handle/123456789/22410

Kapinga, A. F., Suero Montero, C., & Mbise, E. R. (2019). Mobile marketing application for entrepreneurship development: Codesign with women entrepreneurs in Iringa, Tanzania. *The Electronic Journal of Information Systems in Developing Countries, 85*(2), 1-15.

Karumbaiah, S., Dabholkar, S., Shim, J., Yoon, S., Chandy, B., & Ye, A. (2019). Using participatory design to facilitate in-service teacher learning of computational thinking, In K. Lund, G. P. Niccolai, E. Lavoué, C. Hmelo-Silver, G. Gweon, & M. Baker (Eds.), *A Wide Lens: Combining Embodied, Enactive, Extended, and Embedded Learning in Collaborative Settings, 13th International Conference on Computer Supported Collaborative Learning (CSCL)* (pp. 827-828). France: International Society of the Learning Sciences.

Kavanagh, S., Luxton-Reilly, A., Wuensche, B., & Plimmer, B. (2017). A systematic review of Virtual Reality in education. *Themes in Science and Technology Education, 10*(2), 85-119.

Kennedy, A., Cosgrave, C., Macdonald, J., Gunn, K., Dietrich, T., & Brumby, S. (2021). Translating co-design from face-to-face to online: An Australian primary producer project conducted during COVID-19. *International Journal of Environmental Research and Public Health, 18*(8), 1-14.

Kim, S., Song, S., & Yoon, Y. (2011). Smart learning services based on smart cloud computing. *Sensors, 11*(8), 7835-7850.

Kinshuk, Chen, N. S., Cheng, I. L., & Chew, S. W. (2016). Evolution is not enough: Revolutionizing current learning environments to smart learning environments. *International Journal of Artificial Intelligence in Education, 26*(2), 561-581.

Kohonen, I., Arja K-L., Spoof, S-K. (2019). The ethical principles of research with human participants and ethical review in the human sciences in Finland. https://tenk.fi/sites/default/files/2021-01/Ethical_review_in_human_sciences_2020.pdf

Kolb, D. A. (2014). Experiential learning: Experience as the source of learning and development (2nd ed.). New Jersey: FT Press.

Kolog, E. A. (2017). *Contextualising the application of human-language technologies for counselling* (Doctoral dissertation, University of Eastern Finland). https://erepo.uef.fi/handle/123456789/18481

Koper, R. (2014). Conditions for effective smart learning environments. *Smart Learning Environments, 1*(5), 1-17.

Ko, A. (n.d.). Computing education research FAQ. Retrieved May 25, 2021, from https://faculty.washington.edu/ajko/cer#:~:text=Computing%20education%20research%20(CER)%2C,and%20teach%20computing%2C%20broadly%20construed.

Lai, Y. H., Chen, S. Y., Lai, C. F., Chang, Y. C., & Su, Y. S. (2019). Study on enhancing AIoT computational thinking skills by plot image-based VR. *Interactive Learning Environments*, 1-14. https://doi.org/10.1080/10494820.2019.1580750

Larsen, K. R., Lukyanenko, R., Mueller, R. M., Storey, V. C., VanderMeer, D., Parsons, J., & Hovorka, D. S. (2020). Validity in Design Science Research. In *International Conference on Design Science Research in Information Systems and Technology* (pp. 272-282). Springer, Cham.

Lau, F. (1997). A review on the use of action research in information systems studies. In *Information Systems and Qualitative Research* (pp. 31-68). Springer, Boston, MA.

Lawler, R. W., & Yazdani, M. (1987). *Artificial intelligence and education: Learning environments and tutoring systems* (Eds.). Intellect Books.

Lee, K.J., Roldan, W., Zhu, T.Q., Kaur Saluja, H., Na, S., Chin, B., Zeng, Y., Lee, J.H. and Yip, J. (2021). The show must go on: A conceptual model of conducting synchronous participatory design with children online. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1-16) ACM.

Lee, C. K., & Shea, M. (2020). Exploring the use of virtual reality by pre-service elementary teachers for teaching science in the elementary classroom. *Journal of Research on Technology in Education, 52*(2), 163-177.

Li, C. T., Hou, H. T., Li, M. C., & Kuo, C. C. (2021). Comparison of mini-game-based flipped classroom and video-based flipped classroom: An analysis of learning performance, flow and concentration on discussion. *The Asia-Pacific Education Researcher*, 1-12.

Lin, Y., Wang, G., & Suh, A. (2020 Exploring the effects of immersive virtual reality on learning outcomes: A two-path model. *International Conference on Human-Computer Interaction* (pp. 86-105). Springer, Cham.

Linn, M. (1985). The cognitive consequences of programming instruction in classrooms. *Educational Researcher, 14*(5), 14-29.

Liu, D., Huang, R., & Wosinski, M. (2017). Contexts of smart learning environments. *Smart Learning in Smart Cities* (pp. 15-29). Singapore: Springer.

Liu, G., & Hwang, G. (2010). A key step to understanding paradigm shifts in e-learning: Towards context-aware ubiquitous learning. *British Journal of Educational Technology, 41*(2), E1 – E9.

Lwoga, E. (2012). Making learning and Web 2.0 technologies work for higher learning institutions in Africa. *Campus-Wide Information Systems, 29*(2), 90-107.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? [Review]. *Computers in Human Behavior, 41*, 51-61. https://doi.org/10.1016/j.chb.2014.09.012

Lyon, J. A., & Magana, J. A. (2020). Computational thinking in higher education: A review of the literature. *Computer Applications in Engineering Education, 28*(5), 1174-1189.

Maphosa, V., Dube, B., & Jita, T. (2021). Sustainable information and learning access at a rural university in Zimbabwe through a mobile application. *International Journal of Information and Education Technology, 11*(2), 82-87.

Mejabi, O. V., Sikiru, I. A., & Agbeti, O. D. (2017). Gap analysis of the service quality of a university ICT Center in Nigeria. *Pacific Journal of Science and Technology, 18*(2), 163-186.

Meinel, C., & Leifer, L. (2012). Design thinking research. In *Design thinking research* (pp. 1-11). Springer, Berlin, Heidelberg.

Merchant, Z., Goetz, E. T., Cifuentes, L., Keeney-Kennicutt, W., & Davis, T. J. (2014). Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis. *Computers & Education*, 70, 29-40. https://www.sciencedirect.com/science/article/pii/S0360131513002108

Mohaghegh, M., & McCauley, M. (2016). Computational thinking: The skill set of the 21st century. *International Journal of Computer Science and Information Technologies, 7*(3), 1524-1530.

Molina-Carmona, R., Villagrá Arnedo, C. J., Gallego-Durán, F. J., Llorens-Largo, F., Meca, I., Pérez Escoda, A., Serrano-Iglesias, S., Sobrino, E., Rabasa, A., & Ruiz Calleja, A. (2020). Research topics on smart learning. In F. J. García-Peñalvo (Ed.), *TEEM'20 Proceedings of the Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*. ACM.

Montero, C. S., & Kapinga, A. F. (2019). Design science research strengthened: Integrating co-creation and co-design. *International Conference on Social Implications of Computers in Developing Countries* (pp. 486-495). Springer, Cham.

Motschnig, R., Pfeiffer, D., Gawin, A., Gawin, P., Steiner, M., & Streli, L. (2018). Enhancing stanford design thinking for kids with digital technologies a participatory action research approach to challenge-based learning. *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.

Mufeti, T. K., & Sverdlik, W. (2017). Introducing computer programming in secondary schools: A case study of NAMTOSS. *IST-Africa Week Conference (IST-Africa)* (pp. 1-8). IEEE.

Naidoo, R. G. (2012). An exploratory survey of design science research amongst South African computing scholars. *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (pp. 335-342). South Africa: ACM.

Nigerian Communications Commission (NCC) (Media Team). (2019). Press Release: With 13.8% contribution to GDP, ICT may double Oil and Gas in 2 years, says Communications Minister. Retrieved April 17, 2021, from https://www.ncc.gov.ng/media-centre/news-headlines/665press-release-with-13-8-contribution-to-gdp-ict-maydouble-oil-and-gas-in-2years-says-communications-minister.

Nigerian Educational Research and Development Council (NERDC). (2014). Nigeria national policy on education. Lagos, Nigeria: NERDC press.

Nguyen, V., Zhang, Y., Jung, K., Xing, W., & Dang, T. (2020). VRASP: A virtual reality environment for learning answer set programming. *International Symposium on Practical Aspects of Declarative Languages* (pp. 82–91). Berlin/Heidelberg, Germany: Springer.

Nwankwo, W. (2018). Promoting equitable access to university education through online learning systems. *World Journal of Engineering Research and Technology, WJERT, 4*(2), 517-543.

Nwankwo, W., & Njoku, C. (2020). Sustainable development in developing societies: The place of ICT-driven computer education. *International Journal of Emerging Technologies in Learning (iJET), 15*(12), 290-297.

Obienu, A. C., & Amadin, F. I. (2018). University education vs students' productivity in computing: The environment as judge. *The Pacific Journal of Science and Technology, 9*(1), 206-213.

Odediran, S. J., Gbadegesin, J. T., & Babalola, M. O. (2015). Facilities management practices in the Nigerian public universities. *Journal of Facilities Management, 13*(1), 5-26.

Offermann, P., Levina, O., Schönherr, M., & Bub, U. (2009). Outline of a design science research process. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology* (pp. 1-11) ACM.

Ogugua, E. E. (2016). Developing productive computer science curriculum for Nigeria. *Journal of Teacher Perspective, 11*(1), 240-245.

Okoroma, N. S. (2008). Admission policies and the quality of university education in Nigeria. *Educational Research Quarterly, 31*(3), 2-24.

Oladipo, F. O., Ibrahim, M. A., Obansa, A. A., & Jatto, A. A. (2017). FULangs: A capstone scripting tool. *International Journal of Engineering and Applied Sciences, 4*(9), 44-50.

Oribhabor, C. B. (2020). Investigating the influence of computer anxiety on the academic performance of junior secondary school students in computer studies in Nigeria. *International Journal of Computing Sciences Research, 4*(4), 370-382.

Oriogun, P., & Ogunleye-Johnson, B. (2012). Computing education: A discussion paper on teaching and assessing ethics. In *2012 African Conference for Sofware Engineering and Applied Computing* (pp. 47-54). IEEE.

Ortega, F., Bolivar, S., Bernal, J., Galvan, A., Tarre, K., Rishe, N., & Barreto, A. (2017). Towards a 3D virtual programming language to increase the number of women in computer science education. *Proceedings of the 2017 IEEE Virtual Reality Workshop on K-12 Embodied Learning through Virtual Augmented Reality (KELVAR*) (pp. 1–6). Los Angeles, CA, USA: IEEE.

Oyelere, S. S., Berghem, S. M., Brännström, R., Rutberg, S., Laine, T. H., & Lindqvist, A. K. (2022). Initial Design and Testing of Multiplayer Cooperative Game to Support Physical Activity in Schools. *Education Sciences, 12*(2), 100.

Oyelere, S. (2017). *Design and development of a mobile learning system for computer science education in Nigerian higher education context*. (Doctoral dissertation, University of Eastern Finland). https://erepo.uef.fi/handle/123456789/18930

Oyelere, S. S., & Suhonen, J. (2016). Design and implementation of MobileEdu m-learning application for computing education in Nigeria: A design research approach. *2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE*) (pp. 27-31). IEEE.

Oyelere, S. S., Suhonen, J., & Sutinen, E. (2016). M-learning: A new paradigm of learning ICT in Nigeria. *International Journal of Interactive Mobile Technologies, 10*(1), 35-44.

Oyelere, S. S., Suhonen, J., & Laine, T. H. (2017). Integrating parson's programming puzzles into a game-based mobile learning application. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research* (pp. 158-162).

Oyelere, S. S., Suhonen, J., Wajiga, G. M., & Sutinen, E. (2018). Design, development, and evaluation of a mobile learning application for computing education. *Education and Information Technologies, 23*(1), 467-495.

Oyelere, S. S., Tomczyk, L., Bouali, N., & Agbo, F. J. (2019a). Blockchain technology and gamification-conditions and opportunities for education. *Adult Education 2018-Transformation in the Era of Digitization and Artificial Intelligence.* (pp. 85-96)

Oyelere, S. S., Agbo, F. J., Sanusi, I. T., Yunusa, A. A., & Sunday, K. (2019b). Impact of puzzle-based learning technique for programming education in Nigeria context. In *2019 IEEE 19th International Conference on Advanced Learning Technologies* (ICALT) (pp. 239-241). IEEE.

Pack, A., Barrett, A., Liang, H. N., & Monteiro, D. V. (2020). University EAP students' perceptions of using a prototype virtual reality learning environment to learn writing structure. *International Journal of Computer-Assisted Language Learning and Teaching (IJCALLT),* 10(1), 27-46.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. NY: Basic Books.

Parmar, D., Isaac, J., Babu, S., D'Souza, N., Leonard, A., Jörg, S., Gundersen, K., & Daily, S. (2016). Programming moves: Design and evaluation of applying embodied interaction in virtual environments to enhance computational thinking in middle school students. *2016 IEEE Virtual Reality (VR)* (pp. 131-140). IEEE.

Parong, J., & Mayer, R. E. (2020). Cognitive consequences of playing brain-training games in immersive virtual reality. *Applied Cognitive Psychology, 31*(1), 29–36.

Patriot, S. (2019). Management of public examinations in secondary schools for national security in Nigeria. *Niger Delta Journal of Education, 11*(2), 171-180.

Paulmani, G. (2020). Improving research skills with Lean-Agile-based instruction: Lean-Agile led, analytics oriented platform for flexible, self-regulated, inclusive learning. *IEEE 20th International Conference on Advanced Learning Technologies (ICALT)* (pp. 387-389). Estonia: IEEE.

Peel, A., Dabholkar, S., Anton, G., Wu, S., Wilensky, U., & Horn, M. (2020). A case study of teacher professional growth through co-design and implementation of computationally enriched biology units. In M. Gresalfi, & I. S. Horn (Eds.), *The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS)* (pp. 1950-1957). Tennessee: International Society of the Learning Sciences.

Pellas, N., & Vosinakis, S. (2018). Learning to think and practice computationally via a 3D simulation game. *Interactive Mobile Communication, Technologies and Learning (*pp. 550–562). Springer: Springer.

Pereira, A. R. (2017). Shaping an integrative front end of innovation (FEI) in a design science approach. Porto: Publication of the University of Porto.

Pfeiffer, A., Lukarov, V., Romagnoli, G., Uckelmann, D., & Schroeder, U. (2020). Experiential learning in labs and multimodal learning analytics. In *Adoption of Data Analytics in Higher Education Learning and Teaching* (pp. 349-373). Springer, Cham.

Piirainen, K. A., & Briggs, R. O. (2011). Design theory in practice–making design science research more transparent. In *International Conference on Design Science Research in Information Systems* (pp. 47-61). Springer, Berlin, Heidelberg.

Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of game-based learning. *Educational Psychologist, 50(*4), 258-283.

Purao, S. (2021). Design science research problems…Where do they come from? In *International Conference on Design Science Research in Information Systems and Technology* (pp. 99-111). Springer, Cham.

Quevedo-Torrero, J. U. (2009). Learning theories in computer science education. *Sixth International Conference on Information Technology: New Generations* (pp. 1634-1635). IEEE.

Radianti, J., Majchrzak, T. A., Fromm, J., & Wohlgenannt, I. (2020). A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education, 147*(103778), 1-29.

Ratnadewi, R., Sartika, E. M., Rahim, R., Anwar, B., Syahril, M., & Winata, H. (2018). Crossing rivers problem solution with breadth-first search approach. *IOP Conference Series: Materials Science and Engineering*. Indonesia: IOP Publishing.

Rich, K. M., Yadav, A., & Larimore, R. A. (2020). Teacher implementation profiles for integrating computational thinking into elementary mathematics and science instruction. *Education and Information Technologies, 25(*4), 3161-3188.

Rojas-López, A., & García-Peñalvo, F. J. (2018). Learning scenarios for the subject methodology of programming from evaluating the computational thinking of new students. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje (IEEE RITA), 13*(1), 30-36. https://doi.org/10.1109/RITA.2018.2809941

Rojas-López, A., Rincón-Flores, E. G., Mena, J. J., García-Peñalvo, F. J., & Ramírez-Montoya, M. S. (2019). Engagement in the course of programming in higher education through the use of gamification. *Universal Access in the Information Society, 18*(3), 583-597. https://doi.org/10.1007/s10209-019-00680-z

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior, 72*, 678-691. https://doi.org/http://dx.doi.org/10.1016/j.chb.2016.08.047

Ronghuai, H., Jing, D., Ting-wen, C., Michael, S., Yan, Z., & Aofan, L. (2017). A conceptual framework for a smart learning engine. Springer Science+Business Media Singapore.

Rossi, M., Henfridsson, O., Lyytinen, K., & Siau, K. (2013). Design science research: The road traveled and the road that lies ahead. *Journal of Database Management (JDM), 24*(3), 1-18.

Sanusi, I. T., Oyelere, S. S., Agbo, F. J., & Suhonen, J. (2021, October). Survey of resources for introducing machine learning in K-12 context. In *2021 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.

Saqr, M., Ng, K., Oyelere, S. S., & Tedre, M. (2021). People, ideas, milestones: A scientometric study of computational thinking. *ACM Transactions on Computing Education (TOCE), 21*(3), 1-17.

Savage, J. M. (2015). Usability assessment of a suicide intervention-prevention mini-game. *In International Conference on Human-Computer Interaction* (pp. 703-708). Springer, Cham.

Segura, R. J., del Pino, F. J., Ogáyar, C. J., & Rueda, A. J. (2020). VR-OCKS: A virtual reality game for learning the basic concepts of programming*. Computer Applications in Engineering Education, 28*(1), 31-41.

Seidel, A., Weidle, F., Börner, C., Flagmeier, L., & Vossler, J. (2019). Learn&play: Co-designing a game-based learning scenario for engineering mechanics. *Academic Conferences International Limited*, (pp. 1033-1037).

Seo, J. H., Malone, E., Beams, B., & Pine, M. (2021). Toward constructivist approach using virtual reality in anatomy education. In *Digital Anatomy* (pp. 343-366). Springer, Cham.

Seo, S., & Kim, E. (2016). A study of collaborative problem-based learning model based on smart education. *International Information Institute (Tokyo). Information, 19*(4), 1095-1100.

Sharma, S., & Ossuetta, E. (2017). Virtual reality instructional modules in education based on gaming metaphor. *Electronic Imaging*, *8*(1), 11–18.

Shonola, S. A., Joy, M. S., Oyelere, S. S., & Suhonen, J. (2016). The impact of mobile devices for learning in higher education institutions: Nigerian universities case study. *International Journal of Modern Education and Computer Science, 8(*8), 43.

Simonsen, J., & Robertson, T. (2012). Routledge international handbook of participatory design. Routledge.

Sooncharoen, S., Thepphakorn, T., & Pongcharoen, P. (2020). A deep learning tool using teaching learning-based optimization for supporting smart learning environment. In *International Conference on Blended Learning* (pp. 392-404). Springer, Cham.

Spector, J. M. (2014). Conceptualizing the emerging field of smart learning environments. *Smart Learning Environments, a SpringerOpen Journal, 1*(2), 1-10.

Spector, J. M. (2016). Smart learning environments: Concepts and issues. *Society for Information Technology & Teacher Education International Conference* (pp. 2728-2737). Waynesville: Association for the Advancement of Computing in Education (AACE).

Spinuzzi, C. (2005). The methodology of participatory design. *Technical Communication, 52*(2), 163-174.

Srimadhaven, T., AV, C., Harshith, N., & Priyaadharshini, M. (2020). Learning analytics: Virtual reality for programming course in higher education. *Procedia Computer Science*, *172*, 433–437.

Srivastava, A. (2016). Enriching student learning experience using augmented reality and smart learning objects. *Proceedings of the 18th ACM International Conference on Multimodal Interaction* (pp. 572-576). ACM.

Stigall, J., & Sharma, S. (2018). Usability and learning effectiveness of game-themed instructional module for teaching stacks and queues. *SoutheastCon* (pp. 1-6). USA: IEEE.

Sunday, K., Ocheja, P., Hussain, S., Oyelere, S., Samson, B., & Agbo, F. (2020). Analyzing student performance in programming education using classification techniques. *International Journal of Emerging Technologies in Learning (iJET), 15*(2), 127-144.

Taisiya, K., Ji, Y. C., & Bong, G. L. (2013). Evolution to smart learning in public education: A case study of Korean public education. *IFIP International Federation for Information Processing, 295*, 170–178.

Tedre, M., & Denning, P. J. (2016). The long quest for computational thinking. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (pp. 120-129). Koli: ACM.

Temdee, P. (2019). Smart learning environment for enhancing digital literacy of Thai youth: A case study of ethnic minority group. *Wireless Personal Communications,* 1-12. https://link.springer.com/content/pdf/10.1007/s11277-019-06637-y.pdf

Tommelein, I. D. (2020). Design science research in construction management: multi-disciplinary collaboration on the SightPlan system. *Construction Management and Economics, 38*(4), 340-354.
Tsalapatas, H., Heidmann, O., Alimisi, R., & Houstis, E. (2012). Game-based programming towards developing algorithmic thinking skills in primary education. *Scientific Bulletin of the Petru Maior University of Targu Mures, 9(*1), 56-63.
Tsortanidou, X., Daradoumis, T., & Barberá, E. (2019). Connecting moments of creativity, computational thinking, collaboration and new media literacy skills. *Information and Learning Science, 120*(11), 704-722.
Vaajakallio, K., & Mattelmäki, T. (2014). Design games in codesign: as a tool, a mindset and a structure. *CoDesign, 10*(1), 63-77.
Vakaloudi, A. D. (2020). Concepts of propaganda: Educating responsible citizens by integrating multiple intelligences and learning styles into a smart learning environment. *In Examining Multiple Intelligences and Digital Technologies for Enhanced Learning Opportunities* (pp. 184-214). IGI Global.
Valtonen, T., López-Pernas, S., Saqr, M., Vartiainen, H., Sointu, E. T., & Tedre, M. (2022). The nature and building blocks of educational technology research. *Computers in Human Behavior*, 128, 107123. https://www.sciencedirect.com/science/article/pii/S0747563221004465
Valle-Tourangeau, F., Guthrie, L., & Villejoubert, G. (2013). Moves in the world are faster than moves in the head: interactivity in the river crossing problem. *Proceedings of the Annual Meeting of the Cognitive Science Society*, (pp. 1504-1509).
Van Borkulo, S., Van Den Heuvel-Panhuizen, M., Bakker, M., & Loomans, H. (2011). One mini-game is not like the other: Different opportunities to learn multiplication tables. In *Joint Conference on Serious Games* (pp. 61-64). Springer, Berlin, Heidelberg.
Van der Merwe, A., Gerber, A., & Smuts, H. (2019). Guidelines for conducting design science research in information systems. In *Annual Conference of the Southern African Computer Lecturers' Association* (pp. 163-178). Springer, Cham.
Vanderdonckt, J., & Vatavu, R. (2020). A pen user interface for controlling a virtual puppet. *Proceedings of the 12th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (pp. 1-6). Sophia Antipolis, France: ACM.

Vartiainen, H., Tedre, M., Kahila, J., & Valtonen, T. (2020). Tensions and trade-offs of participatory learning in the age of machine learning. *Educational Media International, 57(*4), 285-298.

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: A framework for evaluation in design science research. *European Journal of Information Systems, 25*(1), 77-89.

Venable, J., Pries-Heje, J., & Baskerville, R. (2012). A comprehensive framework for evaluation in design science research. In *International Conference on Design Science Research in Information Systems* (pp. 423-438). Springer, Berlin, Heidelberg.

Vilner, T., Zur, E., & Gal-Ezer, J. (2008). Recursive thinking in CS1. In *Proceedings of the ACM-IFIP,* (pp.189-197). Italy: ACM.

Vincur, J., Konopka, M., Tvarozek, J., Hoang, M., & Navrat, P. (2017). Cubely: Virtual reality block-based programming environment. *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, (pp. 1-2).

vom Brocke, J., & Maedche, A. (2019). The DSR grid: Six core dimensions for effectively planning and communicating design science research projects. *Electronic Markets, 29*(3), 379-385.

Walsh, G., Druin, A., Guha, M. L., Bonsignore, E., Foss, E., Yip, J. C., Golub, E., Clegg, T., Brown, Q., Brewer, R. & Joshi, A. (2012). DisCo: A co-design online tool for asynchronous distributed child and adult design partners. In *Proceedings of the 11th International Conference on Interaction Design and Children* (pp. 11-19).

Weber, S. (2010). Design science research: Paradigm or approach? *Proceedings of the Sixteenth Americas Conference on Information Systems* (pp. 1-8). Peru: AMCIS.

Wiegers, K., & Beatty, J. (2013). Software requirements. Pearson Education.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725.

Wu, B., Yu, X., & Gu, X. (2020). Effectiveness of immersive virtual reality using head-mounted displays on learning performance: A meta-analysis. *British Journal of Educational Technology, 51*(6), 1991-2005.

Wu, S., Peel, A., Bain, C., Anton, G., Horn, M., & Wilensky, U. (2020). Workshops and co-design can help teachers integrate computational thinking into their K-12 STEM classes. *Proceedings of International Conference on Computational Thinking Education* (pp. 63-68). Hong Kong: The Education University of Hong Kong.

Yakar, U., Ayfer, S. Ü. L. Ü., Porgali, M., & Çaliş, N. (2020). From constructivist educational technology to mobile constructivism: How mobile learning serves constructivism? *International Journal of Academic Research in Education, 6*(1), 56-75.

Yardley, S., Teunissen, P. W., & Dornan, T. (2012). Experiential learning: Transforming theory into practice. *Medical Teacher, 34*(2), 161-164.

Yassine, S., Kadry, S., & Sicilia, MA (2016). Measuring learning outcomes effectively in smart learning environments. In *2016 Smart Solutions for Future Cities* (pp. 1-5). IEEE.

Yeonjeong, P. (2011). A pedagogical framework for mobile learning: Categorizing educational applications of mobile technologies into four types. *The International Review of Research in Open and Distributed Learning, 12*(2), 1-13.

Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. *2020 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1905-1914). Portugal: IEEE.

Zhang, T., Booth, R., Jean-Louis, R. C., Yeung, A., Gratzer, D., & Strudwick, G. (2020). A primer on usability assessment approaches for health-related applications of virtual reality. *JMIR Serious Games, 8*(4), 1-10.

Zhu, C. (2008). E-learning, constructivism, and knowledge building. *Educational Technology, 48(*6), 29-31.

Zhu, Z. T., Yu, M. H., & Riezebos, P. (2016). A research framework of smart education. *Smart Learning Environments, 3*(1), 1-17.

Appendices

APPENDIX 1. TOOLS USED FOR DATA COLLECTION DURING PROBLEM EXPLICATION

Questions eliciting the challenges, awareness, and use of smart learning environment

Interview Questions

Q1. Are you aware of smart learning environment? And have you used an online learning platform?

Q2. What challenges do you face in your university education that affects your learning generally?

Q3. Tell us how you feel about learning from anywhere at any time. Does this make you feel lazy or otherwise?

Q4. What motivates you to learn computer science courses better?

Q5. Do you evaluate your level of knowledge after completing each course to measure your performance? If yes, tell us how you do this.

Q6. What kind of challenges do you face when learning programming?

Q7. What kind of system and methods do you think can make learning of programming interesting, motivating, and easy?

Questions on user's awareness and use of smart learning environment

Kindly rate these questions on your knowledge and experience of smart learning environment

S/N	Items	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The smart learning solution I have used to learn computer programming has intelligent tutoring feature					
2	The smart learning solution I have used to learn computer					

	programming has context-aware feature					
3	The smart learning solution I have used to learn computer programming can adapt to learner's style					
4	The smart learning solution I have used to learn computer programming has personalized learning feature					
5	The smart learning solution I have used to learn computer programming has automatic feedback mechanism feature					
6	The smart learning solution I have used to learn computer programming can adjust its content based on learner's profile					
7	The smart learning solution I have used to learn computer programming has features to engage and motivate continuous learning					
8	The smart learning solution I have used to learn computer programming has learning analytic feature, thereby supporting me and teacher to make important decisions about my progress					
9	The smart learning solution I have used to learn computer programming has all the components mentioned in section C1 to C8					

What component feature of smart learning environment do you consider to be more important?

……
……………………………

Questions on Expectation Regarding Learning Adaptivity/Style:

Kindly rate these questions on your expectation regarding learning style

S/N	Items	Strongly Disagree	Disagree	No Opinion	Agree	Strongly Agree
1	I prefer to study programming in a group					
2	I learn programming with practical example and tasks than theoretical ones					
3	I expect smart learning environment that support and adapt to my learning style					

Questions on Expectation Regarding Ubiquity and Context-aware Feature of SLE

Kindly tick the box as it applies to you

S/N	Items	Strongly Disagree	Disagree	No Opinion	Agree	Strongly Agree
1	I expect a learning platform that will allow access anywhere anytime					
2	I like to enable the GPS feature on my smartphone when connected to the internet					
3	I expect the ubiquitous feature of smart learning environment to support my learning outcome					

Questions on Expectation Regarding location-aware and Social-aware Feature of SLE

Kindly tick the box as it applies to you

S/N	Items	Strongly Disagree	Disagree	No Opinion	Agree	Strongly Agree

1	I expect a learning platform that displays the location of learners					
2	I like to connect with learners within the same location through the Bluetooth or Wi-Fi					
3	I expect the social features of smart learning environment to allow for social networking among friend					

Questions on Expectation Regarding Pedagogy

Kindly tick the box as it applies to you

S/N	Items	Strongly Disagree	Disagree	No Opinion	Agree	Strongly Agree
1	During a programming course, I feel fulfilled when I am able to solve a difficult problem					
2	My teachers DO NOT encourage me to try out new ideas - think independently					
3	I expect an achievement badge as a feature of smart learning environment to motivate my learning					

Questions Regarding the Anticipated Challenges of Implementing SLE

Kindly tick the box as it applies to your feeling

S/N	Items	Strongly Disagree	Disagree	No Opinion	Agree	Strongly Agree
1	I feel that cost of subscribing to Internet data would NOT be a challenge for implementing smart learning environment.					
2	I feel that lack of electricity supply would constitute a challenge for the					

	implementation of smart learning environment.					
3	I think that Nigeria DO NOT have the cloud infrastructure to support the implementation of smart learning environment.					

State other challenges you feel would confront the implementation of the smart learning environment in Nigeria:

1. ....

2. ....

3. ....

4. ....

5. ....

Thank you for your time and honest responses

APPENDIX 2. TOOLS USED FOR DATA COLLECTION DURING OCD PROCESS.

Questionnaire items administered prior to OCD activities.

Course of study:

Computer Science	☐	Computer Engineering	☐	Information Science	☐
Library Science	☐	Mathematics	☐	Others	☐

Questions on prior knowledge and experience in participating in seminars

S/N	Items
1	I have participated in an online seminar
2	I have participated in an online co-design process
	Questions on game and game design
3	I am an active player of games
4	I play game to gain more fun
5	I play game to gain new knowledge or learn new things
6	I have frequently participated in game design
7	I am familiar with the elements of game
8	I am aware of how game elements operate
	Questions on expectation from participating in the OCD process
9	I am eager in participating in the seminar because the purpose in the invitation notice was clear
10	I am eager in participating in the co-design process because I like to collaborate and share knowledge
11	I expect to learn new things in the co-design seminar
12	I am hoping that the co-design seminar will help me identify new areas of computer science
13	The seminar will provide me the opportunity to design my own game
	Question on ease/difficulty in understanding programming topics
14	In each of the programming topics, choose how easy or difficult they are based on your experience: 1. Programming Basics 2. Variables 3. If Statement 4. Loops 5. Functions 6. Methods 7. Recursions 8. File and Exception Handling

Interview instrument for participants of online co-design process

1. Tell me about your experience during the workshop.
2. What was your expectation before the workshop, and was this expectation met?
3. What is the objective of the workshop (what exactly did you learn)?
4. Is your knowledge of co-design improved in any way? How?
5. Tell me about your experience in co-designing within the group.
6. Did you contribute in any way? Share with me how you contributed, and lesson learned?
7. What other things have you gained from the co-design process with your colleagues, for example, brainstorming about the ideas on games, designs, and paper prototypes?
8. How has this exercise of the co-design process affected your interest in educational games?
9. How has your knowledge of VR improved?
10. Amongst the game elements, you have outlined from the group task, which is most important to you and why?

APPENDIX 3. TOOLS USED TO COLLECT DATA DURING THE SYSTEM EVALUATION.

A. Pre-test questions

Constructs	Items
Students' basic knowledge on CT	1 Are you familiar with the term Computational Thinking?
	2 Do you have basic understanding of what the term Computational Thinking is?
	3. Have you taken any course/topic on computational thinking?
	4. Have you taken any course on introductory programming or introduction to programming
	5. How familiar are these terms: - Problem solving - algorithm or algorithmic thinking - problem decomposition - problem abstraction - pattern recognition - recursion/recursive thinking

B. Post-test questions

Instrument aspects	Items
Learning interests	1. Computational thinking course is interesting.
	2. Learning more about computational thinking concepts is interesting.
	3. It is interesting to answer those computational thinking questions while learning through games.
	4. I always look forward to taking computational thinking topics and prepare for programming class.
	5. The teacher's instructions on computational thinking concepts have attracted my attention.
	6. Anything concerning computational thinking is always interesting to me.
	7. The computational thinking course is more interesting to me in comparison with other courses.
	8. Other courses do not attract me as much as the computational thinking course.
Learning attitudes	1. The computational thinking course is worth studying.
	2. It is worth learning things about computational thinking.

	3. It is worth learning the computational thinking concepts well.
	4. It is important to learn more about computational thinking concepts, such as algorithmic thinking, recursive thinking, and problem solving.
	5. It is important to know and apply the computational thinking concepts
	6. I will actively search for more information and learn about computational thinking concept.
	7. It is important for everyone to take the computational thinking course.
Cognitive benefits	1. The game approach made the comprehension of computational thinking topic easier
	2. The game approach made the memorization of computational thinking topic easier
	3. The game approach helped me to better apply what was learned
	4. The game approach helped me to better analyze the problems
	5. The game approach helped me to have a better overview of the content learned

C. Computational thinking questions

CT competency test

CT questions		Multiple choice
Which of the following is an example of thinking computationally?	a.	Planning out your route when going to meet a friend
	b.	When going to meet a friend, wandering around until you find them
	c.	When going to meet a friend, asking a parent to plan your route for you
Which of the following is NOT an example of computational thinking?	a.	Letting the bossiest friend decide where you should all go
	b.	Considering the different options carefully before deciding upon the best one
	c.	Discussing with your friends how much time and money you have before choosing from a shortlist of places
What is a complex problem?	a.	A problem that, at first, is not easy to solve
	b.	A problem that, at first, is not easy to understand
	c.	A problem that, at first, is not easy to solve or to understand
Which of the following is NOT a computational thinking technique?	a.	Abstraction
	b.	Algorithms
	c.	Coding
Which computational thinking technique involves breaking a problem down into smaller parts?	a.	Decomposition
	b.	Abstraction
	c.	Algorithms
To create a successful computer program, how many computational thinking techniques are usually required?	a.	a. Two
	b.	b. Four
	c.	c. Three
When is a computer most likely to be used when using computational thinking?	a.	During decomposition
	b.	At the end, when programming a computer
	c.	When writing algorithms
Four adventurers (Alex, Brook, Chris and Dusty) need to cross a river in a small canoe. The canoe can only carry 100kg. Alex weighs 90kg, Brook weighs 80kg, Chris weighs 60kg and Dusty weighs 40 kg, and they have 20kg of supplies. Select a possibility for getting across.	a.	Brook and Dusty can row across, Dusty returns.
	b.	Alex and Chris row across and Chris returns.
	c.	Chris and Dusty row across, Dusty returns.
I have three special four-sided dice. They have one letter on each side. When I roll them together, I get three random letters which I try to rearrange into a word. In my eight goes so far, I have made the words: CAT, SON, POD, RIG, PEG, TAP, DIN, APE. How many letters are on the dice altogether?	a.	10
	b.	12
	c.	16
There are five gears connected in a row, the first one is connected to the second one, the second one is connected to the third one, and so on. If the first gear is rotating clockwise what direction is the fifth gear turning?	a.	Clockwise
	b.	Anti-clockwise
	c.	Oscillatory-wise
You are about to leave for holiday, but you forgot socks! You race back to your room, but the power is off so you can't see sock colours. Never mind, because you remember that in your drawer there are ten pairs of green socks, ten pairs of black socks, and eleven pairs of blue socks, but they are all mixed up. How many of your socks do you need to take before you can be sure to have at least one matching pair?	a.	Take three sucks, you are guaranteed of having a matching pair
	b.	Take four socks, you are guaranteed to have at least one matching pair
	c.	Take no sucks since the room is dark and no guarantee of having a matching pair
You must take two different tablets (A and B) each day. If you forget one, or take more than one of any type, you will get sick. Unfortunately, they look exactly the same, and to make matters worse, with only two days' supply left, you drop all 4 tablets on the floor together! What can you do?	a.	Take half of each tablet today, and the rest half of each tablet tomorrow.
	b.	Guess and take any two of the tablets today, and the remaining two tomorrow
	c.	Do not take any of the tablets and remain sick

APPENDIX 4. SAMPLES OF LETTER FROM THE NIGERIAN UNIVERSITY GRANTING PERMISSION TO CONDUCT RESEARCH WITH STAFF AND STUDENTS

FEDERAL UNIVERSITY LOKOJA

FACULTY OF SCIENCE

VICE-CHANCELLOR:
Professor Angela Freeman Miri
B.A. Hons. Eng. (UNIMAID) M.A; Phd Lit Eng (UNIJOS)

DEAN, FACULTY OF SCIENCE:
Professor Sunday Eric Adewumi
B.Sc (UI), M.Sc. (JOS), PhD. (ATBU), FNCS, MCPN, C.itp

P.M.B. 1154
Lokoja,
Kogi State,
Nigeria.
E-mail: info @fulokoja.edu.ng,
www.fulokoja.edu.ng,
facultyofscience@fulokoja.edu.ng

6th September 2018

Agbo Friday Joseph
Doctoral Student
School of Computing
University of Eastern Finland
P. O. Box 111
Joensuu, FINLAND

Dear Mr. Agbo.

Your request for a research visit to our institution has been received. Your intended research activities falls within our area of interest. It is therefore my pleasure to invite you to undertake a visit to our University for your research activities.

Staff and Students are always ready to cooperate with researchers like you for a wonderful research experience.

Given the mutual intellectual benefits to you, staff and students, I strongly recommend you for any funding that will help you accomplish this activity.

Thank you.

Sunday E. Adewumi, *Ph.D, FNCS, MCPN, C.itp*
Professor of Computer Science
Dean, Faculty of Science
Federal University, Lokoja
Kogi State, Nigeria
www.fulokoja.edu.ng
+2348034041590

FEDERAL UNIVERSITY LOKOJA

Office of the Registrar

VICE-CHANCELLOR: PROFESSOR OLAYEMI DUROTIMI AKINWUMI, Ph.D., AvHF, FHSN, MNAL

REGISTRAR: USMAN SULEMAN OBANSA, B.Sc. (UNIMAID), MPA (UNIMAID), MCIA

PMB 1154
Lokoja
Kogi State

March 17, 2021

Mr. Friday Joseph Agbo
Early Stage Researcher,
School of Computing, University of Eastern Finland,
P.O.Box 111, Joensuu,
FINLAND.

Letter of Invitation

With reference to your letter dated March 11, 2021 requesting for permission to conduct a scientific research at the Federal University Lokoja, for the purpose of your doctoral study, I write to inform you that your request has been approved. Hence, you are invited accordingly.

You are to also note that, this consideration is based on the reason that the University will not undertake any financial obligation during the course of the said research as you earlier affirmed.

Benjamin John Mshelbwala
For: Registrar

ARTICLE I

Agbo F. J., Oyelere, S. S., Suhonen, J., & Tukiainen, M.
"Smart learning environment for computing education: readiness for implementation in Nigeria," *EdMedia+ Innovate Learning*, pp. 1382–1391. Amsterdam, Association for the Advancement of Computing in Education (AACE), 2019. https://www.learntechlib.org/p/210149/

Smart learning environment for computing education: readiness for implementation in Nigeria

Friday Joseph Agbo
Solomon Sunday Oyelere
Jarkko Suhonen
Markku Tukiainen
School of Computing, University of Eastern Finland
Finland
fridaya@uef.fi
solomon.oyelere@uef.fi
jarkko.suhonen@uef.fi
markku.tukiainen@uef.fi

Abstract: This study examined the readiness and prospect of implementing the smart learning environment (SLE) for programming education in the context of Nigeria. An overview of design science research as the methodology for implementing SLE was introduced. Data collection was conducted through a questionnaire, and a descriptive statistic was adopted to analyse the data. The result showed that students placed a high expectation on SLE features such as location and social awareness, pedagogy and adaptivity to learning preference. Besides, the study investigated whether the implementation of SLE in Nigeria is possible. Consequently, issues regarding basic requirements such as the inadequate supply of electricity, low memory and battery capacity of the smartphone, and inadequate cloud infrastructures were identified as potential challenges that may affect the implementation of the SLE in the context. Taken together, these findings do support strong recommendations to stakeholders regarding the use of smart learning technology as mainstream for programming education, and provision of basic infrastructures needed for the implementation of SLE at higher education institutions.
Keywords: Smart learning environment, programming education, infrastructure, Nigeria

INTRODUCTION

The implementation of a smart learning environment (SLE) is capable of improving programming education. The SLE is a new concept in the digital learning space created to make learning more adaptive to context and content, providing instant feedback, and engaging learners anytime and anywhere. The SLE is envisioned to make the developing of programming and problem-solving skills a flexible experience. This new paradigm of learning can also make learning accessible to everyone irrespective of location, learning status, and preference (Abtar & Hassan, 2017; Hwang, 2014). Learning of computer programming remains one of the challenges facing the students and educators (Kazimoglu, et al., 2012; Maleko, et al., 2012; Williams, et al., 2002). Computer programming is the actual transcription of abstraction into a computer understandable language. The stages of computer programming involve writing, testing, debugging and running a set of codes using different programming languages (Renumol, et al., 2009; Ahmed, et al., 2018). These stages can be tasking and complex, especially for novices. Previous researches (Renumol, et al., 2009; Stamouli, et al., 2004; Oyelere, et al., 2017) have shown that often, programming is relatively considered difficult among other science-related courses, and building the skill takes time and commitment (Maleko, et al., 2012).

Computing education have been researched over the years and fundamental issues regarding the teachers, students, resources, and teaching methodology have been topical (Sentance & Csizmadia, 2016; Dasuki & Quaye, 2016). Some of those studies identified the potential challenges that confronts the computing education and provided recommendations or solutions (D'Souza, et al., 2008; Oyelere & Suhonen, 2016). In Nigeria context, the challenges are obvious from the resource's perspective, although not too many researches have studied the specific challenges. However, one that has been reported is the use of traditional approach towards teaching and learning of programming and problem-solving skills (Dasuki & Quaye, 2016; Oyelere, et al., 2017). This method is already archaic and should be overhauled by providing innovative technology approach to improve teaching and learning experiences.

Efforts to ease the task of learning how to develop problem-solving skills and programming education exist. Some authors applied approaches such as visualization, games, puzzles, and computational thinking to motivate students and increase interactions between learners and educators (Kazimoglu, et al., 2012; Oyelere, et al., 2017;

D'Souza, et al., 2008; Stamouli, et al., 2004; Williams, et al., 2002). For instance, Oyelere, et al. (2017), developed a mobile learning system for computing education with puzzles to motivate students' learning experience in Nigeria context. The emergence of smart learning technology such as personalized and self-regulated learning system, adaptive, context and location-awareness leaning tools with timely feedback mechanism, provides ways to learn in an efficient and motivating form (Kim, et al., 2012). Previous research on SLEs has shown the theoretical and conceptual design and how this approach allows for enhanced learning experience (Zhu, et al., 2016). This is evident in the survey conducted by Agbo & Oyelere (2019), where different solutions developed with SLE features to enhance programming education were presented.

This study is explicating the challenges of computing education in Nigeria as part of the goal of the authors toward designing and implementing a smart learning solution for programming education in the context. To achieve this goal, the authors employs the design science research method, which is later introduced in this study. Consequently, this study and other previous research deals with the problem explication phase of the design science research. For instance, a study on the awareness and use of SLE in Nigeria shows that teachers and students have no experience of SLE but are eager to enhance their programming education by embracing it (Agbo, et al., 2018). Therefore, this study discusses the potential of using SLE to make learning and teaching problem-solving and programming education more interesting. SLE personalizes student's learning experience, motivates the student, and provides the teachers with the opportunity to give feedback to develop students' confidence. This study also aims to explore the possibility of implementing SLE as a new paradigm for developing problem-solving skills and enhancing programming education in Nigeria. In addition, the use of the smart technology such as smartphone and other wearable devices in education is an emerging area in the learning ecosystem. One of the ways to achieve the aim of implementing SLE for computing education in Nigeria is to leverage the diffusion of smartphones in the country (Oyelere, et al., 2016). Smartphones are affordable and can be useful within and outside the classroom for enhanced learning experience. With the intended intervention of using SLE for teaching and learning problem-solving and computing education in Nigeria, it is possible to have many learners participating in a course at the same time without converging in a physical location. This kind of learning paradigm reduces the dependence on the limited infrastructure to accommodate large numbers of students at the higher education institutions (HEI) as commonly practiced in traditional classrooms. Thus, the following research questions would be addressed in this study.

RQ1: What are the expectations from students regarding the implementation of the smart learning environment for programming education in Nigeria?

RQ2: What are the potential challenges that may confront the implementation of the smart learning environment for programming education in Nigeria?

LITERATURE REVIEW

The concept of smart learning in computing education

Smart learning has recently been researched to allow learners to be wholly immersed in the learning itself. The concept of smart learning is based on certain features: context awareness, adaptivity, ubiquity, and social awareness. Although this concept is still new, and no unanimity exists on its definition (Abtar & Hassan, 2017), some researchers defined it as the application of technology to make pedagogy seamless, flexible, and efficient (Hwang, 2014; Zhu, et al., 2016; Spector, 2014). According to Spector, SLEs can be referred to as an adaptive technology designed to include innovative features and capabilities that improve understanding and performance. This innovation, as stressed by Spector, includes features that make SLEs adaptive, context-aware, and motivate learning. Similarly, Yassine, et al. (2016) defined such environments as a technology-enhanced learning environment that integrates the intelligent learning systems and context-aware ubiquitous learning. The intelligent feature of smart learning environments includes learning analytics and learner's performance evaluation functionalities. Hwang (2014), defined such environments as the technology-supported learning environments that adapt and provide appropriate support. Examples of learning support features include guidance, feedback, hints, and tools provided in the right places and at the right time based on individual learners' needs. These needs might be determined by analysing student's learning behaviour, performance, and online and real-world contexts in which they are situated. These definitions stress the attributes that relate to technologies and learning scenarios of the SLE. However, it is essential also to consider the input needed for the SLE to determine learner's scenarios. Hence, we define the smart learning environment as an enhanced context-aware ubiquitous learning system that leverages social technologies, sensors, and wireless communication for inputs; determine the characteristic of the learners for a personalized learning experience.

Smart learning is an enhancement in mobile learning; thus, one of its aims is to provide users with a learning environment that is not restricted to a single location (Laine & Joy, 2009). According to Gwak (2010), smart learning is focused on learners and content more than on devices; it is effective, intelligent, and tailored learning based on advanced information technology infrastructure. In other words, the focus of SLE should consist the learners and context also, but not only on the utilization of smart devices. Examples of tools designed for computing education with embedded smart features are shown in Table 1.

Table 1 Example of SLE solutions for computing education

Sources	Tool and technology	Pedagogical features
Liew & Xhakaj, (2015)	RedBlackTree Tutor	Help students to learn algorithm for building data structure
Shamsi & Elnagar, (2012)	eGrader	Graph-based grading system for Java programming courses
Grawemeyer, et al., (2017)	LIBE VLE	Personalisation and adaptation features for enhanced learning
Go´mez, et al., (2013)	Units of Learning mobile Player	Adaptation to learners' need—feedback and support, navigation to location, context-aware of educational scenarios
Tillmann & Halleux, (2011)	TouchDevelop	Task-based learning with a little guide in Java or C++
Tillmann, et al., (2013)	Pex4Fun	Game-based, fun-filled learning, and teaching of programming in computer science programming and software engineering class
Oyelere, (2017)	MobileEdu	Puzzle-based learning environment for programming courses
Martin, et al., (2013)	iPro	Game-based tool to learn basic programming
Renny, et al., (2017)	Minerva	Programming educational game that adapts learning content & gameplay
Figueiredo & García-Peñalvo, (2017)	Lightbot	Learning of fundamental programming concepts using Puzzles-based learning platform (JavaScript and Python)
Fabic, et al., (2018)	PyKinetic	A Python tutor mobile tool

Mapping the study context

a. ***Challenges of computing education in Nigeria HEIs***

In Nigeria, students and teachers at HEIs are faced with specific problems that have limited computing education over years (Dasuki & Quaye, 2016; Oyelere, et al., 2016; Agbo, et al., 2018). Similarly, a study by Kamba (2009) had reported that Nigerian universities, for instance, are aware of the technology-enhanced learning environment but are not willing to invest resources for developing such environment to enhance learning. Hence, previous studies' position regarding the state of computing education at HEI in Nigeria remained unclear. Although these issues may not have been studied, but personal experience has shown that teachers find it difficult to teach the subject for comprehensive understanding, and students have also been finding it difficult to comprehend the topics. One of the problems is attributed to a large number of students admitted into a computing degree every year. Hence, it is difficult for teachers to provide intensive tutoring and assessment for their students to identify those with special learning need and provide timely support. Besides, limited number of teachers employed for computing education courses have affected the ratio of students to teachers where teachers are overwhelmed, such that it is impracticable to conduct proper coaching. Sometimes, the number of subjects that are allocated to each teacher in a semester is beyond what the hours available for teaching can allow for effecting course design, which is highly required for programming topics. On the part of students, lack of resilience has caused some students to be discouraged and withdraw from completing computing degree. After failing in a programming course, some students do not have the courage to continue. In addition, the majority are not motivated since the method of teaching is not engaging to arouse their interest to continue the learning (Oyelere, et al., 2016). Sometimes, the period allowed for students to practice in the laboratory is usually inadequate. From personal experience, students, especially in public universities hardly can develop problem-solving skills, understand and read codes, practice and debug errors. Regarding the learning resources, the institutions and government try to provide some infrastructure for learning, but they are not enough. Many of the students cannot afford the cost of learning materials and gadgets such as personal computers or laptops in order to engage themselves in practicing programming skills.

Most of the reported global challenges of computing education (Sentance & Csizmadia, 2016), are viewed from the teachers, students, and resources' perspectives. These challenges as highlighted from the different perspectives by researchers (Dasuki & Quaye, 2016; Kamba, 2009; Ibanga, 2016; Ngene, et al., 2018) are predominant in Nigeria context. The teachers' aspect includes: (i) lack of professional topic knowledge to teach the students; (ii) dry (uninteresting) method and approach of teaching; (iii) limited time spent on designing the course and inadequate preparation. On the students' side, the following are common challenges: (i) low cognitive ability; (ii) lack of problem-solving skills; (iii) poor mathematics understanding; (iv) lack of motivation to learn the topic; (v) limited time for a hands-on practical session. From the resource's perspective, physical infrastructures are the major challenge, which includes: (i) inadequately equipped computing laboratories; (ii) insufficient number of computers for many students admitted into a class; overstressing of the limited resources, which causes them to breakdown. Although these challenges exist, the use of an innovative approach, however, to aid programming education, such as the smart learning environment that comprises of features of computational thinking could reduce these challenges by making the learning experience personalized, adaptive, motivating, and contextualized. For instance, the aspect of assessment and feedback by the teachers can be solved by modelling SLE features that allow automatic feedback mechanism. In addition, students can be motivated to learn through the social-aware features of SLE. Besides, these students, however, possess smartphones, which are affordable, portable, and can allow learning to take place anywhere, anytime, thereby turning computing education from a localized classroom learning to ubiquitous learning.

a. *Infrastructure to support smart learning environment in Nigeria*

The availability of information and communication technology (ICT) infrastructure at universities and other HEIs in Nigeria varies from one to another. Some privately own institutions make use of the contemporary ICT infrastructure such as strong Wi-Fi internet connections; well-equipped laboratories with modern smart visual displays; sophisticated computers, smartphones, and tablets; cloud server infrastructure and services customizable for educational use within the campuses. On the other hand, most of the public universities are only able to use moderate infrastructure. This has limited the staff and students at the public HEI in Nigeria where there is more number of student ratio to staff. Considerable improvement has been reported in the aspect of cloud infrastructure (Matthew, 2016; Gital & Zambuk, 2011). Statistically, the study shows that 82% of Nigerian public universities have adopted the use of cloud computing for educational purpose (Matthew, 2016). Some institutions have installed campus area network supported by wireless connectivity at strategic offices for limited users (Gital & Zambuk, 2011). Besides our experience shows that internet connection has improved at the urban areas leaving the rural areas to weak internet connection due to poor services by the internet service providers (ISPs).

Nigeria is yet to deploy the much-needed infrastructure to support state-of-the-art technologies for education. For instance, strong internet required for such technology is lacking in schools. Besides, other generic challenges that have limited the technology-enhanced learning from penetrating HEIs includes high cost of the implementation of ICT laboratories, cost of computers and internet facilities, inadequately qualified teachers, poor incentives and motivation for the educators, government inability to provide the funding needed for the installation, and insufficient electricity supply (Ibanga, 2016). These challenges can inhibit the implementation of SLE in Nigeria if concrete steps are not taken to resolve the fundamental requirements necessary for such implementation. Recently, the online reports of the Nigerian Communications Commission (NCC)—the body that regulates ICT related matters in Nigeria —shows that there are over 169 million mobile telephone subscribers as at November 2018, and over 108 million active internet data subscribers (Commission, 2018) (See Figure 1 & 2).

This statistic indicates that Nigeria has massive mobile phone and internet users with over 60% of its population being active users, which forms a strong base to make mobile learning possible. It is also an enabler for the implementation of SLE HEI in the context. This report is consistent with the result of one of the recent studies regarding the number of students that possess smartphones and have used it to engage learning in Nigeria (Agbo, et al., 2018). The study discovers that most of the students in tertiary institutions in Nigeria possess smartphones compared to the laptop computer.

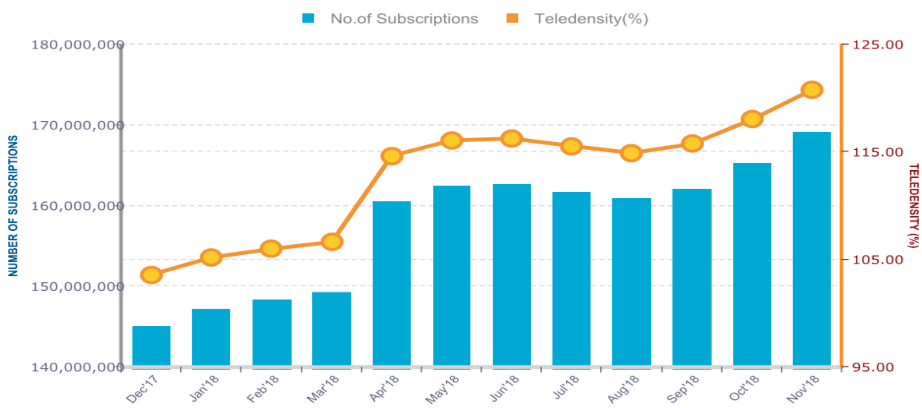


Figure 1 Mobile telephone subscribers in Nigeria as at November 2018: Source (Commission, 2018)

The possible reason for this vast number of smartphone users and mobile subscribers in Nigeria could be due to the affordability of the device, low cost of maintenance, portability and a small amount of power it requires to function. Many of the students regularly subscribe to internet data from their smartphone for online activities such as chatting, messaging, voice and video calling, and accessing learning materials on the internet through social media.

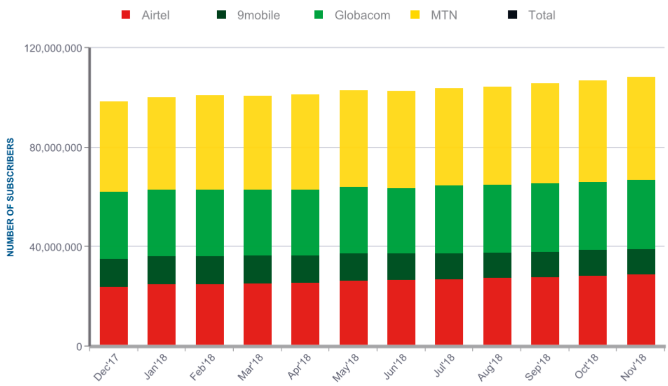


Figure 2 Internet data subscribers in Nigeria as at November 2018: Source (Commission, 2018)

The primary telecommunication firms in Nigeria have internet data bundle subscription plans that are affordable to different categories of users. Even though the internet coverage in Nigeria has not reached the remote rural areas, but the status of coverage is sufficient to encourage the implementation of SLE in Nigeria.

RESEARCH DESIGN AND METHODOLOGY

The methodology of the first author's doctoral research utilizes the design science research (DSR) approach. DSR (Perjons, 2014), comprises of two crucial aspects: developing innovative artifacts and creating practical guidelines for successful decision-making (Oyelere, et al., 2016). DSR creates products and interventions through a rigorous process of iterative refinement of problems, solutions, and concrete scientific methods. One of the unique features of DSR is that it allows the use of feedforward and feedback from the stakeholders to validate the design, development, and evaluation, of research artifacts (Oyelere & Suhonen, 2016). Another benefit of DSR is that it requires researchers and specialists to work in collaboration towards solving problems identified and create innovations (Hevner, et al., 2004). DSR involves the following phases: problem explication, requirements definition, prototype design and development, prototype demonstration, and artifacts evaluation and testing. It iteratively refines problems, solutions, and approaches until the achievement of the desired outcome. The overall aim of our research is to design a novel SLE to support programming education in the context of HEIs in Kogi State, Nigeria. This paper is part of the first phase of the DSR process—problem explication, which investigates the possibilities and readiness for the implementation of SLE in the context in terms of the available infrastructure and stakeholders' perceptions.

In this study, we employ the quantitative analysis of data obtained through a questionnaire. A questionnaire was administered to collect data from students who major in computer science (n=197). Before administering the questionnaire, the first author held a brief session with the students to explain the meaning and concept of SLE as learning aid to support the students learning to program. During the discussions, the participants were informed that choosing to participate in the survey is voluntary. A portion of the questionnaires sought for the consent of the students either to agree to participate or decline. One hundred and sixty-five (83.8%) of the students were from Federal University Lokoja, while thirty-two (16.2%) were from Federal Polytechnic Idah. Moreover, responses were entered according to a five-point Likert-scale as explained by Joshi (Joshi, et al., 2015): Strongly Disagree (SD) to Strongly Agree (SA).

RESULTS

This section is divided into two according to the research questions in the study. The first aspect will present a descriptive analysis of the expectation of students regarding the implementation of the smart learning environment for programming education in Nigeria. While the second part will present findings regarding the readiness of students towards the implementation of the SLE in Nigeria and the presumed challenges it may encounter.

The expectation of students regarding the implementation of a smart learning environment

After the students became aware of the fundamental knowledge of the SLE (including SLE features and potential benefits), the questionnaire that sort for the students' expectations regarding the implementation of SLE to support teaching and learning of programming was distributed to the students. The questions were categorized into two groups as shown in Table 2. The first category includes constructs that are specific to the technical features of SLE— context-aware, location-aware and ubiquity. Similarly, the second category contains constructs that border on the pedagogical components of SLE—adaptivity, pedagogy, and social-awareness. The categorization of these constructs makes it easier to view the outcome of the study concisely from the technical and pedagogical aspects of learning technology.

Regarding context-aware, location-aware, and ubiquity, the students have responded positively to them. For instance, most of the students (M=3.95; SD=1.19) expects a SLE that allows for connecting learners within the same geolocation through the GPS and wireless connection. Even though the students maintain a moderate positive disposition in their responses regarding the ubiquitous operation of SLE, the result shows that the item has the highest standard deviation value in that category (SD=1.48). This, perhaps, shows that the opinion of the students regarding the ubiquity of SLE is scattered across the Likert scale, hence the result falls between positive and neutral.

Table 2 Students' expectations of a smart learning the environment in Nigeria (n=197)

Categories	Constructs	M	SD
Expectation regarding technical features of SLE (adaptivity, context-aware, location-aware. and ubiquity)	I expect a learning platform that will allow access anywhere anytime.	3.73	1.24
	I like to enable the GPS feature on my smartphone when connected to the internet.	4.01	1.19
	I expect a learning platform that displays the location of learners.	4.01	1.07
	I like to connect with learners within the same location through Bluetooth or Wi-Fi.	3.95	1.14
	I expect the ubiquitous feature of the smart learning environment to supports my learning outcome.	3.24	1.48
Expectation regarding pedagogical features of SLE (Pedagogy and social-aware)	I prefer to study programming in a group.	3.66	1.33
	I learn to programme with practical example and tasks than theoretical ones.	3.72	1.31
	I expect a smart learning environment that support and adapt to my ways of learning.	3.93	1.17
	During a programming course, I feel fulfilled when I am able to solve a difficult problem.	3.24	1.51
	My teachers DO NOT encourage me to try out new ideas - think independently.	2.38	1.59
	I expect an achievement badge as a feature of the smart learning environment to motivate my learning.	3.89	1.21
	I expect the social features of the smart learning environment to allow for social networking among friends.	3.70	1.30

Regarding the adaptive feature of the SLE, students have expressed positive to the constructs in this category. For example, the majority of the students (M=3.66; SD=1.33) are expecting a SLE that allows collaborative learning. Collaborative learning is an approach that has been utilized at different educational levels to allow the students to learn from each other (Yassine, et al., 2016; Hayashi, et al., 2015). The result also shows that students preferred to learn with examples (M=3.72; SD=1.31). This shows that the practical session is very relevant during the learning of computer programming. Similarly, the students expect a SLE that can adapt to their preferred ways of learning because 67% of the students (M=3.93) responded positively. This is evident in the students' responses to constructs regarding adaptivity of SLE, which received low standard deviation (SD=1.3), making their responses tilted to the positive side of the scale rather than being scattered. In addition, results in the second category show that students are more interested in features that motivate their learning experience. For instance, most of them (M=3.89; SD=1.21) have expressed that using a SLE with achievement badge will motivate their learning. In another vein, they also expect SLE with features that allow for social networking among learners. However, one of the constructs in this first category that seeks to know whether the students are encouraged to think independently did not present a clear result. This construct (M=2.38; SD=1.59) received the lowest mean and the highest standard deviation in this category. One analysis that is obvious from this result is that the students have a varied opinion on whether there are encouraged to think undependably or not. Again, it can be seen that they are between the negative and neutral. Lack of understanding of the construct by the students may be the reason behind the sparser nature of their responses and hence unclear result.

Anticipated challenges of the implementation of a smart learning environment

The data collected also sort for information regarding the expected challenges that may face the implementation of SLE in the Nigeria context. The descriptive results as shown in Table 3 revealed that the cost of subscribing to the internet could not pose a challenge to the users of SLE in Nigeria. However, the majority of the students (M=3.99; SD=0.97) have strongly expressed that lack or rationing of electricity supply would pose a challenge to the implementation of SLE in the Nigerian context.

Table 3 Anticipated challenges of implementing a smart learning environment

Constructs	M	SD
I feel that the cost of subscribing to internet data would NOT be a challenge for implementing smart learning environment.	3.93	1.22
I feel that the lack of electricity supply would constitute a challenge for the implementation of a smart learning environment.	3.99	0.97
I think that Nigeria DO NOT have the cloud infrastructure to support the implementation of a smart learning environment.	2.76	1.60

On the availability of cloud infrastructure, the result shows that the students seemed to be neutral. Even though the standard deviation value (SD=1.60) being the highest in this list of anticipated challenges, the responses from the students are scattered, with the majority choosing to be neutral. Perhaps, the reason for being neutral is that the students did not understand the detailed meaning of cloud infrastructure or they are not sure of the current infrastructure available in Nigeria. Furthermore, the students in the open-ended part of the questionnaire expressed other perceived challenges. Their opinions form themes that relate mainly to smart devices and government policies. Concerning the devices, they expressed fear about the capacity of its memory, battery strength, and speed. For example, one of the respondents remarked;

"I am concerned about the battery capacity of the smartphones since most of them run low very fast when connected to the internet."

About policies, some of the students nursed the fear of whether SLE would be adopted in the mainstream education for teaching programming at the HEI level. Some respondents further gave reasons for the concern to be based on the manner the government has not reviewed the computer science curricula to introduce the use of 21st century educational technology.

DISCUSSIONS

This study examined the readiness and prospect of implementing SLE for programming education in the context of Nigeria. The study mapped the Nigerian context by presenting the overview of programming education and the teaching methods, exploring the availability of ICT infrastructures, and the potential opportunities for the implementation of SLE for programming education. Next, we carried out a survey where students responded to a list of questions regarding their expectations and presumed potential challenges that the implementation of SLE may face in Nigeria. The participants (n=197) cut across students from two public HEIs in Kogi State, Nigeria. The results from the quantitative survey showed that participants have higher expectations in the technical aspect than the pedagogical aspect. In fact, comparing the results from the technical and pedagogical aspects of the SLE, students' expectation recorded highest positive response in the location awareness using GPS, and allow connecting with friends and other learners within the same geolocation through the wireless technology. The reason for this high level of expectation could be because Nigerian students are very familiar with the social media (Oyelere, et al., 2016), and have used it to connect with friends for messaging or chatting; sharing of files and contents, and many other social network activities, hence would love to embrace such feature in educational solutions. In addition, many of the students were found to respond positively to expectation regarding a SLE that adapt to learning preference. The expectation on adaptive SLE justified that students are more interested in an intelligent, flexible and personalized learning system. The result also revealed that the students' expectation regarding ubiquity and context-awareness features of the SLE is moderate. The authors perceived that since ubiquity and context-awareness are part of the characteristics of smartphone technology, the students expect them to be naturally inherent in the implementation of smart learning systems. On the other hand, a system that allows for a reward such as an achievement badge to motivate learning, received the highest positive response among the items from the pedagogic category. Conversely, while all items in the technical aspect of SLE received a positive response, the result showed that an item in the pedagogical aspect of SLE that discusses the independence of student in solving problem, received a negative response. The authors were wondering whether the students do not understand the construct, or they may not be too sure of their answer. Questions regarding the pedagogic aspect of the SLE may be considered subjective at this stage of the research; hence, students may have varied perception and may not be too quick to place a high expectation on the system they have not utilized.

Regarding the presumed challenges that may face the implementation of SLE, students have expressed that the cost of subscribing to the internet may not pose a challenge to the implementation of SLE for programming education in Nigeria. This outcome is in line with the efforts of the government towards making ICT thrive in Nigeria. For example, the Nigerian Communications Commission (a government institution responsible for regulating the telecommunications industry in Nigeria) has regulated internet providers to ensure that the cost of subscribing to the internet is affordable for users. This result is also in consonant with the statistics of mobile and internet subscribers in Nigeria (Commission, 2018), as the government tries to establish policies that would encourage more users to subscribe to the internet. Besides, competitions among internet providers have also contributed to the low cost of services in order to gain more customers. All of these may have contributed to what influenced the responses of the students regarding internet data subscription. Notwithstanding, the government ought to establish Wi-Fi or internet services free of charge to public universities to enable the students to fully utilize the potential of smart learning. Otherwise, in the long run, students might consider it expensive to purchase internet to study in a smart learning environment. However, the study delineated that inadequate supply of electricity in most part of Nigeria is one of the presumed challenges that the implementation of SLE may encounter. Other challenges identified by this study include lack of sufficient memory and battery capacity of smartphones. Creating a smart learning environment would need electricity to power most of the devices including handheld devices. The availability and supply of this basic commodity is still limited in Nigeria and may constitute a major challenge to the implementation of SLE. The last challenge revealed by the study is concerned with the policy formulation. Students are not sure whether SLE would be adopted as the mainstream education for teaching programming at the university level considering the bureaucratic bottlenecks and limited attention given to the education sector by the government.

CONCLUSIONS

The smart learning environment is envisioned to make learning accessible to everyone irrespective of location, learning status, and preference. The concept of SLE is capable of transforming how to gain problem-solving skills and programming education in this 21st century. In addition, ubiquitous users of smartphones and the penetration of ICT into Africa are a sign of hope and an enabler for the advancement of education by exploring these tools. Moreover, the level of mobile subscribers and ownership of smartphones in Nigeria and familiarity with smartphones by students and teachers at HEIs are good indicators towards the readiness for the implementation of the new learning paradigm. This new paradigm in programming education with SLE leverage the smart devices to make the development of problem-solving skill and the acquisition of programming knowledge interesting. This study explicated the problems faced by teachers and students regarding computing education at HEIs in Nigeria and how features of SLE can help to provide an innovative solution. Although, this study has revealed that Nigeria possesses the basic infrastructural requirements for the implementation of SLE, however, some potential challenges have been identified to be capable of impeding its implementation. Nigeria, and indeed the developing countries can overcome these challenges through collaboration between the stakeholders in the education sector and the government to push for the common goal in favour of smart learning. One of the ways this can be achieved is through rigorous infrastructural development by the government and industries, and an increase in awareness and advocacy about the benefits of smart learning for programming education.

The result of this study indicates that learners placed high expectations regarding the implementation of SLE for programming education in Nigeria and that they are eager to have the experience that would impact their learning. Although the low sample size of participants, data gathering technique, and the research method utilized are obvious limitations of this study, the findings suggest that smart learning approach is the new paradigm for programming education in developing countries. Hence, we recommend that designers of SLE for the educational purpose should take into cognizance the identified expectations of the students to have a user-centered solution; developing countries should leverage the opportunities presented by SLE to deploy state-of-the-art learning technology for enhanced teaching and learning of programming education in this 21st century. Government and the school administration should make further effort to provide basic infrastructures needed for the implementation of SLE at HEIs; for example, providing free Wi-Fi for all government universities and ensuring consistent electricity supply across the country. Our future study would be to design, implement, and evaluate a smart learning system to aid the learning of programming in the context of Nigeria.

REFERENCES

Abtar, D. S. & Hassan, M., (2017). In pursuit of smart learning environments for the 21st century. *In-Progress Reflection No. 12 on Current and Critical Issues in Curriculum, Learning and Assessment*.

Agbo, F.J. & Oyelere, S.S., (2019). Smart mobile learning environment for programming education in Nigeria: adaptivity and context-aware features. *In Proceedings of Computing conference,* Springer.

Agbo, F.J., Oyelere, S.S., Suhonen, J. & Tukiainen, M., (2019). Identifying potential design features of a smart learning environment for programming education in Nigeria, (*in press*).

Ahmed, R. A. M. S., Mahmood, S. M., Nabi, R. M. & Hussein, D. L., (2018). The Impact of Teaching Materials on Learning Computer Programming Languages in Kurdistan Region Universities and Institutes. *Kurdistan Journal of Applied Research (KJAR),* 3(1), pp. 27-33.

Commission, N. C., (2018). Industry statistics and report of the Nigerian Communications Commission. Retrieved from: https://www.ncc.gov.ng/stakeholder/statistics-reports/industry-overview#view-graphs-tables-5.

Dasuki, S. & Quaye, A., (2016). Undergraduate students' failure in programming courses in institutions of higher education in developing countries: a Nigerian perspective. *The Electronic Journal of Information Systems in Developing Countries*, pp. 1-18.

D'Souza, D. et al., (2008). Transforming learning of programming: a mentoring project. *Australasian Computing Education Conference.*

Fabic, G. V., Mitrovic, A. & Neshatian, K., (2018). Adaptive Problem Selection in a Mobile Python Tutor. *In proceeding of User Modeling, Adaptation and Personalization,* ACM*,* pp. 269-274.

Figueiredo, J. & García-Peñalvo, F., (2017). Improving Computational Thinking Using Follow and Give Instructions. *Association for Computing Machinery, ACM,* 978(1), pp. 4503-5386.

Gital, A. Y. & Zambuk, F. U., (2011). Cloud computing: solution to ICT in higher education in Nigeria. *Advances in Applied Science Research,* 2(6), pp. 364-369.

Go´mez, S., Zervas, P., Sampson, D. G. & Fabregat, R., (2013). Context-aware adaptive and personalized mobile learning delivery supported by UoLmP. *Journal of King Saud University –Computer and Information Sciences*, p. 47–61.

Grawemeyer, B., Karoudis, K. & Magoulas, G., (2017). Design and Evaluation of Adaptive Feedback to Foster ICT Information Processing Skills in Young Adults. *In proceeding of International Conference on World Wide Web Companion*, ACM, pp.369-377

Gwak, D., (2010). The meaning and predict of smart learning. *Smart Learning Korea Proceeding, Korean e-Learning Industry*.

Hayashi, Y., Fukamachi, K. & Komatsugawa, H., (2015). Collaborative learning in computer programming courses that adopted the flipped classroom. *In proceeding of International Conference on Learning and Teaching in Computing and Engineering,* IEEE

Hevner, A. R., March, S.T., Park, J. & Ram, S., (2004). Design science in information systems research. MIS Quarterly, pp.75-105.

Hwang, G.-J., (2014). Definition, framework and research issues of smart learning environments - a context-aware ubiquitous learning perspective. *Springer Open Journal*.

Ibanga, I., (2016). Problem and prospect of computer education in Nigeria. Retrieved from https://infoguidenigeria.com/problem-prospect-computer-education-nigeria/

Joshi, A., Kale, S., Chandel, S. & Pal, D. K., (2015). Likert scale: explored and explained. *British Journal of Applied Science & Technology*, pp. 396-403.

Kamba, M. A., (2009). Problems, challenges and benefits of implementing e-learning in Nigerian universities: an empirical study. *International Journal of Emerging Technologies in Learning*, 4(1).

Kazimoglu, C., Kiernan, M., Bacon, L. & Mackinnon, L., (2012). A serious game for developing computational thinking and learning introductory computer programming. *Social and Behavioral Sciences, ScienceDirect-Procedia*, p. 1991 – 1999.

Kim, T., Cho, J. Y. & Lee, B. G., (2012). Evolution to smart learning in public education: a case study of Korean public education. *International Federation for Information Processing (IFIP),* p. 170–178.

Laine, H. T. & Joy, M., (2009). Survey on context-aware pervasive learning environments. *International Journal of Interactive Mobile Technologies,* 3(1), pp. 70-76.

Liew, C. & Xhakaj, F., (2015). Teaching a Complex Process: Insertion in Red Black Trees. In: Switzerland: Springer International Publishing, pp. 698-701.

Maleko, M., Hamilton, M., D'Souza, D., (2012). Novices' perceptions and experiences of a mobile social learning environment for learning of programming. In *Proceeding in 12th International Conference on Innovation and Technology in Computer Science Education*, ACM*,* pp.285-290.

Martin, T., Berland, M., Benton, T. & Smith, C. P., (2013). Learning Programming with IPRO: The Effects of a Mobile, Social Programming Environment. *Internation Journal of Interactive Learning Research,* pp. 301-328.

Matthew, F. T., (2016). Cloud computing in education – A study of trends, challenges and an archetype for effective adoption in Nigerian universities. *University Press of America*.

Ngene, B., Quadri, A., Bamigboye, G. & Tenebe, T., (2018). Nigerian educational system: in the pursuit of right physical environment for learning. In *Proceedings of International Conference on Education and New Learning Technologies Conference*.

Oyelere, S.S. et al., (2019). Impact of puzzle-based learning in computer science education: the case of MobileEdu. *(under review).*

Oyelere, S.S., Paliktzoglou, V. & Suhonen, J., (2016). M-learning in Nigerian higher education: an experimental study with Edmodo. *International Journal of Social Media and Interactive Learning Environments,* 4(1), pp. 43-62.

Oyelere, S.S., Suhonen, J., Wajiga, G. M. & Sutinen, E., (2017). Design, development, and evaluation of a mobile learning application for computing education. *Educational Information Technology,* Springer, p. 467–495.

Oyelere, S.S. & Suhonen, J., (2016). Design and implementation of MobileEdu m-learning application for computing education in Nigeria: A design research approach. *International Conference on Learning and Teaching in Computing and Engineering,* IEEE pp. 48-56.

Oyelere, S.S., Suhonen, J. & Laine, T., (2017). Integrating Parson's programming puzzles into a game-based mobile learning application. *Proceedings of the 17th Koli Calling International Conference on Computing Education Research,* ACM, pp.158-162.

Oyelere, S.S., Suhonen, J. & Sutinen, E., (2016). M-Learning: A new paradigm of learning ICT in Nigeria. *International Journal of Information Management iJIM,* 10(1), pp. 35-44.

Perjons, P. J. E., (2014). A Design Science Primer. An Introduction to Design Science. *Springer International Publishing*.

Renny, S. N., Hasanov, L. A. & Laine, T. H., (2017). *Improving play and learning style adaptation in a programming education game. SCITEPRESS, pp.450-457.*

Renumol, V., Jayaprakash, S. & Janakiram, D., (2009). Classification of cognitive difficulties of students to learn computer programming. *Indian Institute of Technology*, p. 12.

Sentance, S. & Csizmadia, A., 2016. Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technolofies, Springer,* Volume 22, p. 469–495.

Shamsi, F. & Elnagar, A., (2012). ability of grading submission with semantic-errors, effectively, and generating reports for students, as a feedback on their performance, and instructors on the overall performance of the class. *Journal of Intelligent Learning Systems and Applications,* February, pp. 59-69.

Spector, J. M., (2014). Conceptualizing the emerging field of smart learning environments. *Smart Learning Environments, a Springer Open Journal*, 1(2).

Stamouli, I., E., D. & Huggard, M., (2004). Establishing structured support for programming students. *In Proceedings of the 34th American Society of Engineering Education ASEE/IEEE frontiers in Education Conference*, Savannah.

Tillmann, N. & Halleux, J. d., (2011). Pex4Fun: Teaching and LearningComputer Sciencevia Social Gaming. *IEEE* .

Tillmann, N. et al., (2013). Teaching and Learning Programming and Software Engineering via Interactive Gaming. San Francisco.

Williams, L. et al., (2002). In support of pair programming in the introductory computer science course. *Journal of Computer Science Education*, pp. 197-212.

Yassine, S., Kadry, S. & Sicilia, M., (2016). Measuring learning outcomes effectively in smart learning environments. Smart *Solutions for Future Cities* IEEE.

Zhu, Z., Sun, Y. & Riezebos, P., (2016). Introducing the smart education framework: Core elements for successful learning in a digital world. *International Journal of Smart Technology and Learning,* p. 53.

ARTICLE II

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Adewumi, S.
“A systematic review of computational thinking approach for programming education in higher education institutions,” *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, pp. 1–10, 2019. https://dl.acm.org/doi/abs/10.1145/3364510.3364521

ARTICLE III

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Tukiainen, M.
“Identifying potential design features of a smart learning environment for programming education in Nigeria,” *International Journal of Learning Technology, 14*(4), 331–354.
https://www.inderscienceonline.com/doi/abs/10.1504/IJLT.2019.106551

Identifying potential design features of a smart learning environment for programming education in Nigeria

Friday Joseph Agbo*,
Solomon Sunday Oyelere, Jarkko Suhonen and Markku Tukiainen

School of Computing,
Joensuu Campus,
University of Eastern Finland,
P.O. Box 111, FIN-80101 Joensuu, Finland
Email: fridaya@uef.fi
Email: solomon.oyelere@uef.fi
Email: jarkko.suhonen@uef.fi
Email: markku.tukiainen@uef.fi
*Corresponding author

Abstract: Smart learning environment (SLE) has been researched to enhance teaching and learning by providing personalised learning, quick feedback, motivation and learning support. This study discusses the features of SLE that are relevant to programming education and the general design features for developing SLEs. In addition, the study provides insights into the level of awareness and use of the SLE for programming education in the Nigerian higher education institutions (HEIs). In this study, mixed research method was employed to conduct a survey among the teachers and students of computer science at HEI in Nigeria. Data were collected through questionnaire and interview instruments. The study showed that the students and teachers have no experience of SLEs but indicate strong willingness to embrace the use of the SLE for programming education. Besides, tentative features of SLE such as learning guides, personalised learning, quick feedback mechanisms, and automatic task scheduling were identified and presented.

Keywords: smart learning environment; SLE; programming education; design principles; Nigeria context; Nigeria.

Reference to this paper should be made as follows: Agbo, F.J., Oyelere, S.S., Suhonen, J. and Tukiainen, M. (xxxx) 'Identifying potential design features of a smart learning environment for programming education in Nigeria', *Int. J. Learning Technology*, Vol. X, No. Y, pp.xxx–xxx.

Biographical notes: Friday Joseph Agbo is a PhD candidate at School of Computing, University of Eastern Finland. He received his MSc in Computer Science from University of Ilorin, Nigeria in 2017. His research interests include smart learning environment, mobile learning for computer programming, computer science education, and computational thinking. His doctoral research is focused on designing a smart learning environment for programming education. He has experience in web development, and a University Assistant Lecturer.

Solomon Sunday Oyelere is a Postdoctoral Researcher and holds teaching position at the School of Computing, University of Eastern Finland. He received his MSc from Technische Universität Ilmenau, Germany in 2013 and his PhD from University of Eastern Finland in 2018. His research interests include computing education, educational technology, interactive mobile technologies, and learning environments.

Jarkko Suhonen holds a research manager position at the School of Computing, University of Eastern Finland. He received his MSc in Computer Science from University of Joensuu, Finland in 2000 and PhD in Computer Science from University of Joensuu in 2005. He has published over 100 peer-reviewed articles in scientific journals, conferences, workshops, and chapters of books. His research interests include online and blended learning, design science research, computing education, and ICT for development.

Markku Tukiainen is a Professor in School of Computing, University of Eastern Finland and head of IMPDET-LE studies. His research interests are in topics on interactive technologies, human-technology interaction, educational software and systems, or ICT4D. His other topics are in the field of computing education (for example developing tools for learning and teaching basic programming).

1 Introduction

A smart learning environment (SLE) is relevant to programming education, since it supports ubiquitous and personalised learning. The features of SLEs include adapting to learners' preferred ways of learning, context awareness, ubiquity, and intelligent feedback mechanism (Laine and Joy, 2009). Intelligent feedback mechanism can improve learners' programming experience. For example, an intelligent tutoring system, Ask-Elli (Gerdes et al., 2016), helps students learn functional programming to incrementally add to their knowledge and receive feedback on their choice of response, accompanied by a hint when in a dilemma. Besides, the SLE considers learners' affective aspects such as motivation and emotional states. This kind of environment offers great potential for producing professionals who can positively support the social, economic and technological growth of the developing countries.

The failure rate of students in programming courses has been reported to be high (Guzdial and Soloway, 2002). In Nigeria, many graduates acquire the theoretical knowledge of programming, which is related to either teaching methods or teachers' or students' attitudes to programming education (Akinola and Nosiru, 2014). According to Kamba (2009), the Nigerian university teachers and students have great awareness of the technology-enhanced learning environment; however, investments in and commitments to developing such tools are poor and below expectation. Several other challenges facing programming education in Nigeria have been identified (Kamba, 2009), and these include inappropriate teaching methods, poor teacher-student ratio, inadequate computers and laboratories and the inability to afford learning materials. Besides, from the author's experience, computer science students in Nigeria can barely afford a laptop to hone their programming skills; these students, however, own smartphones, which are affordable and have the necessary features to support smart learning.

Table 1 Technology-enhanced learning for teaching/learning programming

Digital learning environments	*Explanation*
CSmart (Gajraj et al., 2011)	CSmart is an integrated development environment that visualises each statement of a C program to the learner while typing in. CSmart has built-in intelligence that already knows exactly what code it requires the learner to type for each assignment.
ViLLE (Laakso et al., 2008)	ViLLE is an online visualisation system that displays numerous program examples, predefined to aid students' learning.
Jype (Helminen, 2009)	Jype is an IDE and a web-based platform for automatic assessment of programming exercises. It is a beginner friendly system that is intended to support programme and algorithm visualisation.
Jaliot (Levy and Ben-Ari, 2007)	Jaliot is a user-friendly graphic user interface (GUI) platform that has animations to enhance visualised learning of computer programming. Since its introduction, several versions of Jaliot have been released with enhancements and new features.
ViRPlay3D2 (Jiménez-Díaz et al., 2011)	ViRPlay3D2 is a platform that provides a virtual 3D object. Students learn by controlling avatars, represented as an object in the execution of an object-oriented program.
UUhistle (Sorva and Sirkiä, 2010)	A visual program simulation that is developed to allow learners debug programs with animation and exploratory examples.

Studies on how to help students learn and teachers teach programming have been conducted in the past. For example, Kordaki (2010) conducted a pilot study on LECGO, a learning environment for programming education. This environment helps beginners of C programming learn the concept of problem-solving through drawing simple geometrical objects. Similarly, other interventions such as IPRO have been offered to enable students to learn programming on their smartphones (Chao et al., 2013; Martin et al., 2013). Despite these studies (see Table 1), the ability to integrate smart features of the mobile technology into programming education to enhance the learning experience has not been given sufficient attention.

In our effort to seek a solution that allows practical programming knowledge in Nigeria, we propose a smart learning approach that provides the opportunity for improving the learning experience. Besides, the study leverages the widespread use of smartphone technology and advocates integrating its features into programming education. The SLE aims to enable flexible, accessible and efficient teaching and learning. To this end, this research investigates the awareness and perception of students and teachers of computer science in Nigerian higher education institutions (HEIs); this study also seeks to understand the extent of students and teachers experience of SLEs for programming education. This investigation intends to gain insights into the identified problems of teaching/learning of programming and create a road map for designing and developing an SLE for programming education in the context of Nigeria.

This study is significant because computer programming expertise has become essential for students in the 21st century, irrespective of the level of education and course major (Fessakis et al., 2013; Verdú et al., 2012). Learning computer programming entails comprehending the essential theoretical and practical aspects, which most learners usually find uninteresting (Yeh et al., 2010). This study presupposes that the SLE is able to transform programming education by providing individual learners content adaptation, tailored feedback, intelligent support and personalised recommendations. Therefore, this study is part of the plan towards implementing the SLE for programming education in Nigeria.

Research questions

This study considered the following research questions (RQs):

RQ1 What is the extent of computer science students' and teachers' awareness of the SLE in HEIs in Kogi State, Nigeria?

RQ2 Has the SLE been used in computing courses at HEI in Kogi State, Nigeria?

RQ3 What are the potential design features of the SLE in the Nigerian HEI context?

The structure of this article is as follows. Section 2 introduces the concept of the SLE and its features relevant to programming education. Section 3 focuses on the research design, context and methodology. Section 4 presents the results of the study regarding the students' and teachers' awareness and use of the SLE in Nigerian HEI. Section 5 discusses the findings and presents the potential features of SLE to guide its modelling for programming education as a reflection of the findings from the survey. Finally, Section 6 presents the concluding remarks regarding the findings of this study and offers recommendations to the stakeholders and future researchers in the SLE.

2 Background

The SLE emerged in research publications in 2012 when Huang et al. (2012b) introduced it as the highest level of digital environment for a learning system. Since then, many authors (e.g., Hwang, 2014; Spector, 2014) have made a tremendous contribution to the concept. The advancement of technology transformed the educational learning environment from one that was associated with a mobile learning environment to one that began to be characterised as a ubiquitous learning environment and today as a SLE (Taisiya et al., 2013). Accordingly, through the building block of learning technology (i.e., the technology-enhanced system), learning is transitioning from web-based learning to wireless mobile-based learning, from mobile-based learning to context-aware ubiquitous learning (Yeonjeong, 2011) and from context-aware-based learning to socially aware learning technology (Liu and Hwang, 2010).

Although previous research has shown that the concept of the SLE is still new and lacks definitional unanimity (Abtar and Hassan, 2017), some researchers (e.g., Abtar and Hassan, 2017; Hwang, 2014; Sahar et al., 2016; Spector, 2014) have tried to conceptualise it as an application of technology that renders pedagogy seamless, flexible and efficient. According to Spector (2014), the SLE can be construed as an adaptive technology, designed to include innovative features to improve understanding and performance. Innovation, as stressed by Spector (2014), includes features that make the SLE adaptive, context-aware and motivating for learners. Similarly, Sahar et al. (2016) define the SLE as a technology-enhanced learning environment that incorporates the criteria and roles of intelligent learning systems and context-aware ubiquitous learning. The intelligent feature of the SLE includes the learning analytics and learners' performance evaluation functionalities. Hwang (2014, p.5) defined the SLE "as the technology-supported learning environments that make adaptations and provide appropriate support." The supporting features include "guidance, feedback, hints, and tools in the right places and at the right time based on individual learners' needs. These needs might be determined via their learning behaviors, performance, and the online and real-world contexts in which they are situated" [Hwang, (2014), p.5]. The SLE is the form of an intelligent, adaptive and personalised learning intervention that can be integrated with a diversity of devices (Huang et al., 2017).

Drawing on the array of definitions from different authors, we define the SLE as an enhanced context-aware ubiquitous learning system that leverages social technologies, sensors and wireless communication of mobile devices to engage learners in hands-on experiences and present contents in a stimulating form; capable of connecting the learning community, increasing awareness of the physical environment, tracking and providing learning support.

2.1 The SLE components

Based on the SLE literature, we identified the key components that play a vital role in the design of SLEs. The components include the user, device, technology, context and pedagogy (see Figure 1); these components helped in outlining the proposed guiding principles of the SLE design for programming education (see Table 7).

Each of the components is connected with the context of the application. For instance, the user can engage in learning from a different context; the device context varies with specific characteristics; the technology can also be discussed from a different context, and the pedagogy depends on the context at every instance of learning. The user directly benefits from engaging in the learning process, thereby expecting a better experience in the end.

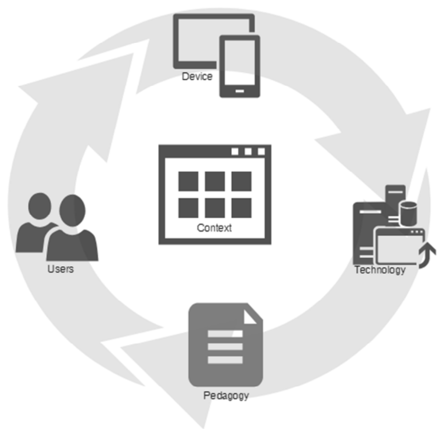
1 *The context* of the learner, device, technology and learning contents plays a vital role in determining the state that the solution should assume at any instance (Yaghmaie and Bahreininejad, 2011).

2 *The types of devices* used to engage learning is important, as not all devices have the prerequisite features to enable smart learning. For example, smartphones, tablets and other wearable devices are useful options (Periera and Rodrigues, 2013), whereas old generation computer systems such as the mainframe, desktop computers and other premised-based computers do not have the required technology to enable smart learning.

3 *Technology* needs to address the design architecture, system communication flow, input and output processes and connection and storage facilities of all the technical aspects of the learning environment (Periera and Rodrigues, 2013; Roussos, 2002).

4 *Pedagogy* is the entire goal of developing an SLE; it includes the anticipated learning theory, strategy, method, outcome and feedback to make the learner aware of the progress made after an instance of learning. Pedagogy, as conceptualised in this study, is connected with the learning theories, since pedagogical principles are basically concerned with the fundamental theories of learning (Ben-Ari, 1998; Jill and Carol, 2004; Quevedo-Torrero, 2009).

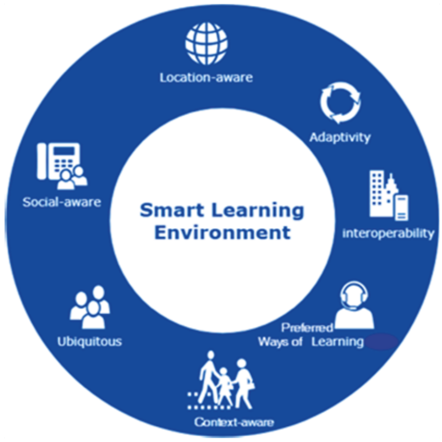
Figure 1 The SLE components



2.2 The SLE features relevant to programming education

The concept of adaptivity, context-awareness, ubiquity, and preferred ways of learning and intelligent system are the critical elements of the smart learning system (Laine and Joy, 2009). These elements are referred to as the features of the SLE (Zhu et al., 2016), and they are particularly relevant when designing the SLE for programming education. Although Zhu et al. (2016) have identified ten features of the SLE, within a broader perspective of discipline, we concentrate on seven critical features of SLEs, which are computer science education specific; the rationale behind this decision is because designing an SLE for programming education requires significant components that can enhance learners' cognitive ability and problem-solving skill. In this section, we discuss the seven features of the SLE, relevant to programming education (see Figure 2).

Figure 2 The SLE features (see online version for colours)



2.2.1 Location-aware

The SLE is expected to be location-aware (Zhu et al., 2016); in other words, an SLE should be aware of all the environments and the situation in a particular environment to allow the user to learn within any location. The locations of a learner at any point in time can impact on his/her learning process and level of understanding. For example, in the context of location-awareness of computer programming study, a task that is interesting to a learner at a play garden may not be attractive to the same learner while on the road; some users may prefer a simple computational arithmetic task, such as addition, subtraction or multiplication while at a shopping mall, whereas others may prefer to work out logical problems when in the classroom or laboratory. A learner's contextual location can be acquired from the environment, whether indoors or outdoors, using global system for mobile (GSM) communication, global positioning system (GPS), a combination of both GSM and GPS methods and radio frequency identification (RFID). Of these methods, RFID is most common because of its low price, independence of deployment and ease of implementation (Roussos, 2002). For example, a location-based and adaptive mobile learning system called multi-object identification augmented reality (MOIAR) made use of RFID and GPS to improve the learning content adaptability of learners (Chang et al., 2010).

2.2.2 Adaptivity

The concept of adaptivity in learning systems has recently been discussed to enhance personalised learning (Graf and Kinshuk, 2008). In the context of the learning environment, adaptivity is a function in an intelligent tutoring system which allows students to learn according to their characteristics, which include current context, needs, situation and scenarios (Graf and Kinshuk, 2008; Laine and Joy, 2009; Huang et al., 2012a). Adaptivity is attainable from different perspectives. For instance, some studies have modelled the learning environment to adapt it to the learning content, device, environment, colour, language, learners' preferred interface or other personal preferences (Akbari and Taghiyareh, 2014; Renny et al., 2017; Yaghmaie and Bahreininejad, 2011). A practical example of an adaptive learning environment is the Java tutorial environment, which is aligned with the learner's current method, navigation and needs (Vesin et al., 2012). When a learner with a particular learning need is unable to attain the required grade for a specific concept, the adaptive learning environment changes to a more preferred learning option. Another example of adaptive learning environment is Problet (Kumar, 2013). Problet allows users to learn problem-solving exercises in introductory C++/Java/C# programming courses. The mechanism for adaptivity in Problet is achieved by generating problems for only those concepts that the student has not yet mastered. This adaptive mechanism minimises the time spent on learning and better captures the interest of the student (Kumar, 2013).

2.2.3 Interoperability

Interoperability ensures that solutions can support various technologies in order to enhance information exchange. In other words, the system should be able to support and operate according to different types of technology (e.g., Android, iOS, Windows, etc.) to reduce the cost of usage (Roussos, 2002). Interoperability is crucial for developing an SLE, since users can be scattered across different locations and device contexts. The fast health interoperability resources (FHIR) prototype (Mandel et al., 2016), for example, was built for interoperability of medical applications; once built, it was able to run unmodified across different healthcare IT systems.

2.2.4 Preferred ways of learning

Learners tend to adopt different learning traits; in other words, the ways learners process learning contents may depend on their preference. For instance, some learners can process audio-visual material better than text. These cognitive capabilities and styles of content processing can affect students' preferred ways of learning. An SLE for programming education context tends to incorporate the features that allow different modes of teaching and content presentation conducive to learners' preferences, with a view to helping the learner understand the programming concept and build his/her skills over time. Minerva, for example, was developed to aid programming education by adapting learning content and gameplay to the learning and play styles of the player (Renny et al., 2017).

2.2.5 Ubiquitous

Ubiquity has been defined as "a new learning paradigm in which we learn about anything at anytime, anywhere utilising ubiquitous computing technology and infrastructure" (Peter et al., 2010). Ubiquitous learning is the kind of learning technology that is usually associated with a considerable number of microelectronic devices (small computers), which are capable of performing the functions of computation and communication; examples of such devices include smartphones, contactless smart cards, handheld terminals, sensor network nodes, RFIDs and many more devices for everyday use (Peter et al., 2010). The ubiquitous component of the learning environment can further detect learners' conditions and contexts, including locations, actions, time and weather (Saadiah et al., 2010). The characteristics of ubiquitous learning, according to Kawahara et al. (2003), are permanence, accessibility, immediacy and interactivity. Graf and Kinshuk (2008) define the ubiquitous learning system as a learning environment that combines mobile and pervasive learning. With the advancement of mobile technologies, new innovative approaches are employed to achieve ubiquitous context-aware learning. For example, mobile learning support system (MLSS) is a ubiquitous learning system, which was designed to enable students to have access to learning materials and benefit from the functions of mobile devices, such as the camera for barcode reading and GPS for location detection (Huang et al., 2010; Yang et al., 2007).

2.2.6 Context-aware

The concept of context-aware learning (Hwang et al., 2008) is aimed at transmitting proper instructional materials and other information to learners, according to their individual needs; this is done through the sensors embedded in the medium of transmission (Hwang et al., 2008; Ching-Bang, 2017). Macredie et al. (2006) conducted a systematic survey and analysis of publications in the field of the context-aware mobile learning system to arrive at a classification framework. In order to investigate the content in the field of a context-aware learning system, the authors classified the framework into layers. The hardware architecture layer consisted of the device used, system infrastructure and the connection type; the context determination layer consisted of the type of content and the type of sensors; evaluation layer consisted of the methods by which the studies and the participants of the studies were evaluated (Macredie et al., 2006). The components of the evaluation layer are the questionnaire, pre-post-test and interview. This classification is relevant to the development of the smart learning system by X-raying the components from the type of devices, context and sensors during the design process. For example, Laine and Joy (2009) surveyed the context-aware and pervasive learning environment and identified personal digital assistants (PDAs) and RFIDs as the most common sensor technologies for context acquisition.

2.2.7 Social awareness

The social awareness features of the SLE mean that the system is aware of its environment, knows what is happening around it and is able to accurately interpret the emotions of users with whom the system interacts (Airth, 2018). Social awareness, according to Airth (2018), requires competency in areas such as empathy and emotional intelligence. Theoretically, social awareness involves the interworking of multiple concepts, including social sensitivity (empathy for others and the ability to infer), social insight (moral judgement and the ability to comprehend situations quickly) and social communication (the ability to interact appropriately with others, including problem-solving interactions) (Airth, 2018). For instance, the *friends, messages, and blogs* features of the system allow social interaction between two or more friends using the platform. These features also enhance social awareness and collaboration among learners, which renders it a useful example of a learning environment for programming education in context.

Other features of the SLE that are relevant to teaching and learning of programming include:

a automatic feedback, a feature that has been implemented in technology enhanced learning (TEL), such as Web-CAT and TRAKLA (Korhonen et al., 2003; Edwards and Perez-Quinones, 2008; Annamaa et al., 2017)

b program visualisation (e.g., Alice, Microbit, Jeliot, ANIMAL and Jsvee visualisation) (Cooper et al., 2000; Roßling and Freisleben, 2002; Moreno et al., 2004; Rogers and Siever, 2018)

c intelligent tutoring and learning support systems such as MicK, Blue and FLEXauth (Barnes and Kolling, 2006; Arends et al., 2017; Opgen-Rhein et al., 2018).

Recent studies have shown that the SLE features can effectively enhance education. For example, Ha and Kim (2014) conducted a literature review of the use of smart tools and social platforms (e.g., Twitter) among higher education students and teachers and reported on the benefits of the SLE.

3 Research design and methodology

3.1 Research context, methods and data analysis

The participants of this study were students and teachers of two public HEIs in Kogi State, Nigeria. In this study, the mixed methods research was used, which combines the qualitative and quantitative research approaches (Schoonenboom and Johnson, 2017). The combined approaches serve to confirm the data obtained for the purpose of the study. Therefore, the interview questions were designed in tandem with the questions in the questionnaire. Data were collected using questionnaires and interviews. This approach provides in-depth insights, which can guide the SLE design for programming education in Nigerian HEIs.

3.2 The questionnaire

The questionnaire instruments were developed and administered to random groups of students and teachers. The authors did not use the existing items due to the nature and context of the study; however, a professor and three senior lecturers of computer science who also participated in the study were consulted to validate the questionnaire and the interview instruments to identify and remove any ambiguous and/or misleading questions. Besides, a small sample of participants was selected randomly to pilot the instruments. To further test for reliability of the scales, Cronbach alpha coefficient ($\alpha = 0.93$) was reported which shows that the scales are reliable. During the distribution of the questionnaire, we sought the participants' consent to allow us to use the data for the research purpose and subsequent publication, although their identities remained anonymous. Since participation was voluntary, there was no compensation of any form. The participants were also notified of their right to withdraw from the study at any time and stage. A total of 210 students and 15 teachers agreed to complete the questionnaires; 180 (i.e., 85%) of the students were from Federal University, Lokoja, while 30 (i.e., 15%) of the students were from Federal Polytechnic, Idah. The items in the questionnaire elicit information on the principles that guide the design of SLE for programming education. The survey questions consisted of Likert scale options (Joshi et al., 2015).

3.3 The interview

The interview was used as the second data collection instrument. The main reason for collecting qualitative interview data was to confirm the quantitative analysis results (Shenton, 2004) and gain further insights such as the students' challenges of learning programming. Six students were selected randomly for the interviews to elicit their opinions on the SLE. Two focused groups were formed. Each of the interview groups consisted of three students. At the time of the study, the students were engaged in a semester's examination and therefore had a tight schedule, which precluded recruiting more students. Moreover, one of the purposes of collecting and analysing qualitative data was for triangulation, which allows confirming the findings (Bekhet and Zauszniewski, 2012) and compensating for the limitations, if any at all, of the quantitative method (Shenton, 2004). In addition, some of the identified themes in the qualitative analysis dealt with the challenges of teaching/learning programming. Therefore, a semi-structured interview with two focused groups was considered for the study. The interview was conducted at different times. The responses from the group interview were recorded using the recorder of an Android smartphone. We adopted the procedures presented by Raymond (1992) to analyse the interview data, which is explained in the following steps:

a After transcription, we read through the entire transcript and noted some of the repeated and noticeable words or phrases. For example, 'I prefer to …', '…use smartphone to learn…', and 'the screen resolution…'. These words or phrases provides insight towards users' expectation of SLE and it can aid the design.

b A more careful reading of the transcripts allowed highlighting and underlining the words and phrases that are either related to the features/components of SLE or connected to programming education. For example, 'screen resolution' is related to the devices component of SLE while learner's 'preference' and 'style' are related the pedagogy component of SLE.

c The highlighted words and phrases were coded (using alphabets such as A, B, C…), and the codes were categorised based on how they were related.

d The categories were grouped together according to the components of the SLE – users, context, devices, technology and pedagogy.

e The five groups of SLE components is focused on this study as presented in Section 2.1 and they formed the thematic basis of the results in Table 7.

4 Results

The analysis of data was performed according to the Likert scale, and the options were coded as follows: strongly agree (SA = 1), agree (A = 2), neutral (N = 3), disagree (D = 4) and strongly disagree (SD = 5). This means that, in our descriptive statistics, the lower the value of the mean (M), the greater the number of responses in favour of the questionnaire constructs; similarly, the higher the mean (M) value, the smaller the number of responses in favour of the questionnaire constructs.

4.1 Awareness of the SLE

The results show that 76.2% (*N* = 210) of the students owned smartphones, while 100% (*N* = 15) of the teachers owned smartphones. Regarding the awareness of the SLE, the results in Table 2 illustrate awareness among a great number of students (M = 1.86). Similarly, the mean score for teachers (M = 1.3) suggests that a large number of them are aware of the SLE. Interestingly, the results show that the teachers have greater awareness of the SLE than the students.

Table 2 Awareness of the SLE

	1. I am aware of the SLE		*2. I knew about the SLE before coming to the university*		*3. I access learning/teaching materials on my mobile phone*	
	M	*SD*	*M*	*SD*	*M*	*SD*
Students (*N* = 210)	1.86	0.94	2.29	0.94	1.52	0.94
Teachers (*N* = 15)	1.30	0.49	3.00	0.85	2.00	1.46

Concerning awareness of smart learning prior to coming to HEI, surprisingly the results show that 57.1% of the students (M = 2.29) knew about the SLE before coming to the university, as opposed to 33.3% of the teachers (M = 3.0). Similarly, more students access learning materials on their mobile devices compared to the teachers (students, M = 1.52 < teachers, M = 2.00).

4.2 Use of the SLE

With respect to the use of the SLE for programming courses, more than half of the students (M = 3.10) indicated that they had not used the SLE programming courses.

Table 3 Use of the SLE

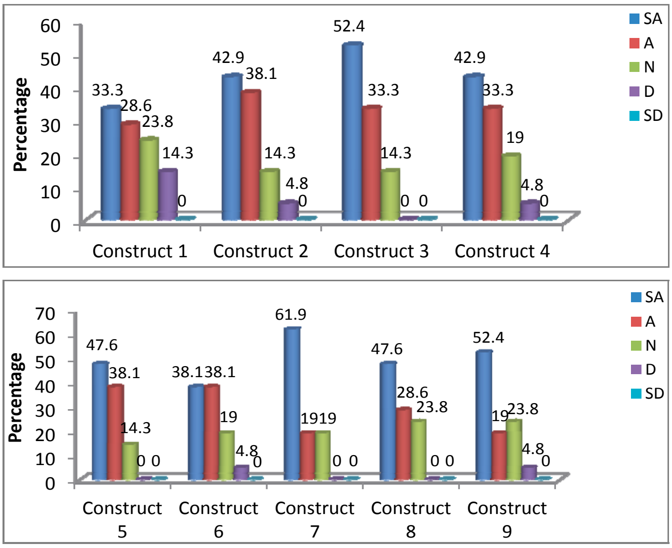
		Constructs (C)	*M*	*SD*
Students (*N* = 210)	C1	I have been taught a programming course using the mobile smart learning solution.	3.10	1.11
	C2	I prefer learning programming using the smart learning solution rather than the white/blackboard style.	1.62	0.85
Teachers (*N* = 15)	C1	I have been teaching programming courses using the mobile smart learning solution.	3.67	1.11
	C2	I prefer teaching programming using the smart learning solution rather than the white/blackboard style.	1.67	0.49

Regarding the teachers' experience of the use of the SLE, most of them responded that they had not used the SLE for introductory programming. As for interest in learning programming education using the SLE, Table 3 shows that, interestingly, both students and teachers prefer the SLE for programming courses to the conventional whiteboard.

4.3 Students' and teachers' contributions towards the SLE design

To answer the RQ about the implementation of the SLE, the analysis results of the quantitative data for students and teachers are presented in Tables 4 and 5, respectively.

Figure 3 Students' opinions on the SLE design (see online version for colours)



Teachers tend to show concern for a learning environment that can help in evaluating students' performance, make teaching more interesting; work both online and offline and have proper feedback mechanisms (see Table 5 and Figure 4).

Table 6 presents the mean scores for each construct reflecting the responses by the students and teachers regarding the components and features of the SLE.

Further computation to test whether there is a statistically significant difference in preferences for features of SLE among users (students and teachers) was conducted using the Mann-Whitney test (Pallant, 2005). For this test, the categorical variable is users (students and teachers), while the continuous variable is the preference scores. The result shows that the Z value is –0.12 and asymp. sig. (two-tailed) is 0.91. The probability value ($p = 0.91$) is not less than or equal to 0.05; hence, there is no statistically significant difference in the preference for SLE students and teachers.

Table 4 Students' opinions on the SLE design

Constructs (C)		*M*	*SD*
C1	In a mobile learning situation, the environment and location at any point in time affect my understanding and performance.	2.19	1.05
C2	I would like the mobile smart learning system to keep a record of my profile and learning progress.	1.81	0.85
C3	I would like the mobile smart learning system to be adaptive to the device screen resolution and to personalise learning.	1.62	0.72
C4	I would prefer the mobile smart learning system to be implemented on a device that has features such as sensors, camera, RFID and speakers.	1.86	0.89
C5	I would like the smart learning system to be developed with robust back-end and front-end technologies and be flexible to work both online and offline.	1.67	0.71
C6	I would prefer a smart learning system that has analytics and can evaluate my performance and take certain decisions to improve my learning.	1.90	0.87
C7	I prefer a smart learning system that offers a tutorial, learning guides to support learners and feedback mechanisms.	1.57	0.79
C8	I like a smart learning system that allows automatic scheduling of tasks.	1.76	0.81
C9	I prefer a smart learning environment with components (e.g., puzzles) that motivate learning.	1.81	0.96

Figure 4 Teachers' opinions on the SLE design (see online version for colours)

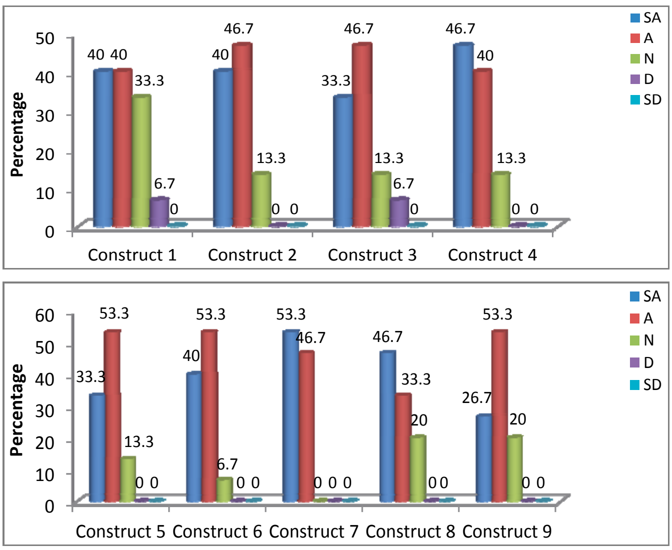


Table 5 Teachers' opinions on the SLE design

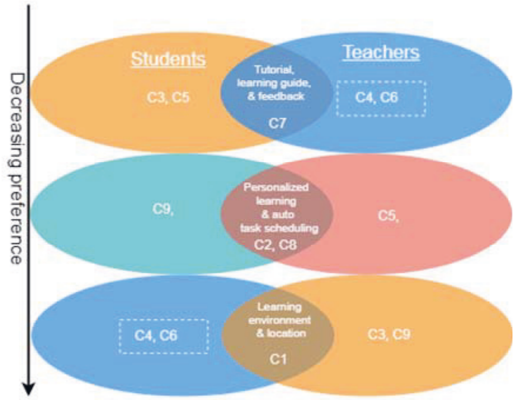
Constructs(C)		*M*	*SD*
C1	In a mobile learning situation, the environment and location at any point in time affect learners' understanding and performance.	1.87	0.92
C2	I would like the mobile smart learning system to keep a record of my profile and teaching processes.	1.73	0.70
C3	I would like the mobile smart learning system to be adaptive to the device screen resolution and personalise learning/teaching.	1.93	0.89
C4	I would prefer the mobile smart learning system to be implemented on a device that has features such as sensors, camera, RFID and speakers.	1.67	0.72
C5	I would like the smart learning system to be developed with robust back-end and front-end technologies and be flexible to work both online and offline.	1.80	0.68
C6	I prefer a smart learning system that has analytics and can evaluate my students' performance and take certain decisions on how to improve it.	1.67	0.62
C7	I prefer a smart learning system that offers a tutorial, learning guides to support learners and feedback mechanisms.	1.47	0.52
C8	I like a smart learning system that allows automatic scheduling of tasks.	1.73	0.80
C9	I prefer a smart learning environment with components (e.g., puzzles) that motivate learning.	1.93	0.70

Table 6 Students' and teachers' mean scores regarding the SLE features

Constructs (C)	*Students mean (SM)*	*Teachers mean (TM)*
C1	2.19	1.87
C2	1.81	1.73
C3	1.62	1.93
C4	1.86	1.67
C5	1.67	1.80
C6	1.90	1.67
C7	1.57	1.47
C8	1.76	1.73
C9	1.81	1.93

Figure 5 draws on the results of the constructs (C), the student mean (SM) scores and the teacher mean (TM) scores in Table 5. It can be seen from Figure 5 that students and teachers recognise tutorials, learning guides and feedback mechanism in C7 as the most important features of the SLE. Figure 5 also shows that personalised learning (i.e., user profiling) and automatic task scheduling (C2 and C8) are the second most important features. Finally, location awareness in C1 is the least important feature of the SLE. The analysis did not reveal any common grounds between students' and teachers' responses for the other SLE features (i.e., C3, C4, C5, C6 and C9). However, we speculate that, since C4 and C6 happened to be teachers' first and students' last preferences, it is likely there exist a likelihood of them being the preferred features of SLEs with the least priority.

Figure 5 The identified SLE features in descending order of priority (see online version for colours)



4.4 *Qualitative analysis: challenges of teaching/learning programming and expectations of the SLE design features*

Although the qualitative data are complementary to the quantitative data, the mixed results of this section reveal the students' expectations of the SLE features, which can guide the design and challenges of teaching/learning programming.

4.4.1 *Expectations*

The quantitative analysis revealed that students had no experience of the SLE; however, they expressed certain expectations about the SLE features, which are presented in this section. For example, one of the participants expressed that "smart learning environment should be able to classify learners and the learning content into beginner, average and master levels." Another participant added this comment: "I prefer to learn with the smart learning environment, because it allows access to learning materials in different file formats."

One of the participants wished for a kind of feedback mechanism that could stimulate learning: "I prefer a smart learning environment that has a grading or credit rewarding mechanism to encourage learning." Some of the participants were concerned about the interactivity and adaptive interface of the system. In a different vein, some of the participants commented that if learning could be accessed via the readily available smartphones and other affordable handheld portable devices, learning of computer programming could then become more accessible and motivating: "Since the smartphone is always in our hands, at any time one can log in to engage in learning." These expectations and demands form part of the principles that guide the design of SLEs (see Table 4).

Table 7 The potential SLE design features

The SLE components	*Design features*	*Design guidelines*
Context	• Learning environment (location) • Learner's preferences and scenarios	Consider the learner's location at any instance (students can learn computer programming while on a bus, in the lab, in the class, at cafeteria, etc.). For example, a student's response to an interview question in our earlier analysis was thus: 'I like learning in an environment that is not noisy, like in the library or …'. Users can specify the learning module and style by quickly responding to a few survey questions. Other learning scenarios are equally important.
Users	• Personalised learning • Adaptive interface • Responsive design to screen resolution	As depicted in Table 6, which indicates the prominent and common features of the smart learning system among students and teachers, the learning experience should be personalised, and feedback needs to be given to the learner at different stages of learning. The system should have logs for users and to track their performance. The system has to have the ability to adapt to the user's context, which can be determined, specified or implied. Screen resolution should be responsive (adjustable to devices' screen resolution) without losing tabs, menus or features.
Devices	• Mobile devices with smart features and sensors	In conjunction with the results of the study that indicate that the majority of the students and all the teachers own smartphones, it is important to design an SLE that is compatible with devices such as smartphones, tablets, PDAs and wearable devices. The features of these devices allow the acquisition of context/contents of the users through camera, GPS, RFID or speaker.
Technology	• Network communication • Server • Databases • Web services • Front and back-end technologies	The internet, wireless connections and cloud technology are needed to ensure a seamless flow of communication between the system and the users. However, due to limited resources, some students wished to have an offline system to subscribe to the internet. For example, students ($M = 1.67$) and teachers ($M = 1.80$) indicated that they would want SLEs to be accessible both offline and online.
Pedagogy	• Learning content • Task scheduling • Performance evaluation • Feedback mechanism • Supporting tips and guide • Motivation to learn more	According to our quantitative results (see Figure 3) and a student who preferred an SLE with a grading or credit rewarding mechanism to stimulate learning, it would be appealing to learners if quiz-like features were integrated into the design. Similarly, learning analytics are required to measure learners' performance, especially in a programming task. For any correct or incorrect attempt, users should receive feedback. In case of an incorrect attempt, detailed analysis should be provided for quick correction. Learning guide, tips and other support tools will be helpful to both novice and amateur computer programmers. The system should also allow for inputting, editing and updating contents.

4.4.2 Challenges

The participants remarked that the current method of programming education hardly enhances their understanding. Computer programming has been taught under poor conditions where teachers, teaching materials and facilities for practical sessions are extremely limited. One of the year four students remarked thus: “We studied programming courses right from year two using the whiteboard and the marker; the lecturers come into the class and explain on the whiteboard; we then research on our own.” Another respondent added this observation: “Sometimes the lecturer comes into the class with a soft copy of learning materials and explains them; for example, Visual Basic was taught using the same method.” During the interview, when they were asked how, for example, they usually practised programming courses after class, one student remarked thus: ‘Those that have laptops, install the programming tool on their laptops and learn with the guide from an online tutorial.”

Moreover, certain factors have been identified to influence the learners’ learning experience. For example, the students appreciated the impact of the environment during learning especially programming. One of the respondents made this comment: “I like learning in an environment that is not noisy, like in the library or a secluded area where I can stay focused and not be distracted.” The participants also discussed the issue of devices that are capable of enhancing smart learning. Although the majority of the students owned smartphones, the cost of the Internet subscription was of great concern to them. Many acknowledged that they would not be able to afford the internet connectivity continuously and hence preferred a smart learning solution that could be accessed offline.

The learner’s emotional state has been recognised as a factor that affects learning; other factors are the material and means of learning. Other themes and expectations about the SLE components (i.e., context, location, preferences and devices) were discussed by the students. For example, they expressed interest in being able to learn from any location rather than the traditional classroom or laboratory setting. In addition, the harsh weather conditions because of high temperatures and scorching sun, the sitting arrangements and the number of students per class were some of the issues raised. Hence, they prefer a system that makes learning convenient, flexible and stress-free. One of the respondents remarked thus: “I do not always like the sitting arrangement in my class; there are too many students and sometimes we have to use a poorly ventilated classroom with scorching weather condition.”

4.5 The potential SLE design features based on the study

In this section, we present the tentative design features of the SLE for programming education. The design features emanate from the outcome of this study. The convergence of interest among the students and the teachers of the researched HEIs in the Nigerian context illustrate the connection between the SLE features and the SLE components (see Figure 1 and Table 6).

5 Discussion

The objective of this study was to investigate the extent of awareness and use of the SLE in Nigerian HEIs; it also intended to identify the potential features and offer guidelines for designing an SLE for programming education in Nigeria, based on the reflections from the students and teachers. As for the components of the SLE, the results of the study showed that the majority of the students (76.2%) owned smartphones (see Table 2), which is consistent with the results of the previous researchers (e.g., Periera and Rodrigues, 2013). In contrast, all the teachers (100%) who participated in the study owned smartphones.

The results suggested that several students and teachers knew about the SLE. Comparing the extent of awareness between the students and the teachers, we found that the teachers had greater awareness of the SLE than the students. Respecting the use of the SLE for learning or teaching introductory programming, both the students and teachers said that they had not used the SLE for learning and teaching; this may be because SLEs are not available for teaching and learning in Nigeria. This finding echoes Kamba's (2009) concern. The first author's doctoral research aimed to bridge this gap by providing a smart learning tool for programming education in Kogi State HEIs.

In addition, this study found that the students preferred a learning environment that could be geared towards their needs, stimulate learning, personalise individual learning and be accessed online and offline; a possible explanation for these preferences may be that the extant traditional learning methods lack motivation. According to the students, the desire for learning computer programming can be roused by providing adequate computer and Internet facilities in the laboratory. Occasionally, teachers use the whiteboard to present programming materials without practical sessions for the students to apply the acquired knowledge. Therefore, students have to make further effort to practise the programming topics.

Another important finding was that teachers desired a smart system that aided them in teaching programming and allowed efficiency and evaluating student performance. The findings of this study are consistent (Chao et al., 2013), which found that students and teachers desire methods that have a direct impact on the pedagogy of programming education and a system that makes the learning of programming interesting.

Similarly, our study showed that the students and teachers shared a common interest in the features of the smart learning system; for example, they both expected the system to be robust and flexible, accessible online and offline, support quick feedback, contain teaching guide and tips, allow user profiling and personalising, and stimulate continuous learning. More specifically, the study showed that learning guides and feedback mechanisms are the prominent features of the SLE. The next important features are learners profiling, personalisation and automatic task scheduling. However, location awareness was given the lowest priority among the SLE features.

Moreover, the identified potential features of the SLE are essential and can inform when modelling an SLE for programming education. Furthermore, it is important to note that both the students and teachers were found interested in embracing the relevant features of the SLE for programming education, which is important to be considered when designing an SLE. On this basis, we propose an SLE framework that encompasses different components and features of the SLE relevant to computing education.

6 Conclusions and future work

This paper discussed the advancement of learning technology, the transition of learning environments and the significance of the SLE for the education system. This study offered insightful SLE design guidelines for programming education in Nigeria. In addition, this study illustrated the proportion of the students and teachers who own smartphones in the researched HEIs in Nigeria. We found that both students and teachers knew about the SLE, although they lacked sufficient experience of teaching and learning when using it. Interestingly, the results revealed that they are willing to make full use of their smartphones to learn programming education. These findings serve to justify the authors' intention of implementing the SLE for programming education in Nigeria. Our results also reflect the students' and teachers' opinions on the components of the SLE for learning and teaching introductory programming in Nigerian HEIs. A noteworthy contribution of this work is the connection between the features and components of SLEs. To the best of our knowledge, this connection has not been discussed by previous researchers. Hence, it is important that future research pays more attention to this relationship.

The scope of this study was confined to the students and teachers of programming education in Nigerian HEIs. This perceived limitation may be addressed by including other stakeholders (e.g., the government and educational agencies) whose views may provide a wider perspective on the use of the SLE for programming education. In conclusion, the authors recommend using the SLE for programming education in Nigerian HEIs to enhance students' learning experience. The SLE designers are advised to consider the identified features when designing a smart learning solution. As part of the first author's doctoral studies, future work should involve defining the SLE requirements for programming education in line with the identified potential features and forthcoming prototypes.

References

Abtar, D.S. and Hassan, M. (2017) *In Pursuit of Smart Learning Environments for the 21st Century*, Progress Reflection No. 12 on Current and Critical Issues in Curriculum, Learning and Assessment.

Airth, M. (2018) *Social Awareness: Definition, Example & Theories* [online] https://study.com/academy/lesson/social-awareness-definition-example-theories.html (accessed 12 August 2018).

Akbari, F. and Taghiyareh, F. (2014) 'E-SoRS: a personalized and social recommender service for e-learning', *The International Conference on E-learning and E-teaching*, pp.1–12.

Akinola, O.S. and Nosiru, K.A. (2014) 'Factors influencing students' performance in computer programming: a fuzzy set operations approach', *International Journal of Advances in Engineering & Technology*, Vol. 7, No. 4, pp.1141–1149.

Annamaa, A., Suviste, R. and Vene, V. (2017) 'Comparing different styles of automated feedback for programming exercises', *Proceedings of the 17th Koli Calling International Conference on Computing Education Research*, ACM, pp.183–184.

Arends, H., Heeren, B., Keuning, H. and Jeuring, J. (2017) 'An intelligent tutor to learn the evaluation of microcontroller I/O programming expressions', *Proceedings of the 17th Koli Calling International Conference on Computing Education Research*, ACM, pp.2–9.

Barnes, D.J. and Kolling, M. (2006) *Objects First with Java – A Practical Introduction Using BlueJ*, 3rd ed., Prentice Hall/Pearson Education, USA.

Bekhet, A.K. and Zauszniewski, J.A. (2012) 'Methodological triangulation: an approach to understanding data', *Nursing Faculty Research and Publications Marquette University*, Vol. 20, No. 2, pp.40–43.

Ben-Ari, M. (1998) 'Constructivism in computer science education', *SIGCSE '98 Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education*, ACM, pp.257–261.

Chang, W., Tan, Q. and Fang, W.T. (2010) 'Multi-object oriented augmented reality for location-based adaptive mobile learning', *Proceeding of 2010 IEEE 10th International Conference on Advanced Learning Technologies*, pp.450–451.

Chao, M., Edwin, B. and Hussein, S. (2013) 'A mobile scaffolding application to support novice learners of computer programming', *ICTD '13 Proceedings of the Sixth International Conference on Information and Communications Technologies and Development*, ACM, Cape Town, South Africa, pp.84–87.

Ching-Bang, Y. (2017) 'Constructing a user-friendly and smart ubiquitous personalized learning environment by using a context-aware mechanism', *IEEE Transactions on Learning Technologies*, Vol. 10, No. 1, pp.104–114.

Cooper, S., Dann, W. and Pausch, R. (2000) 'Alice: a 3-D tool for introductory programming concepts', *The Journal of Computing in Small Colleges*, Vol. 15, No. 5, pp.107–116.

Edwards, S.H. and Perez-Quinones, M.A. (2008) 'Web-cat: automatically grading programming assignments', in *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*, ACM, p.328.

Fessakis, G., Gouli, E. and Mavroudi, E. (2013) 'Problem solving by 5–6 years old kindergarten children in a computer programming environment: a case study', *Computers & Education*, Vol. 63, pp.87–97.

Gajraj, R.R., Williams, M., Bernard, M. and Singh, L. (2011) 'Transforming source code examples into programming tutorials', in the *Sixth International Multi-conference on Computing in the Global Information Technology, ICCGI*, pp.160–164.

Gerdes, A., Heeren, B., Jeuring, J. and Binsbergen, L.T. (2016) 'Ask-Elle: an adaptable programming tutor for Haskell giving automated feedback', *International Journal of Artificial Intelligence in Education*, Vol. 27, No. 1, pp.65–100.

Graf S. and Kinshuk (2008) 'Adaptivity and personalization in ubiquitous learning systems', in Holzinger, A. (Eds.): *HCI and Usability for Education and Work, USAB 2008, Lecture Notes in Computer Science*, Vol. 5298, Springer, Berlin, Heidelberg.

Guzdial, M. and Soloway, E. (2002) 'Log on education: teaching the Nintendo generation to program', *Communications of the ACM*, Vol. 45, No. 4, pp.17–21.

Ha, I. and Kim C. (2014) 'The research trends and the effectiveness of smart learning', *International Journal of Distributed Sensor Networks*, Vol. 10, No. 5, pp.1–9.

Helminen, J. (2009) *Jype – An Education-oriented Integrated Program Visualization, Visual Debugging, and Programming Exercise Tool for Python*, Master's thesis, Department of Computer Science and Engineering, Helsinki University of Technology.

Huang, H., Wang, T. and Hsieh, F. (2012a) 'Constructing an adaptive mobile learning system for the support of personalized learning and device adaptation', in *Proceedings of International Educational Technology Conference*, Elsevier, ScienceDirect, Vol. 64, pp.332–341.

Huang, R., Yang, J. and Hu, Y. (2012b) 'From digital to smart: the evolution and trends of learning environment', *Open Education Research*, Vol. 1, No. 1, pp.75–84.

Huang, R., Du, J., Chang, T., Spector, M., Zhang, Y. and Zhang, A. (2017) 'A conceptual framework for a smart learning engine', in Popescu, E. et al. (Eds.): *Innovations in Smart Learning. Lecture Notes in Educational Technology*, pp.69–73, Springer, Singapore.

Huang, Y.M., Lin, Y.T. and Cheng, S.C. (2010) 'Effectiveness of a mobile plant learning system in a science curriculum in Taiwanese elementary education', *Computers & Education*, Vol. 54, No. 1, pp.47–58.

Hwang, G. (2014) 'Definition, framework and research issues of smart learning environments – a context-aware ubiquitous learning perspective', *Springer Journal of Smart Learning Environments*, Vol. 1, No. 4, pp.1–14.

Hwang, G.J., Tsai, C.C. and Yang, S.J.H. (2008) 'Criteria, strategies and research issues of context-aware ubiquitous learning', *Journal of Educational Technology & Society*, Vol. 11, No. 2, pp.81–91.

Jill, A. and Carol, S. (2004) *Mobile Learning Anytime Everywhere*, pp.12229–12246, UK Learning and Skills Development Agency.

Jiménez-Díaz, G., González-Calero, P.A. and Gómez-Albarrán, M. (2011) 'Role-play virtual worlds for teaching object-oriented design: the ViRPlay development experience', *Software – Practice and Experience*, Vol. 42, No. 2, pp.235–253.

Joshi, A., Kale, S., Chandel, S. and Pal, D.K. (2015) 'Likert scale: explored and explained', *British Journal of Applied Science & Technology*, Vol. 7, No. 4, pp.396–403.

Kamba, M.A. (2009) 'Problems, challenges and benefits of implementing e-learning in Nigerian universities: an empirical study', *International Journal of Emerging Technologies in Learning*, Vol. 4, No. 1, pp.66–69.

Kawahara, Y., Minami, M., Morikawa, H. and Aoyama, T. (2003) *Areal-world Oriented Networking for Ubiquitous Computing Environment*, SIG Technical Reports No. UBI-1-1, Vol. 39, pp.1–6.

Kordaki, M. (2010) 'A drawing and multi-representational computer environment for beginner learning of programming using C: design and pilot formative evaluation', *Computers & Education*, Vol. 54, No. 1, pp.69–87.

Korhonen, A., Malmi, L. and Silvasti, P. (2003) 'TRAKLA2: a framework for automatically assessed visual algorithm simulation exercises', in *Proceedings of the Third Annual Baltic Conference on Computer Science Education*, Joensuu, Finland, pp.48–56.

Kumar, A.N. (2013). 'Using Problets for problem-solving exercises in introductory C++/Java/C# courses', *Frontiers in Education Conference*, IEEE, Oklahoma City, OK, USA.

Laakso, M-J., Rajala, T., Kaila, E. and Salakoski, T. (2008) 'The impact of prior experience in using a visualization tool on learning to program', in *Proceedings of Cognition and Exploratory Learning in Digital Age, CELDA '08*.

Laine, H.T. and Joy, M. (2009) 'Survey on context-aware pervasive learning environments', *International Journal of Interactive Mobile Technologies*, Vol. 3, No. 1, pp.70–76.

Levy, R.B-B. and Ben-Ari, M. (2007) 'We work so hard and they don't use it: acceptance of software tools by teachers', *SIGCSE Bulletin*, Vol. 39, No. 3, pp.246–250.

Liu, G. and Hwang, G. (2010) 'A key step to understanding paradigm shifts in e-learning: towards context-aware ubiquitous learning', *British Journal of Educational Technology*, Vol. 41, No. 2, pp.1–9.

Macredie, R.D., Chen, S.Y. and Fan, J. (2006) 'Navigation in hypermedia learning systems: experts vs. novices', *Computer Human Behavior*, Vol. 22, No. 2, pp.251–266.

Mandel, J.C., Kreda, D.A., Mandl, K.D., Kohane, I.S. and Ramoni, R.B. (2016) 'SMART on SMART on FHIR: a standards-based, interoperable apps platform for electronic health records', *Journal of the American Medical Informatics Association*, Vol. 23, No. 5, pp.899–908.

Martin, T., Berland, M., Benton, T. and Smith, C.P. (2013) 'Learning programming with IPRO: the effects of a mobile, social programming environment', *Journal of Interactive Learning Research*, Vol. 24, No. 3, pp.301–328.

Moreno, A., Myller, N., Sutinen, E. and Ben-Ari, M. (2004) 'Visualizing programs with Jeliot 3', in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ACM, pp.373–376.

Opgen-Rhein, J., Kuppers, B. and Schroeder, U. (2018) 'An application to discover cheating in a digital exams', *Proceedings of the 18th Koli Calling International Conference on Computing Education Research*, ACM, pp.22–25.

Pallant, J. (2005) *SPSS Survival Manual: A Step by Step Guide to Data Analysis Using SPSS*, Ligare, Sydney.

Periera, O.R.E. and Rodrigues, J.P.C. (2013) 'Survey and analysis of current mobile learning applications and technologies', *ACM Computing Surveys*, Vol. 46, No. 2, pp.1–35.

Peter, S.E., Bacon, E. and Dastbaz, M. (2010) 'Adaptable, personalized e-learning incorporating learning styles', *Campus-Wide Information Systems*, Vol. 27, No. 2, pp.91–100.

Quevedo-Torrero, J.U. (2009) 'Learning theories in computer science education', *The Sixth International Conference on Information Technology: New Generations*, Las Vegas, Nevada, pp.1634–1635.

Raymond, G. (1992) *Basic Interviewing Skills*, F.E. Peacock, Itasca, IL [online] http://www.damiantgordon.com/Courses/ResearchMethods/Lectures/Week4/5-CodingInterviewResponses.pdf (accessed 14 May 2019).

Renny, S.N., Hasanov, L.A. and Laine, T.H. (2017) 'Improving play and learning style adaptation in a programming education game', *Proceedings of the 9th International Conference on Computer Supported Education (CSEDU)*, SCITEPRESS, pp.450–457.

Rogers, M.P. and Siever, B. (2018) 'A macro view of the micro:bit in higher education', *Consortium for Computing Sciences in Colleges*, Vol. 33, No. 5.

Roßling, G. and Freisleben, B. (2002) 'ANIMAL: a system for supporting multiple roles in algorithm animation', *Journal of Visual Languages and Computing*, Vol. 13, No. 3, pp.341–354.

Roussos, G. (2002) *Location Sensing Technologies and Applications*, TSW 02-08, School of Computer Science and Information Systems, Birkbeck College, University of London, London.

Saadiah, Y., Erny, A.A. and Kamarularifin, A.J. (2010) 'The definition and characteristics of ubiquitous learning: a discussion', *International Journal of Education and Development using Information and Communication Technology*, Vol. 6, No. 1, pp.117–127.

Sahar, Y., Seifedine, K. and Miguel-Angel, S. (2016) 'Measuring learning outcomes effectively in smart learning environments', *Proceedings of 2016 Smart Solutions for Future Cities Conference*, IEEE, Kuwait City, Kuwait, pp.1–5.

Schoonenboom, J. and Johnson, R.B. (2017) 'How to construct a mixed methods research design', *KZfSS Kölner Zeitschrift für Soziologie und Sozialpsychologie*, Vol. 69, No. 2, pp.107–131.

Shenton, A.K. (2004) 'Strategies for ensuring trustworthiness in qualitative research projects', *Education for Information*, Vol. 22, No. 2, pp.63–75.

Sorva, J. and Sirkiä, T. (2010) 'UUhistle – a software tool for visual program simulation', in *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*, ACM, pp.49–54.

Spector, J.M. (2014) 'Conceptualizing the emerging field of smart learning environments', *Smart Learning Environments, a Springer Open Journal*, Vol. 1, No. 2, pp.1–10.

Taisiya, K., Ji, Y.C. and Bong, G.L. (2013) *Evolution to Smart Learning in Public Education: A Case Study of Korean Public Education*, Vol. 295, pp.170–178, International Federation for Information Processing (IFIP).

Verdú, E., Regueras, M.L., Verdú, M.J., Leal, P.J., Castro, J.P. and Queirós, R. (2012) 'A distributed system for learning programming online', *Computers & Education*, Vol. 58, No. 1, pp.1–10.

Vesin, B., Ivanovic, M., Klasnja-Milicevic, A. and Budi-mac, Z. (2012) 'Protus 2.0: ontology-based semantic recommendation in programming tutoring system', *Expert Systems with Applications*, Vol. 39, No. 15, pp.12229–12246.

Yaghmaie, M. and Bahreininejad, A. (2011) 'A context-aware adaptive learning system using agents', *Expert Systems with Applications*, Vol. 38, No. 4, pp.3280–3286.

Yang, S.J.H., Zhang, J. and Chen, I.Y.L (2007) 'Ubiquitous provision of context-aware web services', *International Journal of Web Service Research*, Vol. 4, No. 4, pp.83–103.

Yeh, Y.F., Chen, M.C., Hung, P.H. and Hwang, G.J. (2010) 'Optimal self-explanation prompt design in dynamic multi-representational learning environments', *Computers & Education*, Vol. 54, No. 4, pp.1089–1100.

Yeonjeong, P. (2011) 'A pedagogical framework for mobile learning: categorizing educational applications of mobile technologies into four types', *The International Review of Research in Open and Distributed Learning*, Vol. 12, No. 2, pp.1–13.

Zhu, Z.T., Yu, M.H. and Riezebos, P. (2016) 'A research framework of smart education', *Smart Learning Environment*, Vol. 3, No. 1, pp.1–17.

ARTICLE IV

Agbo, F. J., Sanusi, I. T., Oyelere, S. S., & Suhonen, J.
“Application of virtual reality in computer science education: A systemic review based on bibliometric and content analysis methods,” *Education Sciences, 11*(3), 1–23. https://www.mdpi.com/2227-7102/11/3/142

education sciences

Article

Application of Virtual Reality in Computer Science Education: A Systemic Review Based on Bibliometric and Content Analysis Methods

Friday Joseph Agbo [1,*], Ismaila Temitayo Sanusi [1], Solomon Sunday Oyelere [2,*] and Jarkko Suhonen [1]

[1] School of Computing, University of Eastern Finland, FIN-80101 Joensuu, Finland; ismaila.sanusi@uef.fi (I.T.S.); jarkko.suhonen@uef.fi (J.S.)
[2] Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden
* Correspondence: friday.agbo@uef.fi (F.J.A.); solomon.oyelere@ltu.se (S.S.O.)

Abstract: This study investigated the role of virtual reality (VR) in computer science (CS) education over the last 10 years by conducting a bibliometric and content analysis of articles related to the use of VR in CS education. A total of 971 articles published in peer-reviewed journals and conferences were collected from Web of Science and Scopus databases to conduct the bibliometric analysis. Furthermore, content analysis was conducted on 39 articles that met the inclusion criteria. This study demonstrates that VR research for CS education was faring well around 2011 but witnessed low production output between the years 2013 and 2016. However, scholars have increased their contribution in this field recently, starting from the year 2017. This study also revealed prolific scholars contributing to the field. It provides insightful information regarding research hotspots in VR that have emerged recently, which can be further explored to enhance CS education. In addition, the quantitative method remains the most preferred research method, while the questionnaire was the most used data collection technique. Moreover, descriptive analysis was primarily used in studies on VR in CS education. The study concludes that even though scholars are leveraging VR to advance CS education, more effort needs to be made by stakeholders across countries and institutions. In addition, a more rigorous methodological approach needs to be employed in future studies to provide more evidence-based research output. Our future study would investigate the pedagogy, content, and context of studies on VR in CS education.

Keywords: computer science education; virtual reality; VR; content analysis; bibliometric analysis; immersion; 3D simulation; presence; game-based learning

check for updates

Citation: Agbo, F.J.; Sanusi, I.T.; Oyelere, S.S.; Suhonen, J. Application of Virtual Reality in Computer Science Education: A Systemic Review Based on Bibliometric and Content Analysis Methods. *Educ. Sci.* **2021**, *11*, 142. https://doi.org/10.3390/educsci11030142

Academic Editor: João Piedade

Received: 22 February 2021
Accepted: 16 March 2021
Published: 23 March 2021

1. Background of the Study

Virtual reality (VR) has recently become a popular technology in different contexts such as entertainment, military, and education [1]. VR combines technologies to provide an immersive presence through highly interactive objects in a virtual environment but stimulates users' sensory awareness to perceive being in an almost natural environment. The use of VR in education to support training, teaching, and learning through 3D simulation and visualization of learning content in a virtual presence has grown recently [2]. This increasing VR application growth in the educational field is evident, as revealed by the literature, including a recent VR study in computer science education [3]. VR technology provides an opportunity to develop a state-of-the-art smart learning environment with a high level of interaction, engagement, and motivation for an enhanced learning experience [1–8]. This study refers to computer science (CS) education as the art and science involved in learning and teaching computer science, including computing, algorithmic and computational thinking [9]. For example, the science behind curriculum design, pedagogical approach, and instructional tools and techniques educators adopt to support computer science teaching and learning.

This study investigated the role of VR in CS education by conducting a comprehensive content and bibliometric analysis of relevant articles published between 2011 and 2020 in journals and conferences. Bibliometric and content analysis of articles focused on VR in CS education would provide a deeper understanding of the evolution of research conducted in this field and how VR applications have advanced CS education over the years [4,10,11]. From the standpoint of bibliometric mapping analysis, this study investigates the publication growth of studies on VR in CS education within the last 10 years, reveals the most active authors and affiliations contributing to the development of VR in CS education, and anticipates the future direction on the basis of the co-occurrence pattern analysis of current studies. In addition, this study explicates the role of VR in CS education from the perspective of methodological approaches used in studies related to VR in CS education [7], the kind of data collected for such studies, the sample size, and the types of data analysis conducted.

Research on VR in education has claimed several benefits, such as positively affecting users' attitude [12,13], presenting an effective and efficient learning and training environment [14,15], and increasing students' motivation to learn within a virtual environment [14–17]. Furthermore, many systematic review studies related to VR in education have been published in recent years. However, there have been only a limited number of such studies focused on computer science education. For example, Pirker et al. [3] conducted a systematic literature review of VR in CS education, focusing on the technology used to deploy VR applications for CS education, the learning objectives, and challenges recorded in studies related to VR in CS education. Pirker and colleagues revealed that VR desktop applications using Oculus Rift and HTC Vive dominate the technology currently used to deploy VR in CS education. On the other hand, the majority of studies on VR in CS education focused on cognitive learning with topics such as fundamental components of algorithms and object-oriented programming [3].

Similarly, Oyelere et al. [1] studied VR games in CS education, focusing on developmental features such as the technology, pedagogy, and gaming elements used in such studies. In terms of technology, Oyelere et al. [1] finding was in congruence with that of Pirker et al. [3], where Oculus Rift, HTC Vive, and PC-based applications dominate the technology aspect. Both studies show that mobile-based VR applications for CS education are still growing, with less than 15% of deployment of VR applications on mobile devices.

We could find only a few studies regarding recent studies that focused on content and bibliometric analysis of articles related to VR in education. For example, Arici et al. [11] conducted content and bibliometric mapping analysis of augmented reality (AR) in science education. Lorenzo et al. [17] investigated VR articles' scientific production for inclusive learning of people with autism spectrum disorder (ASD). Sobral and Pestana [18] studied a bibliometric analysis of articles related to VR application to learn about dementia from 1998 until 2018 by focusing on articles' intellectual structure and emerging trends. Lai et al. [19] conducted a bibliometric analysis of VR research in engineering education published and indexed in the Scopus database that spans over 26 years. Thus, Lai et al. [19] provided valuable insights in terms of article production, trends, and co-occurrence network of VR studies within the field of engineering. Another bibliometric study related to VR in CS field-specific was recently conducted by Enebechi and Duffy [20]. This study [21] focused on bibliometric analysis of VR and artificial intelligence (AI) articles in mobile computing and applied ergonomics.

While all these related studies highlighted above are relevant and provided essential knowledge about the field, our current research would expand on the existing research rather than re-inventing the wheel. For example, while the work of Pirker et al. [3] mainly focused on the technology used to deploy VR application for CS education, the learning objectives, and challenges recorded in studies related to VR in CS education, our research would address the aspect of methodological approach used in studies on VR in CS education, kind of data collected for such studies, the sample size, and types of data analysis conducted. The majority of these related studies analyzed a small sample size, limiting the study, and cannot justify the generalization of their findings. For example, Pirker et al. [3] analyzed 13 pieces of data, Lorenzo et al. [17] revealed 18 articles, Lai et al. [19] conducted bibliometric analysis on 274 articles, and Enebechi and Duffy [20] presented a content analysis of 8 papers. Our study took a different approach by analyzing more extensive data to discover more profound knowledge in the field. It is worth mentioning that our study drew motivation from [11] by focusing content analysis of variables such as materials and method trends, sample sizes, and method of an investigation conducted by articles on VR in CS education in the last 10 years. The authors hope that the approach used in this study would contribute to the existing knowledge in terms of unveiling how VR has supported CS education and what scientific achievement have been made in this field.

As a result of this comprehensive content and bibliometric analysis of studies on VR in CS education, we hoped that our findings would contribute to the existing knowledge by providing a deeper understanding of VR applications' role in honing CS education over the last decade. In addition, the authors believe that this study will unveil information regarding what scholars have made a scientific achievement in this field in terms of advancing teaching and learning of CS topics in the different contexts, which will serve as a boost for active researchers. In contrast, new scholars would derive motivation and valuable resources for future studies. To achieve objectives, this study set out to answer the following research questions:

RQ1 How is the growth of research publication and citation of articles on VR in computer science education?
RQ2 Who are the most active authors, institutions, and countries publishing articles on the use of VR in computer science education?
RQ3 What co-occurrence patterns exist in studies on the use of VR in computer science education?
RQ4 What is the trend of the research methodology employed in articles on VR in computer science education?
RQ5 What were the most preferred data collection tools and sampling methods in articles on the use of VR in computer science education?
RQ6 What were the sample sizes in articles on the use of VR in computer science education?
RQ7 What were the most preferred data analysis methods in articles on the use of VR in computer science education?

2. Methods

The method explored in this study was centered on content and bibliometric mapping analysis. This study followed the recommended workflow for science mapping provided by Aria and Coccurullo [21] to conduct our bibliometric mapping analysis. In contrast, the approach shown by [11] was followed to present the content analysis, respectively.

Article selection process

The article selection process for this study includes 3 phases similar to the one presented by [4], namely, (i) literature search and data collection; (ii) data extraction, loading, and conversion; and (iii) data synthesis. A graphical representation of the data collection process is presented in Figure 1, showing detailed actions in each phase.

(i) Literature search and data collection

This study obtained data from 2 databases, i.e., the Web of Science (WoS) and the Scopus databases. These 2 databases have been acclaimed to contain comprehensive data of scientific outputs relevant to this study [14]. To conduct an extensive data collection needed for this study, we define the search keywords to include "virtual reality" "VR", "computer science", and "computing education". A number of common protocols for data collection were applied to both databases. They include the same search keywords used in combination with the binary operators such as "OR" and "AND" across the 2 databases, limited time span to the period from 2011 to 2020, and language selected as "English". Table 1 presents details of the search protocol, how they were applied in each database, and the result obtained.

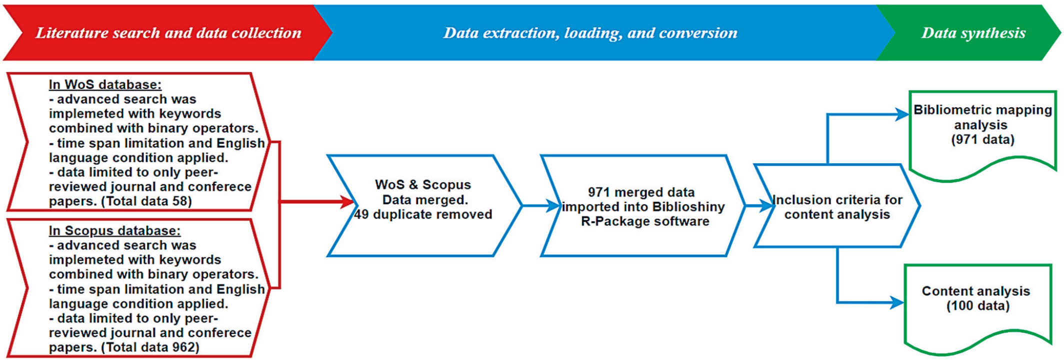


Figure 1. The procedure followed to obtain data used for bibliometric and content analysis.

Table 1. Data search procedures and obtained amount of data.

Database	Description of the Protocol	Combination of Search String Based on Database Algorithm	Search Outcome
WoS	Applying the search keywords in quotation to the WoS TOPIC field with binary operators.	TOPIC: ("virtual reality" OR "VR") AND TOPIC: ("computer science" OR "computing education").	80
	Additional conditions were applied by limiting the results to only articles and proceedings papers, with time span set to 2011–2020.	TOPIC: ("virtual reality" OR "VR") AND TOPIC: ("computer science").Refined by: DOCUMENT TYPES: (ARTICLE OR PROCEEDINGS PAPER) AND PUBLICATION YEARS: (2020 OR 2014 OR 2019 OR 2013 OR 2018 OR 2012 OR 2017 OR 2011 OR 2016 OR 2015)Timespan: All years. Indexes: SCI-EXPANDED, SSCI, A&HCI, ESCI.	58
Scopus	Applying the search keywords in quotation to Scopus title, abstract, and keywords field with binary operators and limiting the time span to 2011–2020.	(TITLE-ABS-KEY ("virtual reality" OR "VR") AND TITLE-ABS-KEY ("computer science" OR "computing education")) AND PUBYEAR > 2010 AND PUBYEAR < 2021.	1058
	Applying additional conditions by limiting to only articles and conference papers.	(TITLE-ABS-KEY ("virtual reality" OR "VR") AND TITLE-ABS-KEY ("computer science" OR "computing education")) AND PUBYEAR > 2010 AND PUBYEAR < 2021 AND (LIMIT-TO (DOCTYPE, "cp") OR LIMIT-TO (DOCTYPE, "ar")) AND (LIMIT-TO (LANGUAGE, "English")).	962
		After merging both files, we removed 49 duplicated documents.	971 Total

(ii) Data extraction, loading, and conversion

After data from the independent databases were collected and downloaded in BibTex format, we conducted data extraction and conversion into a comma-separated values CSV file to merge the 2 datasets from WoS and Scopus. The process of merging the data is presented in Table 2, followed by executing command line instructions (CLI) shown in Figure 2. R-studio is an integrated development environment for R programming language (https://rstudio.com, accessed on 18 January 2018) software was used to combine the data into a single CSV file before uploading it to biblioshiny (Biblioshiny is a web interface for bibliometrix r-package (https://www.bibliometrix.org/Biblioshiny.html, accessed on 18 January 2018) for bibliometrix R-package [17].

Table 2. Data conversion and merging steps.

Steps	Instructions on How to Merge Two Points of Data from WoS and Scopus Databases
1	Download in BibTex format independently from databases (in this case, WoS and Scopus).
2	Save data in a directory with a name that says “rawData”.
3	Open RStudio and import the bibliometrix library by running the script < library(“bibliometrix”) > in the command-line interface (CLI).
4	In Rstudio CLI, run the script < setwd (“C:/../ . . . / . . . /rawData”) > to open the directory where data would be imported from and saved. Not that the ellipsis (. . .) indicates the paths to the directory and should be correctly inserted.
5	Download in BibTex format independently from databases (in this case, WoS and Scopus).
6	Save data in a directory with a name that says “rawData”.

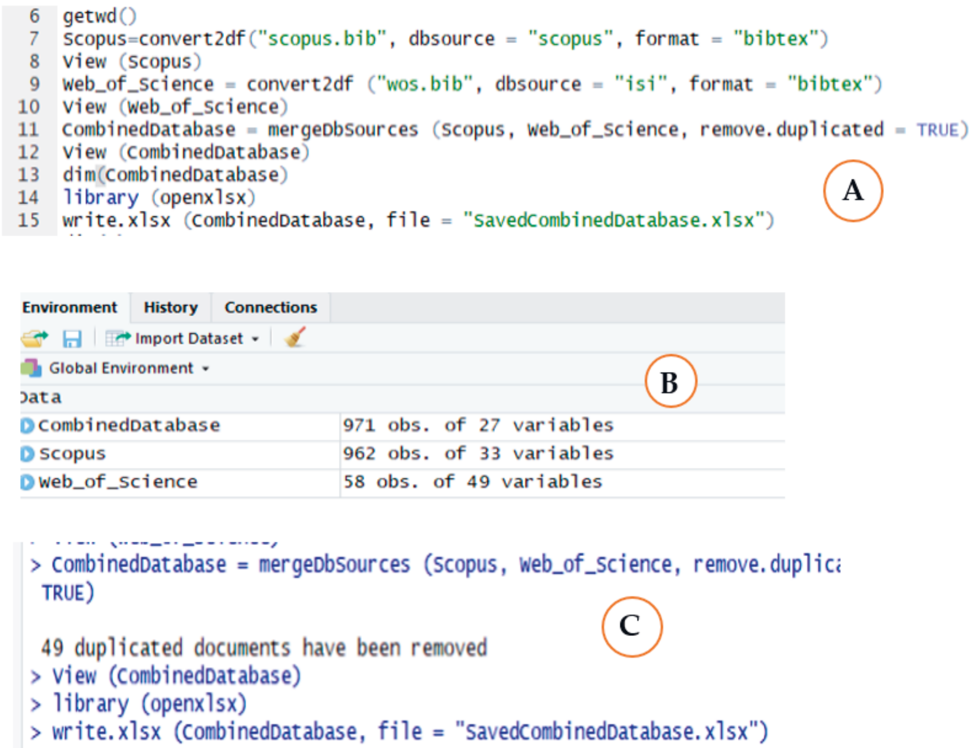


Figure 2. (**A**) shows the set of commands to be executed in R-Studio command line instructions (CLI) to implement the conversion and to merge of data downloaded from Web of Science (WoS) and Scopus databases; (**B**) shows the output of the executed commands; (**C**) depicts the console for the CLI where the line execution returns a value including line errors.

After completing the steps in Table 2, we executed the line of commands (lines 6 to 15) in Figure 2 to complete the remaining process of data conversion and merging. This merging of the two converted points of data by running the command in line 11 of Figure 2A triggered the R- Function that identified 49 similar articles from WoS and Scopus databases. The identified similar articles were removed to avoid having duplicate data. Removing duplicate articles left the remaining data at 971, which was uploaded to biblioshiny for bibliometric mapping analysis. The search was conducted on 2 January 2021.

(iii) Data Synthesis

In Table 3, we present the synthesized data used for the bibliometric analysis. However, for the content analysis, 3 researchers screened the entire data by reading each paper's abstract to decide whether it was relevant or not. Further criteria for selecting relevant papers suitable for the content analysis included:

Table 3. Data synthesis indicating the primary information about the data and document type.

Description	Results
Main information about data	
Timespan	2011–2020
Sources (journals, books, etc.)	378
Documents	971
Average years from publication	4.53
Average citations per documents	3.754
Average citations per year per doc	0.7841
References	21,021
Document types	
Article	157
Conference paper	814
Document contents	
Keywords plus (ID)	6281
Author's keywords (DE)	2848
Authors	
Authors	2738
Author appearances	3308
Authors of single-authored documents	98
Authors of multi-authored documents	2640
Author collaboration	
Single-authored documents	102
Documents per author	0.355
Authors per document	2.82
Co-authors per documents	3.41
Collaboration index	3.04

(i) the paper must focus on virtual reality for education in computer science education;
(ii) the paper designed a study or developed a solution to facilitate CS education in a VR environment;
(iii) the study reported any outcome by evaluating with users (students, educators, or experts);
(iv) the paper is open access and could be downloaded for detailed review.

After applying the criteria, we arrived at 39 papers that met the content analysis requirements presented in Section 3.2.

3. Results

3.1. Findings from Bibliometric Mapping Analysis

This section presents our findings from the bibliometric analysis on the basis of the data generated from WoS and Scopus databases. This bibliometric analysis intends to provide insight into how studies on the use of VR for CS education have grown in the last 10 years. In addition, the result reveals authors, institutions, and countries who have been contributing to the field by actively publishing research related to VR in CS education. Furthermore, the result presents how studies on VR in CS education have had an impact in terms of their citations and authors co-occurrence pattern analysis. The section delineates the analysis of common keywords used in articles on VR for CS education, thereby presenting the thematic area of the current research landscape and topic hotspots.

3.1.1. Research Publication Growth of Articles on the Use of VR in Computer Science Education

Figure 3 shows the articles' distribution in terms of the publication year regarding the article production and development across 10 years. The overall publication trend of articles related to VR in CS education shows that 2011 witnessed the highest production year, reaching 148 articles, followed closely by 135 articles in 2018.

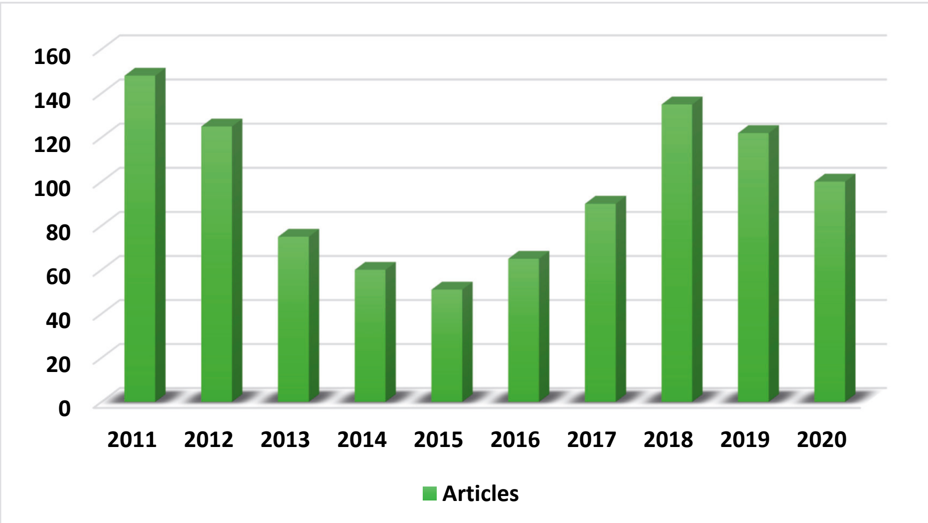


Figure 3. Annual scientific production of articles on virtual reality (VR) in computer science (CS) education.

The publication volume decreased from 2012 to 2015 and from 2019 to 2020. There was an increase in article production from 2016 to 2018 before the slight decline until 2020. This trend occurred probably because the selected articles were limited to only education, leaving out other domains, such as health, business, entertainment, and media.

3.1.2. Most Active Authors, Institutions, and Countries Publishing Articles on the Use of VR in Computer Science Education

Regarding authors' production over time, we investigated the top 20 authors. Our findings showed that most of those top authors were already publishing articles on VR in CS education by 2011. However, about half of those authors were not active from 2019. As shown in Figure 4, many articles related to VR in CS education were published between 2011 and 2020.

As we can see in Figure 4, the author Li Y. had the highest publication over time, having had several articles published yearly for 7 years from 2011 to 2020, except in 2013, 2014, and 2016. With the least productivity over time was the author is Dengel A., with publications only in 2019 and 2020.

We analyzed the top 20 authors' number citations across the production years (m-index) regarding their impact. M-index is calculated by dividing the total number of citations by the total number of years of production. In order words, this study measures the authors' impact by dividing the H-index by the total number of years of production. Note that the total years of production varied for different authors. Although the total number of years investigated in this study remained at 10, some authors did not start publishing from 2011; therefore, such an author's total number of years of production would count from the year the author published his/her first paper. For example, Dengel A. started publishing articles on VR in CS education in 2019; hence, the total number of years remained at two. Therefore, the m-index would be the total number of citations in 2019 and 2020, divided by 2.

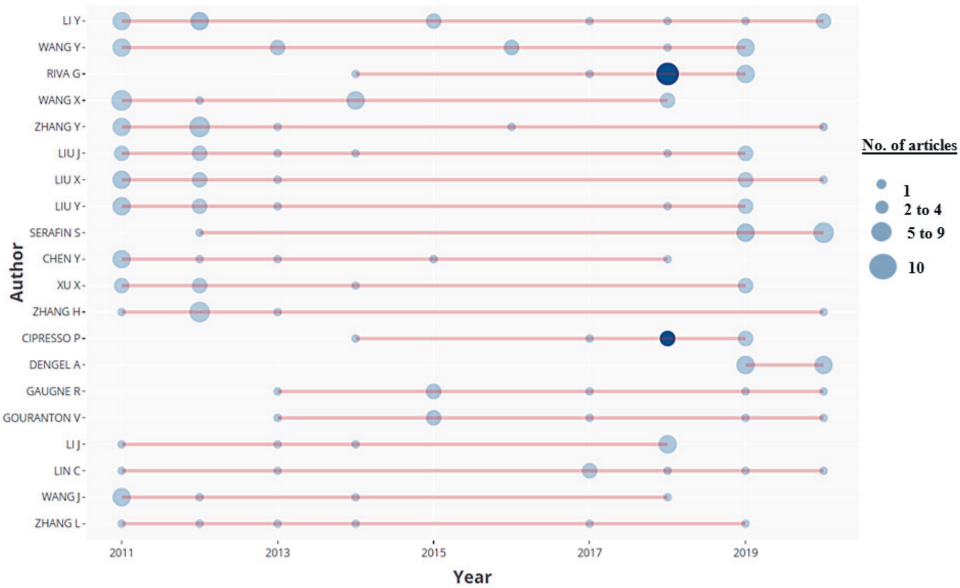


Figure 4. Top 20 authors publishing articles on VR in CS education between 2011 and 2020: the size of each circle indicates the number of articles. The amount of boldness of the circles shows the number of citations in that year.

As shown in Figure 5, the authors' m-index was highest at 1.0 (to a single decimal). Therefore, the result indicates that Dengel A., with the highest m-index, remained the most impactful author at the end of 2020. This finding suggests that Dengel A. had had an unbroken research activity in the area of VR in CS education since the first publication and had received a significant number of citations.

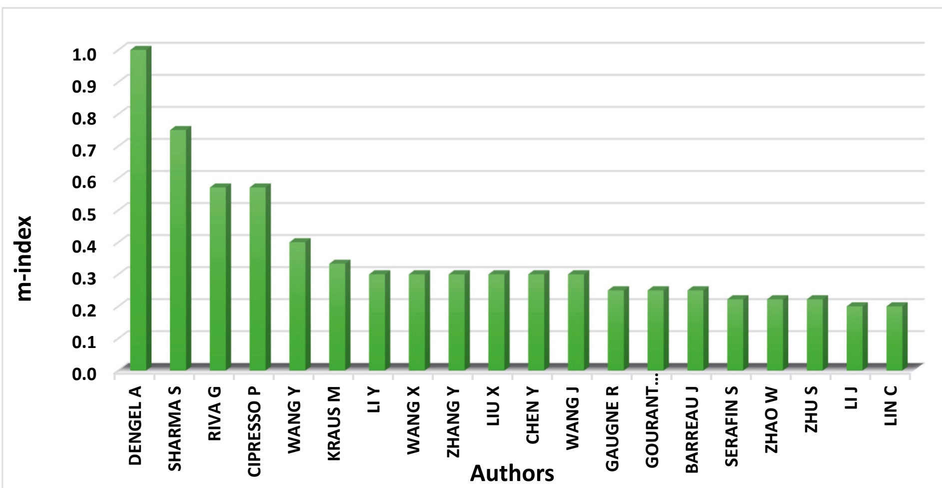


Figure 5. Top-20 authors' impact analysis within 10 years.

Our analysis revealed some top universities regarding institutions (authors' affiliations) and countries fronting VR in CS education. As shown in Figure 6, some of these universities, to name a few, were the University of Southern California, USA; Aalborg University, Denmark; and University of Rennes, France.

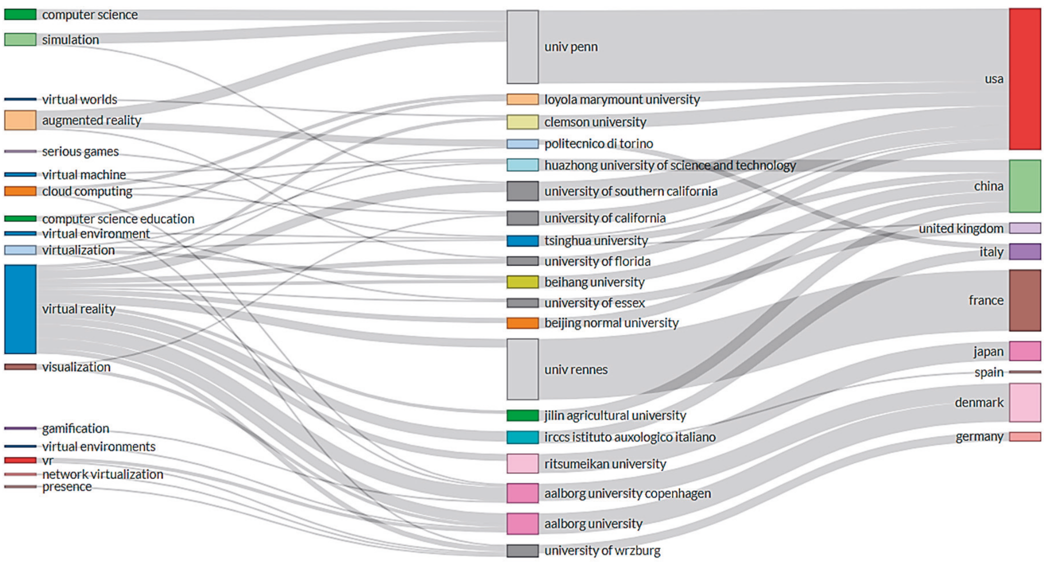


Figure 6. Three-field plot of active institutions and countries publishing articles related to VR in CS education between 2011 and 2020.

Figure 6 shows the USA as the most productive country in terms of publishing articles related to VR in CS education in countries. From the European continent, France, Denmark, Italy, the UK, Germany, and Spain made significant contributions. Only China and Japan made contributions regarding VR in CS education from the Asian continent.

3.1.3. Keywords Co-Occurrence Patterns of Studies on the Use of VR in Computer Science Education

A keywords co-occurrence pattern (KCP) focuses on understanding the knowledge components and knowledge structure of a scientific field by examining the links between keywords in the published articles within the same area [4].

Figure 7 focuses on keyword co-occurrence patterns of studies on the use of VR in computer science education. As observed in Figure 7, the root keyword in the field remains "virtual reality". Other keywords that are frequently used by articles on VR in CS education are shown in red color. For instance, we notice keywords such as gamification, simulation, higher education, mixed reality, serious games, and more. In addition, as expected, keywords that define the characteristics of virtual reality technology were seen to be strongly connected to the root keyword. For example, we observe a thick line connecting keywords such as immersion, interaction, and presence, to the root keyword "virtual reality". Moreover, virtualization, cloud computing, and virtual machine are keywords that show a strong connection. Other keywords that show a close relationship to virtual reality include augmented reality and computer science.

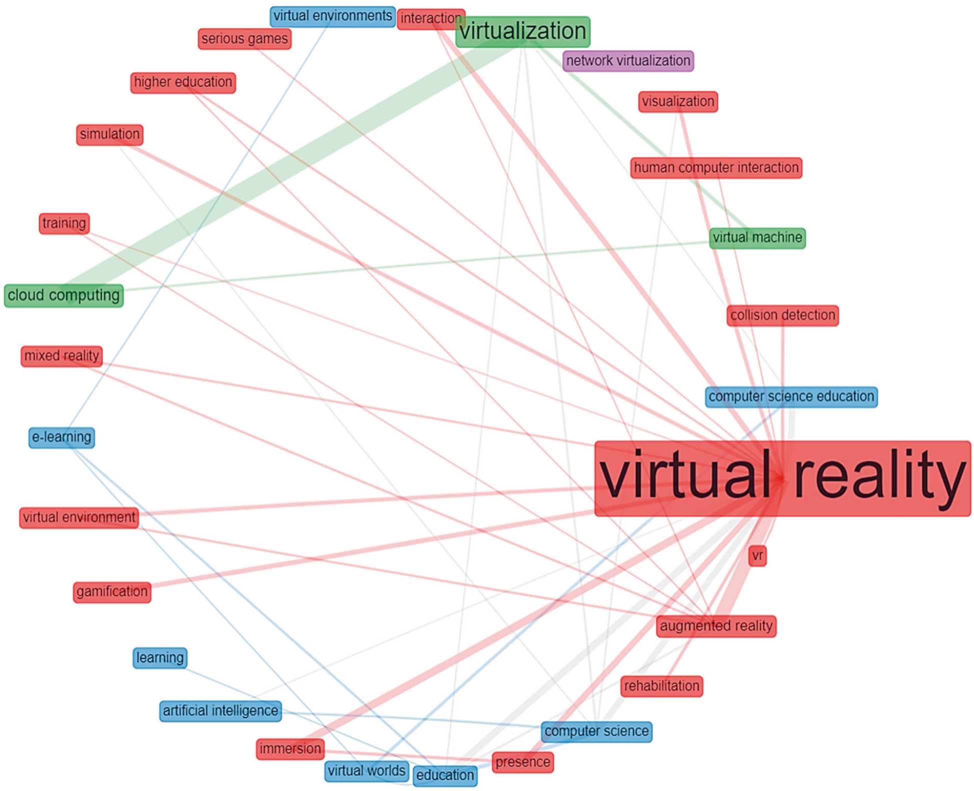


Figure 7. Co-occurrence patterns of authors' keywords in articles on VR in CS education between 2011 and 2020.

Furthermore, Figure 8 presents a visualization of frequently used keywords in VR for CS education. It is clear from the size of the nodes that other related terms used for virtual reality, for example, "virtualization" and "virtual environment" were found to be highly connected to "computer science" and "education". In addition, some pedagogical concepts for teaching and learning, such as games, gamification, collaborative learning, and immersive learning, are visible in the network. Figure 8 also shows clustering of concepts where terms such as virtualization, virtual environment, computer science, and education form clusters depicted with different colors.

One way to examine how VR application has influenced CS education is to analyze trending topics over the period considered in this study. Figure 9 presents the trending topics or approaches scholars have explored to provide VR intervention for CS education.

This study analyzed the authors' keywords to determine what research hotspot in terms of topics and approaches have been explored by VR applications in CS education in the last decade. This analysis was conducted through the word cloud of authors' keywords, which gives a pointer to what has been the scholars' interest. This analysis also provides insight regarding the future outlook of VR interventions in CS education. Figure 9 delineates that virtualization, cloud computing, the virtual world, and virtual machine dominate VR studies in CS education between the years 2011 and 2015. In addition, slightly different changes were observed where keywords such as computer science education, serious games, and higher education emerged among the trending topics between 2015 and 2017.

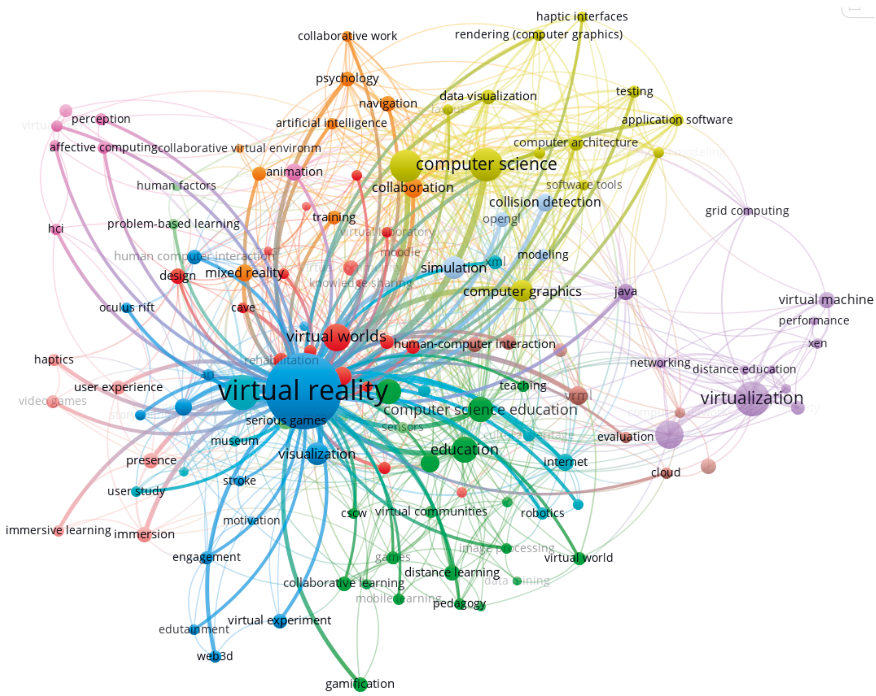
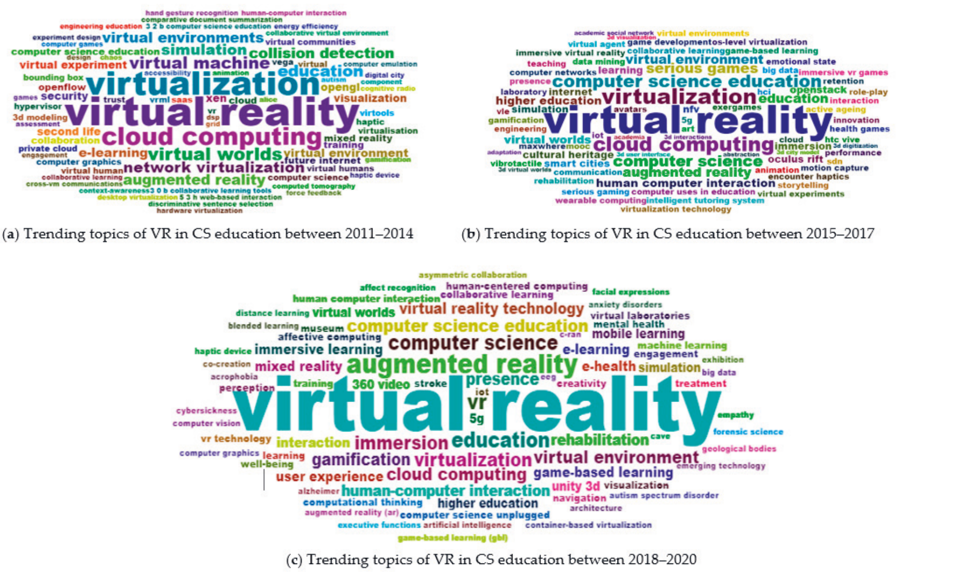


Figure 8. Visualized authors' keywords co-occurrence analysis of articles on VR in CS education: these are among the highest number of repetitive keywords within the field.



(**a**) Trending topics of VR in CS education between 2011–2014

(**b**) Trending topics of VR in CS education between 2015–2017

(c) Trending topics of VR in CS education between 2018–2020

Figure 9. (**a**–**c**) Word cloud showing the trending topics of VR in CS education in terms of authors keywords. While (**a**) shows the trending topics between 2011 and 2015, (**b**) presents the topics between 2015 and 2017, and (**c**) depicts the trending topics on VR in CS education between 2018 and 2020.

Furthermore, it was observed that between the years 2018 and 2020, new keywords such as augmented reality, immersion, presence, gamification, game-based learning, and human–computer interaction were added to the trending topics. Therefore, topics such as immersion, presence, human–computer interaction, gamification, and game-based learning dominate the list of research hotspots in recent times. This finding suggests that one of the most appreciated learning and teaching approaches used by studies on VR application in CS education is game-based learning.

3.2. Findings from Content Analysis

This section presents the content analysis findings to address some of the research questions (RQ4 to RQ7). Moreover, an overview of the data analyzed in this section is presented as an Appendix A. In the Appendix A, information regarding the study focus and outcome are highlighted to showcase how the selected articles have employed VR in CS education.

3.2.1. Trends of the Research Methodology Employed in Articles on the Use of VR in Computer Science Education

According to Figure 10, 47% of the articles used a quantitative design approach, 16% used a qualitative design, 3% used mixed design, and 12% utilized a design and development research approach. In comparison, others may include review/meta-analysis research accounts for 5%.

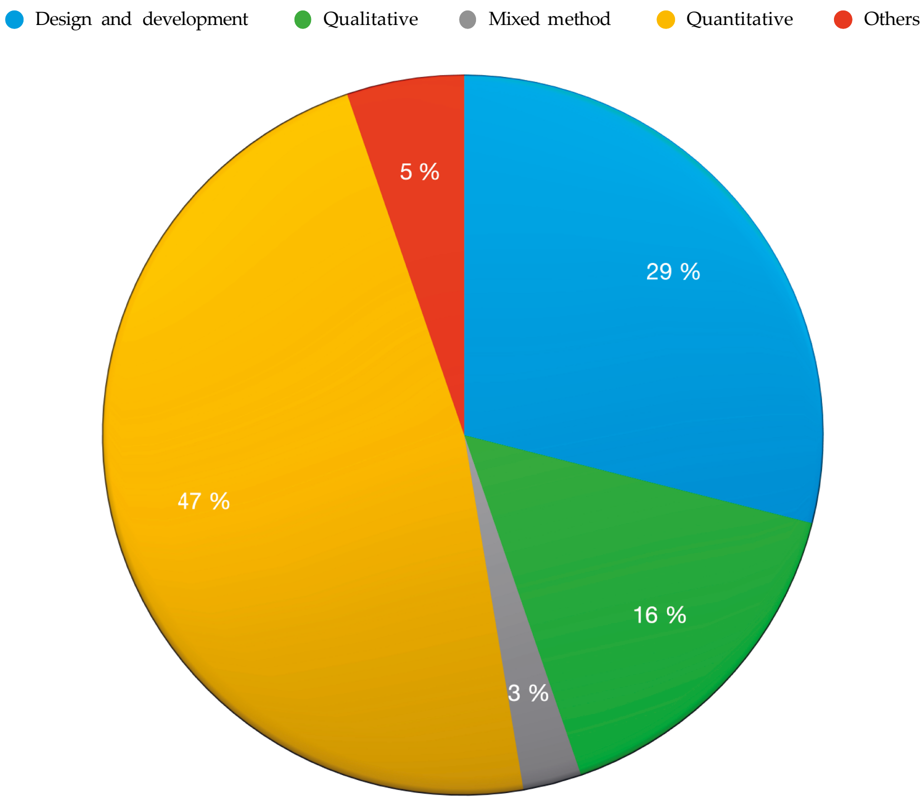


Figure 10. Frequency of research methods in articles on VR in CS education between 2011 and 2020.

Figure 11 revealed the research method trends related to VR in CS education in the past 10 years. The use of quantitative methods increased in 2018 and declined from 2019 to 2020. The next prominent method utilized is the design research method used in 2011 and in 2014, and witnessed an increase in 2020. While mixed methods are almost inexistent, qualitative and other methods showed no significant distribution variations over time. Review and meta-analysis began to be used in 2019 as the quantitative design was found to be the most used research method over the years.

Figure 11. Trends of research methods in articles on VR in CS in the past 10 years.

3.2.2. The Most Preferred Data Collection Tools and Sampling Methods in Articles on the Use of VR in Computer Science Education

Data collection tools and sampling methods in research conducted on VR in CS education show that the questionnaire (46%) remains the most used tool. However, quite a number of studies (23%) either did not conduct evaluation or did not specify what method of data collection was used.

As shown in Figure 12, the use of interviews (13%) is still growing as fewer studies have been seen to use the method.

3.2.3. Sample Populations and Sample Sizes in Articles on the Use of VR in Computer Science Education

According to Figure 13, the most commonly used sample size in articles published between 2011 and 2020 fell between 11–20 participants. Closely followed were 1–10 persons and 51–100 people. Although other studies utilized samples between 21–50 and 101–200 respondents, a few studies did not specify the sample size they used.

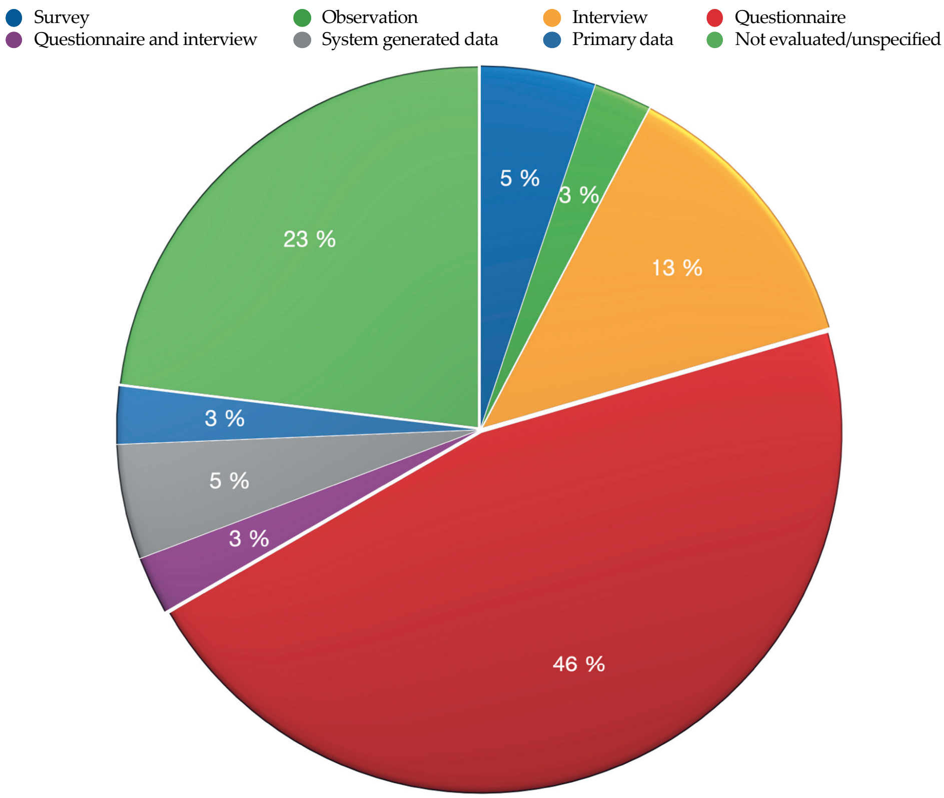


Figure 12. Data collection tools and sampling methods of articles on the use of VR in CS education.

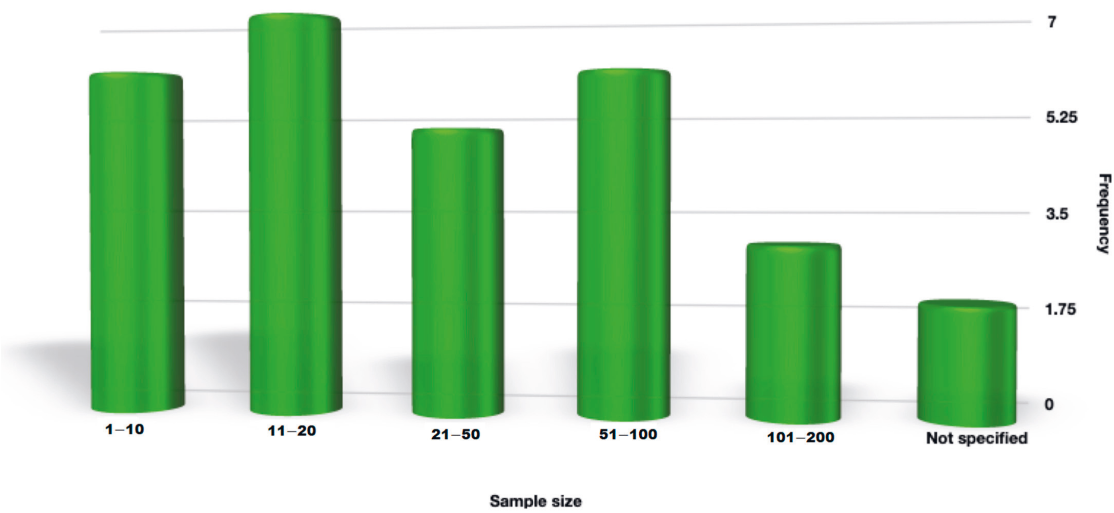


Figure 13. Frequency of use of sample sizes in articles.

3.2.4. Most Preferred Data Analysis Methods in Articles on the Use of VR in Computer Science Education

The findings show that most studies were performed using descriptive analysis regarding the most preferred data analysis conducted in studies focused on VR in CS education.

Other preferred analysis methods, as shown in Figure 14, are meta-analysis and content analysis. Moreover, some studies adopted a theoretical approach while some other studies did not conduct any form of research, and therefore we categorized these types of studies as "others/not specified".

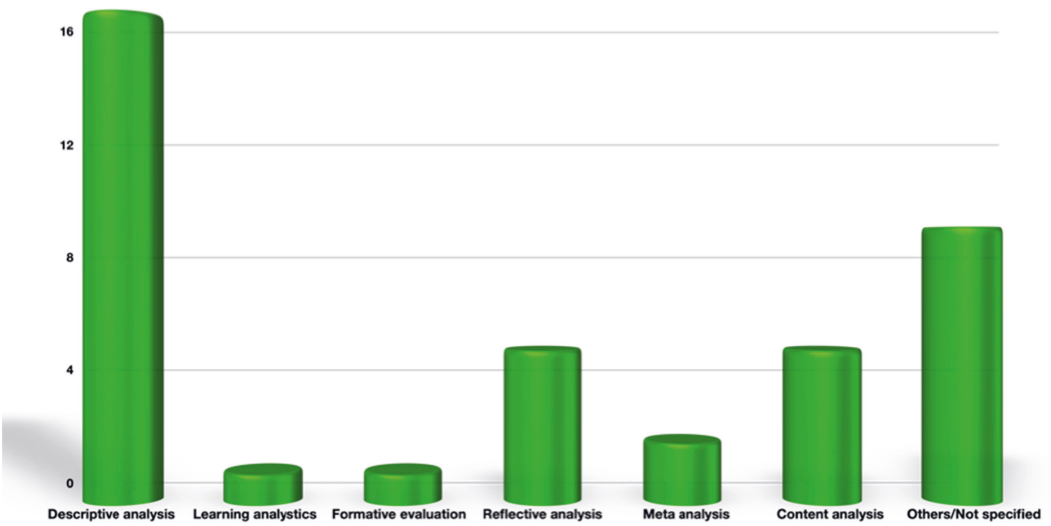


Figure 14. Most preferred data analysis method between 2011 and 2020.

4. Discussion

The bibliometric method's potential is seen by earlier research [4]. It was opined that bibliometric study advances complement meta-analysis and qualitative research for the scientific evaluation of literature. This study delved into VR's role in CS education to provide a deeper understanding of the evolution of research conducted in this field and anticipate the future direction on the basis of the analysis of the co-occurrence pattern of keywords used in studies conducted in the last 10 years. The study contributes to knowledge by presenting valuable findings that can boost the morale of prolific scholars who have been contributing to this field and researchers and practicing managers who may be starting to research into VR for CS education. This current study obtained its bibliometric and content analysis data from the Web of Science and Scopus databases.

The bibliometric analysis of articles related to the use of VR in CS education, together with the methodological research trends over the last 10 years, was revealed. Bibliometric analysis results showed that the year 2011 was the highest in article production (148 articles). This result was closely followed by the year 2018 with 135 articles. This finding implies that between 2012 and 2017, articles related to VR in CS education dwindled. Regarding the authors' production over time, Li Y. had the highest number of articles produced in the field, which is not surprising as the author consistently published in 2011–2012, 2015, and 2017–2020.

Moreover, we analyzed studies' impact by investigating the number of citations obtained by authors within 10 years. The analysis was focused on the m-index of each author. Considering the 10 years duration in this study, we calculated the m-index by dividing the total number of citations by the total number of years authors have been publishing. For example, Dengel A. emerged as the most impactful author because this author had produced one paper per year for only two years. This means that Dengel's impact analysis was computed on the basis of the output of these two years. However, it was surprising to discover that Li Y., who had the highest number of articles produced over the years, was not as impactful as Dengel A., who had a limited number of articles published within just two years. Earlier studies have examined intrinsic factors affecting the number of citations of articles [22,23]; however, some indicators are not directly related to the quality or content of articles' extrinsic factors [24]. The previous finding reveals that price index, number of references, keywords, and length of studies are essential explanatory factors [24]. It can be concluded that it is likely that Li's articles are easily accessible to researchers via open access medium. The relevancy of their topic or even the quality of their paper in terms of content and presentation may account for the citations and rapid impact.

Regarding the institutions and countries contributing to VR in CS education, the results further showed that the University of Southern California, USA; Aalborg University, Denmark; and the University of Rennes, France, remain the top universities in terms of publishing VR in CS education articles. On the other hand, the USA emerged as the most productive country. However, other countries from Europe (France, Denmark, Italy, the UK, and Germany) and Asia (China) are making a significant contribution towards advancing CS education using VR technology. The co-occurrence pattern of authors' keywords revealed that VR characteristics are leveraged for CS education. For example, immersion, presence, interaction, and gamification are being explored in advancing CS education [1,16,18]. Moreover, these keywords also form the research hotspots in VR, primarily to support learning. Therefore, this study anticipates that VR in CS education would continue to be researched within the scope of these keywords [14].

The content analysis results showed that quantitative studies (47%) dominate the studies in terms of research methodology. The reason for quantitative method preference may be due to the simplified way of presenting quantitative research, as well as less time and effort required to conduct and analyze quantitative data [25]. It might also be the case that the generalization and replicability that the quantitative approach provides accounts for its dominance in the studies. The percentage for the use of mixed methods studies was meager, reflecting that the use of mixed approach studies presents methodological difficulties and challenges [12]. It is safe to conclude that only a few studies consider the potential of mixed-method research, which adds rigor and validity to research through triangulation and convergence of multiple and different sources of information [26,27]. Moreover, few qualitative studies have been conducted in the last 10 years. This may have been due to the rigor and non-use of numbers, making it difficult to simplify findings and observations [25]. On the contrary, Johnson and Christensen [28] assert that reliance on collecting non-numerical primary data such as words and pictures makes qualitative research well-suited for providing factual and descriptive information.

Regarding the frequency of the sampling size utilized over the years, the most used sample sizes were 11–20. We were surprised to find out that most published articles on VR in CS education were evaluated with about 11 to 20 participants. Since the research method's preference was quantitative research, we expected that many studies would have used more participants to arrive at a generalized outcome. Although studies that used 51–100 sample sizes were also seen in the result, one could have thought that 20 participants may be too small for a quantitative study. According to Faber and Fonseca [29], very small samples undermine the internal and external validity, while huge samples tend to transform minor differences into statistically significant differences.

Our findings revealed that the questionnaire is the most used data collection tool, while descriptive analysis remains the preferred data analysis method. One way to reflect on this result is that the questionnaire seems more straightforward, quicker, and cost-effective to collect data from participants. Moreover, the preference for descriptive analysis may be used to simplify data efficiently [30]. The researcher may have adopted this data analysis method to reduce the time and effort required to format and present beneficial, easily interpretable results to practitioners, policymakers, and other researchers to understand a phenomenon better.

5. Conclusions

This study provides a comprehensive view of scientific papers on VR in CS education published in peer-reviewed journals and conferences between 2011 and 2020. Two main approaches were explored to answer the research questions presented in this study. First, the bibliometric analysis answered the questions regarding the article production growth in the field within a decade, prolific scholars and their affiliations publishing to advance VR in CS education, and research hotspots in the field may guide scholar's future research focus. Second, content analysis of articles that met the inclusion criteria for this study was analyzed to provide a methodological overview of studies conducted on VR in CS education. Several findings were presented in this study. These findings show that VR research for CS education has fared well; however, some of the years (between 2013 and 2016) witnessed low article production. The study also revealed the prolific scholars and authors' impact analysis in this field and provided insightful information regarding research hotspots by analyzing the authors' keywords co-occurrence.

Regarding the scientific methodology and data sampling technique used by studies on VR in CS education, the most preferred is the quantitative method. At the same time, the questionnaire was the most used data collection technique. Moreover, descriptive analysis was mainly used to analyze data in studies on VR in CS education.

This study witnessed a limitation regarding the content analysis. It would be interesting to see the educational context where VR technology is being used and the learning contents deployed in the VR application for CS education. Nonetheless, this study contributes to knowledge in significant ways. The study revealed that pedagogical approaches such as game-based learning and gamification were explored for VR education in CS education. The findings from this study can provide insight into how VR technology research has progressed in a decade. Moreover, the result can be generalized since this study could obtain relevant data from two databases (WoS and Scopus) to conduct its analysis. The process for merging these data is another contribution as scholars interested in running a similar study would find this helpful study. Our future study would address the limitations by providing answers regarding the pedagogy, content, and context of studies on VR in CS education.

By implication, we conclude that findings from this study suggest that even though scholars are leveraging VR to advance teaching and learning in the field of CS, more effort needs to be made, especially from continents, countries, and institutions that were not reported among the top-20 list revealed in this study. In addition, a more rigorous methodological approach needs to be employed in a future study to provide more evident-based research output. For example, our study revealed only a few studies that used a mixed-methods approach, which has been more rigorous in terms of quality of scientific research.

Author Contributions: Conceptualization, F.J.A., I.T.S.; Data curation, F.J.A., I.T.S.; Formal analysis, F.J.A., I.T.S., S.S.O. and J.S.; Investigation, F.J.A., I.T.S. and S.S.O.; Methodology, F.J.A., I.T.S., S.S.O. and J.S.; Project administration, F.J.A.; Resources, F.J.A., I.T.S.; Software, F.J.A., I.T.S., S.S.O. and J.S.; Supervision, S.S.O. and J.S.; Validation, S.S.O. and J.S.; Writing—original draft, F.J.A. and I.T.S.; Writing—review & editing, F.J.A., I.T.S., S.S.O. and J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Published articles contained in the content analysis of VR for CS education (2011–2020).

Authors	Aim of the Study	Results of the Study
Nguyen et al. [31]	Virtual reality (VR) programming environment called VRASP was developed allow students to produce an avatar (agent) in a virtual world that is able to answer questions in spoken natural language.	Findings from the study show that students were able to communicate with the environment intuitively with an accuracy of 78%.
Srimadhaven et al. [32]	The study focused on conducting an experiment with the virtual reality mobile app in order to assess the cognitive level of the students in a Python course.	The authors anticipated that findings can be useful to higher education students and enhance the performance of all levels of learners.
Bouali et al. [33]	This study presented a VR-based learning game to support the teaching and learning of object-oriented programming (OOP) concepts in computing education.	The authors envisaged that the designed game would spark interest for learning CS programming concepts such as IF condition, Arrays, and Loops.
Dengel [34]	This study demonstrated how metaphorical representations in VR can enhance the understanding of theoretical computer science concepts by using the Treasure Hunt game.	The study anticipated measuring students' cognition, presence, usability, and satisfaction in their future study.
Bolivar et al. [35]	This study presented an immersion 3D environment in the form of a video game. The environment offers the player the opportunity to explore basic CS concepts without removing any of the entertaining aspects of games.	The authors anticipated a positive impact of the framework when their future research is completed.
Parmar et al. [36]	This authors developed a virtual reality tool—VEnvI—to support CS students in learning about the fundamental of CS.	The study presented several cases and sample projects developed to assist teachers in their classes.
Kerdvibulvech [37]	This study proposed a virtual environment framework for human–computer interaction.	The author envisaged that this approach could provide significant ducational values.
Rodger et al. [38]	The authors have developed curriculum materials for several disciplines both for student and teacher use. The curriculum materials include tutorials, sample projects, and challenges for teaching CS topics.	Demonstration and evaluation of the tool was expected to produce useful outcome.
Vallance [39]	This study aimed to set a medium of collaboration within a 3D virtual world.	This study was still a work in progress, and hence a concrete result was not presented.
Arrington et al. [40]	This study designed and implemented Dr. Chestr, a virtual human in a virtual environment game aimed at supporting the understanding and retention of introductory programming cources.	The study measured students' cognition, presence, usability, and satisfaction and found that students enjoyed the experience and were successfully engaged the virtual world.
Vanderdonckt and Vatavu [41]	This study present a VR application where the user, a psychologist, controls a virtual puppet (a cartoon-like character in VR).	The study found that when receiving lectures in a virtual environment by a teacher, the child was calm, focused, and capable of working on his assignments without showing any disruptive behaviors.
Parmar et al. [42]	The authors developed a VR tool—VEnvI—to support CS students in learning about the fundamental CS concepts such as sequences, loops, variables, conditionals, and functions.	Participants who tested the VR tool agreed that the visual aspect improved the overall learning experience.

Table A1. *Cont.*

Authors	Aim of the Study	Results of the Study
Adjorlu and Serafin [43]	This study investigated the feasibility of using VR to reduce disruptive classroom behavior of a child diagnosed with autism spectrum disorder (ASD).	The study provided guidelines to educators and instructional designers who wish to offer interactive and engaging learning activities to their students.
Berns et al. [44]	A VR educational platform MYR was built to spark student interest in computer science by allowing them to write code that generates three-dimensional, animated scenes in virtual reality environment. The goal of the project was to gain insight into computing students' success, motivation, and confidence in learning computing.	Evaluation with CS students shows that MYR is hard for CS students to provide clear 3D representations for programming concepts; however, the study was able to derive some common figures.
Christopoulos et al. [45]	Authors investigared what effect instructional design decisions have on motivation and engagement of students learning in virtual and physical world.	Evaluation of this tool suggests that users' experience is enhanced through the 3D animation.
Ortega et al. [46]	The study developed a 3D virtual programming language to provide an interactive tool for beginners and intermediate students to learn programming concepts.	The study reported that the method creates fun and effective means of interdisciplinary study.
Sanna et al. [47]	This study proposed a virtual 3D tool (touchless interface) to support people without any prior knowledge in code writing to promote user friendliness and usability experience.	Feedback from the workshop participants generally shows that they had a good experience.
Cleary et al. [48]	This study explored a style of teaching youths how to write computer program using reactive programming in a 3D virtual environment.	The study tested educational virtual environments (EVEs) with pre- and post-test and found to be significantly effective.
Domik et al. [49]	The authors created "Move the World" workshop in a summer camp to increase high school juniors' interest in computer science by leveraging math and virtual worlds.	Overall comments from participants of the workshop revealed that learning in the virtual world is appealing and inspiring.
Dengel [50]	The study modeled three computer scienc topics- asymmetric encryption/decryption, and finite state machines in a 3 D immersive VR to teach these topics.	The study discusses students' preconceptions towards the inclusion of 3D virtual learning environments in the context of their studies and further elicit their thoughts related to the impact of the "hybrid" interactions
Koltai et al. [51]	This study used a VR game (Mazes) to teache CS concepts.	The study reported positive impact on computer science education by increasing engagement, knowledge acquisition, and self-directed learning.
Christopoulos et al. [52]	This authors developed a tool—FunPlogs application—to deply microlearning.	The study generally indicated that participants perceived a high joy of use while playing FunPlogs, which indicated that despite the simple game concept, complex matters as the while-loop could be transported to programming laymen.
Banic and Gamboa [53]	The study explored a summer course that uses visual design problem-based learning pedagogy with virtual environments as a strategy to teach computer science.	The study concluded that interactions in VR plays a crucial role in learner engagement.
Horst et al. [54]	This study introduced a VR puzzle mini-game for learning fundamental programming principles.	The study outcome shows that the proposed module helps students learn stacks and queues while being satisfactorily usable.

Table A1. *Cont.*

Authors	Aim of the Study	Results of the Study
Christopoulos and Conrad [55]	Authors examined the impact that the virtual reality learning process has on university students who study CS and have almost no experience in the use of virtual worlds.	Results show that the self-overlapping maze is experienced as freely walkable while the map is mostly understandable.
Stigall and Sharma [56]	This study designed a game theme-based instructional (GTI) module to teach undergraduate CS majors about stacks and queues.	The analysis of SEQ usability test shows good acceptance.
Serubugo et al. [57]	This study investigated how working with VR setups can be walkable in small physical spaces or included in non-HMD participants using self-overlapping maze	Analysis of the usability and likeability of the survey shows that students felt motivated and engaged in learning programming concepts.
Pilatásig et al. [58]	This study designed a VR tool to assist in training and rehabilitation of hands and wrist	The study reported that students gained cognitive thinking process and had a greater range of expressing sufficiently alternative to self-explanatory solutions.
Segura et al. [59]	This study designed a VR application (VR-OCKS) to teach basic programming concepts such as flow statements and conditional selections.	The initial evaluation of this tool shows that it enhanced creative thinking of young children.
Pellas and Vosinakis [60]	The authors explored a 3D simulation game to teach computational problem-solving.	Evaluation results demonstrated positive student perceptions about the use of gaming instructional modules to advance student learning and understanding of the concepts.
Stigall and Sharma [61]	This study designed and developed two gaming modules for teaching CS students object-oriented programming (OOP) and binary search.	Result analysis suggests that participants showed similar connectedness in affiliative tour and competitive design.
Sharma and Ossuetta [62]	The authors developed virtual reality instructional (VRI) modules for teaching loops and arrays that can provide a better understanding of the concept.	The study measured participants' intentions toward majoring in a computing discipline, attitudes toward computing, and overall satisfaction with the camp, and showed positive indication.
Ijaz et al. [63]	This study proposed a VR exergaming platform that combines a recumbent tricycle and real-world panoramic images where the player can navigate real locations in a safe virtual environment	This study argued that comparative studies are a useful method for analyzing benefits of different approaches to controlling virtual agents.
Hulsey et al. [64]	This study reported the experience of a summer camp that introduced computing concepts to middle school girls in the context of an online, multiplayer, virtual world.	This study demonstrated that familiarity may reduce working memory load and increase children's spatial memory capacity for acquiring sequential temporal–spatial information from virtual displays.
Gemrot et al. [65]	This study presents results of comparing the usability of an academic technique designed for programming intelligent agents' behavior with the usability of an unaltered classical programming language.	Outcome of the experiment with CodeSpells shows that students were able to understand and write basic Java code after only 8 h of playing the game.
Korallo et al. [66]	This study examined the potential use of virtual environment in general computer knowledge in virtual environment.	Outcome of the study provide overview of the two reviewed approaches for implementing VR gestures, which may guide experts.

References

1. Oyelere, S.S.; Bouali, N.; Kaliisa, R.; Obaido, G.; Yunusa, A.A.; Jimoh, E.R. Exploring the trends of educational virtual reality games: A systematic review of empirical studies. *Smart Learn. Environ.* **2020**, *7*, 1–22. [CrossRef]
2. Carruth, D.W. Virtual reality for education and workforce training. In Proceedings of the 2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA), Stary Smokovec, Slovakia, 26–27 October 2017; pp. 1–6.
3. Pirker, J.; Dengel, A.; Holly, M.; Safikhani, S. Virtual Reality in Computer Science Education: A Systematic Review. In Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology; Association for Computing Machinery (ACM), Tokyo, Japan, 1–4 November 2020; pp. 1–8.
4. Agbo, F.J.; Oyelere, S.S.; Suhonen, J.; Tukiainen, M. Scientific production and thematic breakthroughs in smart learning environments: A bibliometric analysis. *Smart Learn. Environ.* **2021**, *8*, 1–25. [CrossRef]
5. Agbo, F.J.; Oyelere, S.S.; Bouali, N. A UML approach for designing a VR-based smart learning environment for programming education. In Proceedings of the 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 21–24 October 2020; pp. 1–5.
6. Agbo, F.J.; Oyelere, S.S. Smart mobile learning environment for programming education in Nigeria: Adaptivity and context-aware features. In *Intelligent Computing-Proceedings of the Computing Conference*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1061–1077.
7. Agbo, F.J.; Oyelere, S.S.; Suhonen, J.; Tukiainen, M. Identifying potential design features of a smart learning environment for programming education in Nigeria. *Int. J. Learn. Technol.* **2019**, *14*, 331–354. [CrossRef]
8. Agbo, F.J.; Oyelere, S.S.; Suhonen, J.; Tukiainen, M. Smart learning environment for computing education: Readiness for implementation in Nigeria. In Proceedings of the EdMedia+ Innovate Learning, Association for the Advancement of Computing in Education (AACE), Amsterdam, The Netherlands, 24–28 June 2019; pp. 1382–1391.
9. Agbo, F.J.; Oyelere, S.S.; Suhonen, J.; Adewumi, S. A systematic review of computational thinking approach for programming education in higher education institutions. In Proceedings of the 19th Koli Calling International Conference on Computing Education Research, Koli, Finland, 21–24 November 2019; pp. 1–10.
10. Zupic, I.; Čater, T. Bibliometric methods in management and organisation. *Organ. Res. Methods* **2015**, *18*, 429–472. [CrossRef]
11. Arici, F.; Yildirim, P.; Caliklar, Ş.; Yilmaz, R.M. Research trends in the use of augmented reality in science education: Content and bibliometric mapping analysis. *Comput. Educ.* **2019**, *142*, 103647. [CrossRef]
12. Buchholtz, N. Planning and conducting mixed methods studies in mathematics educational research. In *Compendium for Early Career Researchers in Mathematics Education ICME-13 Monographs*; Kaiser, G., Presmeg, N., Eds.; Springer: Cham, Switzerland, 2019; pp. 131–152.
13. Tüysüz, C. The Effect of the Virtual Laboratory on Students'Achievement and Attitude in Chemistry. *Int. Online J. Educ. Sci.* **2010**, *2*, 37–53.
14. Cheng, K.; Tsai, C. Students' motivational beliefs and strategies, perceived immersion and attitudes towards science learning with immersive virtual reality: A partial least squares analysis. *Br. J. Educ. Technol.* **2020**, *51*, 2140–2159. [CrossRef]
15. Bogusevschi, D.; Muntean, C.; Muntean, G.M. Teaching and Learning Physics using 3D Virtual Learning Environment: A Case Study of Combined Virtual Reality and Virtual Laboratory in Secondary School. *J. Comput. Math. Sci. Teach.* **2020**, *39*, 5–18.
16. Huang, H.-M.; Liaw, S.-S. An Analysis of Learners' Intentions Toward Virtual Reality Learning Based on Constructivist and Technology Acceptance Approaches. *Int. Rev. Res. Open Distrib. Learn.* **2018**, *19*. [CrossRef]
17. Lorenzo, G.; Lorenzo-Lledó, A.; Lledó Carreres, A.; Pérez-Vázquez, E. Approach from a Bibliometric Perspective of the Educational Application of Virtual Reality in People with Autism Spectrum Disorder. *Educ. Knowl. Soc.* **2020**, *21*, 4–14.
18. Sobral, M.; Pestana, M.H. Virtual reality and dementia: A bibliometric analysis. *Eur. J. Psychiatry* **2020**, *34*, 120–131. [CrossRef]
19. Lai, N.Y.G.; Wong, K.H.; Yu, L.J.; Kang, H.S. Virtual Reality (VR) in Engineering Education and Training: A Bibliometric Analysis. In Proceedings of the 2020 2nd World Symposium on Software Engineering, San Francisco, CA, USA, 18–20 May 2020; pp. 161–165.
20. Enebechi, C.N.; Duffy, V.G. Virtual Reality and Artificial Intelligence in Mobile Computing and Applied Ergonomics: A Bibliometric and Content Analysis. In Proceedings of the Constructive Side-Channel Analysis and Secure Design, Lugano, Switzerland, 1–3 April 2020; pp. 334–345.
21. Aria, M.; Cuccurullo, C. Bibliometrix: An R-tool for comprehensive science mapping analysis. *J. Inf.* **2017**, *11*, 959–975. [CrossRef]
22. Chen, C. Predictive effects of structural variation on citation counts. *J. Am. Soc. Inf. Sci. Technol.* **2011**, *63*, 431–449. [CrossRef]
23. Didegah, F.; Thelwall, M. Determinants of research citation impact in nanoscience and nanotechnology. *J. Am. Soc. Inf. Sci. Technol.* **2013**, *64*, 1055–1064. [CrossRef]
24. Onodera, N.; Yoshikane, F. Factors affecting citation rates of research articles. *J. Assoc. Inf. Sci. Technol.* **2015**, *66*, 739–764. [CrossRef]
25. Eyisi, D. The Usefulness of Qualitative and Quantitative Approaches and Methods in Researching Problem-Solving Ability in Science Education Curriculum. *J. Educ. Pract.* **2016**, *7*, 91–100.
26. Jokonya, O. The significance of mixed methods research in information systems research. In Proceedings of the Midwest Association for Information Systems Conference (MWAIS), Springfield, IL, USA, 19–20 May 2016.
27. Regnault, A.; Willgoss, T.; Barbic, S. Towards the use of mixed methods inquiry as best practice in health outcomes research. *J. Patient-Rep. Outcomes* **2018**, *2*, 19. [CrossRef] [PubMed]

28. Johnson, R.B.; Christensen, L.B. *Educational Research: Quantitative, Qualitative, and Mixed Approaches*, 6th ed.; SAGE Publications: Thousand Oaks, CA, USA, 2017.
29. Faber, J.; Fonseca, L.M. How sample size influences research outcomes. *Dent. Press J. Orthod.* **2014**, *19*, 27–29. [CrossRef]
30. Loeb, S.; Dynarski, S.; McFarland, D.; Morris, P.; Reardon, S.; Reber, S. Descriptive Analysis in Education: A Guide for Researchers. *NCEE 2017-4023 Natl. Cent. Educ. Eval. Reg. Assist.* **2017**, 1–40.
31. Nguyen, V.T.; Zhang, Y.; Jung, K.; Xing, W.; Dang, T. VRASP: A Virtual Reality Environment for Learning Answer Set Programming. In *International Symposium on Practical Aspects of Declarative Languages*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 82–91.
32. Srimadhaven, T.; AV, C.J.; Harshith, N.; Priyaadharshini, M. Learning analytics: Virtual reality for programming course in higher education. *Procedia Comput. Sci.* **2020**, *172*, 433–437.
33. Bouali, N.; Nygren, E.; Oyelere, S.S.; Suhonen, J.; Cavalli-Sforza, V. Imikode: A VR Game to Introduce OOP Concepts. In Proceedings of the 19th Koli Calling International Conference on Computing Education Research 2019, Koli, Finland, 21–24 November 2019; pp. 1–2.
34. Dengel, A. Seeking the treasures of theoretical computer science education: Towards educational virtual reality for the visualization of finite state machines. In Proceedings of the 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Wollong, Australia, 4–7 December 2018; pp. 1107–1112.
35. Bolivar, S.; Perez, D.; Carrasquillo, A.; Williams, A.S.; Rishe, N.D.; Ortega, F.R. 3D Interaction for Computer Science Educational VR Game. In *International Conference on Human-Computer Interaction*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 408–419.
36. Parmar, D.; Babu, S.V.; Lin, L.; Jörg, S.; D'Souza, N.; Leonard, A.E.; Daily, S.B. Can embodied interaction and virtual peer customization in a virtual programming environment enhance computational thinking? In Proceedings of the 2016 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), Atlanta, GA, USA, 11–13 August 2016; pp. 1–2.
37. Kerdvibulvech, C. Vision and virtual-based human computer interaction applications for a new digital media visualization. In Proceedings of the WSCG 2015 Conference on Computer Graphics, Visualization and Computer Vision, Pilsen near Prague, Czech Republic, 8–12 June 2015; pp. 247–254.
38. Rodger, S.H.; Brown, D.; Hoyle, M.; MacDonald, D.; Marion, M.; Onstwedder, E.; Ward, E. Weaving computing into all middle school disciplines. In Proceedings of the 2014 Conference on Innovation Technology in Computer Science Education 2014, Uppsala, Sweden, 21–25 June 2014; pp. 207–212.
39. Vallance, M. The affect of collaboratively programming robots in a 3D virtual simulation. In Proceedings of the 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Tokyo, Japan, 3–6 March 2013; pp. 245–246.
40. Arrington, C., Jr.; Wilson, D.M.; Lehmann, L. Improving performance and retention in computer science courses using a virtual game show. In Proceedings of the 49th Annual Southeast Regional Conference 2011, Kennesaw, GA, USA, 24–26 March 2011; pp. 320–321.
41. Vanderdonckt, J.; Vatavu, R.D. A pen user interface for controlling a virtual puppet. In Proceedings of the 12th ACM SIGCHI Symposium on Engineering Interactive Computing Systems 2020, Sophia Antipolis, France, 23–26 June 2020; pp. 1–6.
42. Parmar, D.; Isaac, J.; Babu, S.V.; D'Souza, N.; Leonard, A.E.; Jörg, S.; Daily, S.B. Programming moves: Design and evaluation of applying embodied interaction in virtual environments to enhance computational thinking in middle school students. In Proceedings of the 2016 IEEE Virtual Reality (VR), Greenville, SC, USA, 19–23 March 2016; pp. 131–140.
43. Adjorlu, A.; Serafin, S. Head-mounted display-based virtual reality as a tool to reduce disruptive behavior in a student diagnosed with autism spectrum disorder. In *Interactivity, Game Creation, Design, Learning, and Innovation*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 739–748.
44. Berns, C.; Chin, G.; Savitz, J.; Kiesling, J.; Martin, F. Myr: A web-based platform for teaching coding using VR. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education 2019, Minneapolis, MN, USA, 27 February–2 March 2019; pp. 77–83.
45. Christopoulos, A.; Conrad, M.; Shukla, M. Learner experience in hybrid virtual worlds: Interacting with pedagogical agents. In Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019), Heraklion, Crete, Greece, 2–4 May 2019; pp. 488–495.
46. Ortega, F.R.; Bolivar, S.; Bernal, J.; Galvan, A.; Tarre, K.; Rishe, N.; Barreto, A. Towards a 3D virtual programming language to increase the number of women in computer science education. In Proceedings of the 2017 IEEE Virtual Reality Workshop on K-12 Embodied Learning through Virtual Augmented Reality (KELVAR), Los Angeles, CA, USA, 19 March 2017; pp. 1–6.
47. Sanna, A.; Lamberti, F.; Bazzano, F.; Maggio, L. Developing touch-less interfaces to interact with 3D contents in public exhibitions. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 293–303.
48. Cleary, A.; Vandenbergh, L.; Peterson, J. Reactive game engine programming for stem outreach. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education 2015, Kansas City, MO, USA, 4–7 March 2015; pp. 628–632.
49. Domik, G.; Arens, S.; Stilow, P.; Friedrich, H. Helping High Schoolers Move the (Virtual) World. *IEEE Comput. Graph. Appl.* **2013**, *33*, 70–74. [CrossRef]
50. Dengel, A. How Important is Immersion for Learning in Computer Science Replugged Games? In Proceedings of the 51st ACM Technical Symposium on Computer Science Education 2020, New York, NY, USA, 11–14 March 2014; pp. 1165–1171.

51. Koltai, B.G.; Husted, J.E.; Vangsted, R.; Mikkelsen, T.N.; Kraus, M. Procedurally Generated Self Overlapping Mazes in Virtual Reality. In *Interactivity, Game Creation, Design, Learning, and Innovation*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 229–243.
52. Christopoulos, A.; Conrad, M.; Shukla, M. What Does the Pedagogical Agent Say? In Proceedings of the 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Piraeus, Greece, 15–17 July 2019; pp. 1–7.
53. Banic, A.; Gamboa, R. Visual Design Problem-based Learning in a Virtual Environment Improves Computational Thinking and Programming Knowledge. In Proceedings of the 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 23–27 March 2019; pp. 1588–1593.
54. Horst, R.; Naraghi-Taghi-Off, R.; Diez, S.; Uhmann, T.; Müller, A.; Dörner, R. FunPlogs–A Serious Puzzle Mini-game for Learning Fundamental Programming Principles Using Visual Scripting. In *International Symposium on Visual Computing*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 494–504.
55. Christopoulos, A.; Conrad, M.; Shukla, M. Increasing student engagement through virtual interactions: How? *Virtual Real.* **2018**, *22*, 353–369. [CrossRef]
56. Stigall, J.; Sharma, S. Usability and Learning Effectiveness of Game-Themed Instructional (GTI) Module for Teaching Stacks and Queues. *SoutheastCon* **2018**, 1–6. [CrossRef]
57. Serubugo, S.; Škantárová, D.; Evers, N.; Kraus, M. Self-overlapping Maze and Map Design for Asymmetric Collaboration in Room-Scale Virtual Reality for Public Spaces. In *Interactivity, Game Creation, Design, Learning, and Innovation*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 194–203.
58. Pilatasig, M.; Tigse, J.; Chuquitarco, A.; Pilatasig, P.; Pruna, E.; Acurio, A.; Buele, J.; Escobar, I. Interactive System for Hands and Wrist Rehabilitation. In *Advances in Intelligent Systems and Computing*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 593–601.
59. Segura, R.J.; Del Pino, F.J.; Ogáyar, C.J.; Rueda, A.J. VR-OCKS: A virtual reality game for learning the basic concepts of programming. *Comput. Appl. Eng. Educ.* **2020**, *28*, 31–41. [CrossRef]
60. Pellas, N.; Vosinakis, S. Learning to Think and Practice Computationally via a 3D Simulation Game. In *Interactive Mobile Communication, Technologies and Learning*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 550–562.
61. Stigall, J.; Sharma, S. Virtual reality instructional modules for introductory programming courses. In Proceedings of the 2017 IEEE Integrated STEM Education Conference (ISEC), Princeton, NJ, USA, 11 March 2017; pp. 34–42.
62. Sharma, S.; Ossuetta, E. Virtual Reality Instructional Modules in Education Based on Gaming Metaphor. *Electron. Imaging* **2017**, *2017*, 11–18. [CrossRef]
63. Ijaz, K.; Wang, Y.; Milne, D.; Calvo, R.A. VR-Rides: Interactive VR Games for Health. In *Joint International Conference on Serious Games*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; Volume 9894, pp. 289–292.
64. Hulsey, C.; Pence, T.B.; Hodges, L.F. Camp CyberGirls: Using a virtual world to introduce computing concepts to middle school girls. In Proceedings of the 45th ACM Technical Symposium on Computer Science Education, Atlanta, GA, USA, 5 March 2014; pp. 331–336.
65. Gemrot, J.; Hlavka, Z.; Brom, C. Does High-Level Behavior Specification Tool Make Production of Virtual Agent Behaviors Better? In *International AAMAS Workshop on Cognitive Agents for Virtual Environments*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7764, pp. 161–183.
66. Korallo, L.; Foreman, N.; Boyd-Davis, S.; Moar, M.; Coulson, M. Do challenge, task experience or computer familiarity influence the learning of historical chronology from virtual environments in 8–9 year old children? *Comput. Educ.* **2012**, *58*, 1106–1116. [CrossRef]

ARTICLE V

Agbo, F. J., Oyelere, S. S., & Bouali, N.
“A UML approach for designing a VR-based smart learning environment for programming education,” *IEEE Frontiers in Education Conference (FIE)*, pp. 1–5, 2020. https://ieeexplore.ieee.org/abstract/document/9273956

A UML approach for designing a VR-based smart learning environment for programming education

Friday Joseph Agbo
School of Computing
University of Eastern Finland
Joensuu, FInland
fridaya@uef.fi

Solomon Sunday Oyelere
School of Computing
University of Eastern Finland
Joenssu, Finland
solomon.oyelere@uef.fi

Nacir Bouali
School of Computing
University of Eastern Finland
Joenssu, Finland
nacirb@uef.fi

***Abstract*— This study is a work in progress that aims to design and implement a smart learning environment based on virtual reality technology to aid the teaching and learning of programming concepts. The paper followed the approach of designing and modelling of requirement specification for the intended smart platform. This modelling approach is desirable in satisfying activities that engender the design and prototyping of the smart learning environment based on the design science research method. The study discusses the proposed architecture of the system, modelled the system with UML, presents a scenario-based model for teaching and learning of programming concepts, and connect the outcome to the future research.**

Keywords—Smart learning environment, UML, programming, computing education, virtual reality, VR

I. Introduction

Programming education has become a mainstream subject taught in schools and higher education institutions [1] [2] [3] [4]. Teaching and learning of programming concepts have attracted scholars' interest in the recent past [5]. Many studies try to investigate students' performance in programming courses [6] [7]. These studies show that understanding programming concepts such as functions, loops, and recursion have become difficult for students of computer science [8] [9]. It is even more difficult for novices with little or no programming background because of the abstract nature of these programming concepts [10]. Anyango & Suleman [11] investigated programming topics that lecturers consider most difficult to comprehend by the students. Their study revealed that recursion is the most perceived difficult topic for novices pursuing computer science degree. Efforts to teach these programming concepts for better understanding have been made. For example, Henderson & Romero [12] studied how to teach recursion through a functional programming language. However, these challenges persist, as revealed by Piteira & Costa [13]. Nowadays, the proliferation of new technology presents the opportunity to provide educational tools that can enhance teaching/learning of these programming concepts. To this end, this study is focused on designing a smart learning environment (SLE) to provide better ways for teaching/learning of programming concepts by leveraging the latest technologies such as virtual reality (VR); mobile technology—smartphones, wearables; and ambient intelligence—GPS sensors. Smart learning environment is a technology-enhanced teaching and learning approach [14] [15], introduced as a higher level of the digital environment to support teaching and learning [16]. Although there is no consensus definition of a smart learning environment [17], scholars discussed the characteristic features that make SLE "smart" and what differentiate it from traditional learning environments [18]. These characteristics include adaptivity, context-awareness, ubiquity, personalization, automatic assessment, personalized feedback, motivation, and social-awareness. According to Agbo et al. [19], a smart learning environment is "an enhanced context-aware ubiquitous learning system that leverages social technologies, sensors and wireless communication of mobile devices to engage learners in hands-on experiences and present contents in a stimulating form; capable of connecting the learning community, increasing awareness of the physical environment, tracking and providing learning support". One of the emerging technologies that support the development of SLE is the VR, which creates an environment that seems realistic but not normally experienced [23]. VR environment can provide truly personalized learning by adapting to sensed data from the users' senses and physical environment. Nowadays, smartphones, VR headset (e.g., Gear VR), MOGA pro controller, and many other emerging hardware supports the deployment of VR systems for user's interaction and visualization experience.

In the bid to develop the SLE prototype to support the teaching of basic programming concepts, this study focuses on eliciting the requirements and modelling the proposed system using the Unified Modelling Language (UML). The requirement definition phase is a key component in the design science research (DSR) method [20]. The DSR method, according to Johannesson & Perjons [20], involves five phases, which include problem explication, requirements definition, prototype design, prototype demonstration, and artifact evaluation. These DSR phases allow for iterative refinement of problems, solutions, and approaches until the desired outcome is achieved. After the requirement specification, being the focus of this study, we intend to conduct a co-creation process of the prototype involving the researcher and expected users.

In requirement definition, scholars propose different notations [21]. One of these notations is natural language, which is too ambiguous, imprecise, and lack visualization that aid the understanding of complex concepts. The UML is one of the formal notations that is commonly used as a standard for modelling requirement specification [22]. UML is a standard object-oriented modelling language which allows for visualising, specifying, and documenting software systems. This paper employs UML to design a scenario-based requirement definition of a smart learning environment for learning programming concepts within a VR environment. Similar studies that use UML modelling tools for engineering software systems exist, for example, Iordan and Panoiu [23].

However, the use of UML for designing a smart learning environment that is based on VR technology needs to be explored; hence, the focus of this paper.
This paper is organized as follows: section two presents the proposed VR-based smart learning environment architecture; section three presents the UML models of the intended system; section four describes a scenario of puzzle game for learning programming concepts in a VR-based SLE; and section five briefly discusses the study contributions, conclusions, and the future research.

II. Proposed Architecture of Smart Learning Environment

This section introduces the proposed smart learning environment architecture (see Figure 1). In the system architecture, the input data from the learner environment are acquired automatically or specified by the learner at every learning session. These input data, for example, learner's geolocation, temperature, noise level, motion state, etc., are used by the system to support the learner intelligently.

a. The data layer provides different types of input data required to manipulate the system behavior. For example, a first-time learner is required to provide information regarding his/her learning objectives, previous knowledge, other related profile data that the system may need to provide a more personalized learning experience. Additionally, data from learners' environments could come from their devices' sensors to adapt learning to suit learners' contextual needs. Besides, during the learning session, learners will interact with the virtual learning environment through interface controllers and other input devices.

b. The logic layer contains modules that collectively provide a smart learning experience. These modules will include instructions and program logics. Further, a 3D modeling of the game is expected to take place in this layer. Additionally, the use of C# programming language and the Unity game engine will be employed for the scripting of the game. The components, objects, and event-driven scripting that the Unity game engine provides are useful in creating simple games for learning.

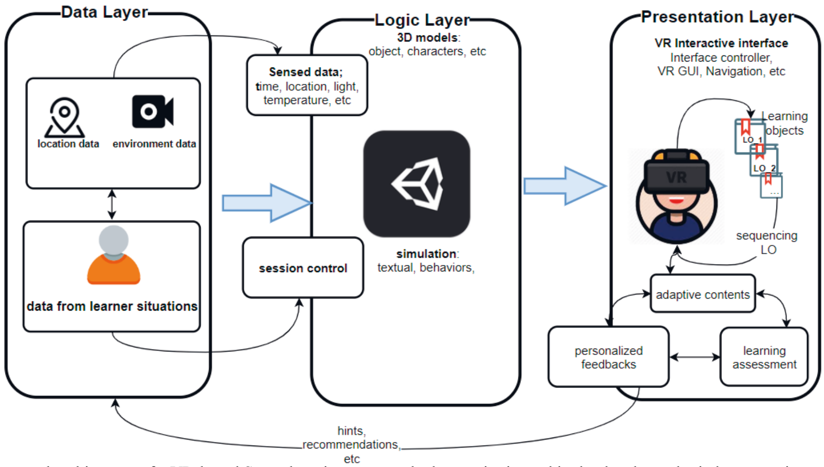


Fig. 1. The proposed architecture of a VR-based Smart learning system: the learner is situated in the data layer; logic layer consists of tools for modelling the VR learning environment; presentation layer immerses the learner in a VR environment for a personalized learning experience

For instance, providing recommendations on how to learn, when to learn, and what to learn at a given time can be helpful to the learner based on the contextual information the system is able to receive. Besides, input data from the user's environment, his/her prior knowledge, if it exists, is also determined and is used to control the internal behavior of the system to achieve a highly adaptive and personalized learning experience.

The VR-based architecture for the smart learning environment aims to immerse learners in a VR environment in terms of visualization and interactions. VR is a technology that creates an environment that seems realistic but not normally experienced [24]. The application of VR has been studied by researchers in different scenarios [25]. This study draws motivation from Bouali et al. [26] to explore VR for teaching and learning of programming concepts. Thus, the system is designed based on Hwang [27] recommendation for designing a smart learning environment, and a brief explanation follows:

c. The presentation layer is a 3-dimensional virtual environment with a head-mounted sensory display. With the smartphone placed in front of a headset, learners can control the behavior of the game object within the virtual environment by using a hand controller/joystick. While learning is taking place, learning objects (LO) automatically adapt to suit the learner based on a set of rules for sequencing LO and outcome of learning assessment carried out during the session. Similarly, sensed data from the device could also influence the behavior of the system in real-time at the presentation layer.

III. UML Design of Smart Learning Environment for Teaching Programming Concepts

This section presents a UML design of an underway smart learning environment to teach programming concepts (i.e., variables, functions, and recursion). The use of UML to model educational tools is significant in order to achieve the design phase of the DSR, as endorsed by other authors [22] [28].

The use case diagram in Figure 2 shows the functionalities of the system (system requirements) where a learner is able to view and take lessons that are designed to teach basic programming concepts, practice quizzes at different stages of learning, and automatic assessment is done to evaluate learning outcome. Learners receive personalized feedback and recommendations based on the learning outcome. The lessons are game-based, which are modelled to contain LO. The outcome from the learner's assessment controls the sequencing of LO. If a learner is evaluated to perform the below satisfactory score, the SLE automatically recommend LOs that simplify the lesson before advancing to the next higher level LO.

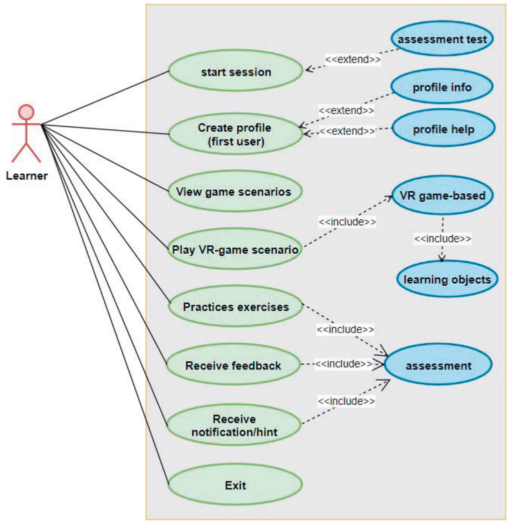


Fig. 2. Use Case Diagram of the smart learning environment: showing the system requirements that are either include a sub-system or extends from a sub-system

Figure 3 is a UML activity model that illustrates how the SLE platform immerses a learner within a VR environment by transforming the learner from the basic concept of programming represented as learning object into a more difficult concept. The diagram shows activities within the system and the flow of control that takes place from one activity to another. For example, a new learner is identified and prompted to provide some profiling data that may be useful to the system to make adaptive and personalized learning possible.

A learner can decide to continuously learn by progressing from one level to another or pause to continue at a later time. The system records the learning progress and evaluates the learner in order to provide personalized feedback, hints, and supports. Similarly, we have presented the sequence diagram of the SLE platform to further show detailed interaction that happens between the actors (users) and the objects (characters/assets). Figure 4 depicts the different sequences of interactions that take place during a particular use case instance. The actor triggers the app after connecting various devices such as Google Cardboard and smartphone. The system initiates a session and responds to the actor with a welcome message and identifies the learner's status, environment, and context. Actors are then redirected to proceed to the virtual learning environment and can control the environment with an input device. As learning progresses at each level, the system evaluates the learner's score. If the score is satisfactory, the system automatically redirects the learner to the next stage of the lesson; otherwise, the system provides hints and personalized support that allows the learner to gain better knowledge.

IV. A SCENARIO-BASED SMART LEARNING ENVIRONMENT FOR TEACHING PROGRAMMING CONCEPTS

Generally, the intended VR-based SLE is expected to hold a few puzzle games that depict real-world problems. While trying to provide solutions to these problems, the player visualizes programming concepts of function and recursion in an interactive manner with the game element for optimal engagement. In other words, during the gameplay, we try to demonstrate these programming concepts to aide player's understanding. For example, we have presented a scenario in this paper called the river crossing puzzle (a computational thinking concept). Aside from teaching students how to acquire problem-solving skills and algorithmic thinking, the river crossing puzzle has been identified to teach programming concepts such as functions and recursion [29]. River crossing puzzle presents a problem where a farmer has got a number of items to be taken across the river. The available boat is able to carry the farmer and only one item at a time. Some combination of these items cannot mutually co-habit in the absence of the farmer. Hence, the farmer ensures that all the items cross the river unharmed.

To solve this problem, the farmer would need to create and call some functions within the gameplay environment and pass parameters to them. For example, function to get an object within the game and place it inside the boat; function to move the object across the river; etcetera will be instantiated and called during the game session. Similarly, some repetition will occur that may require a function to recall itself (recursion). All of these processes happening in the gameplay will demonstrate the practicality of function and recursion and explanation of each event within the gameplay be given to the learner by a game agent.

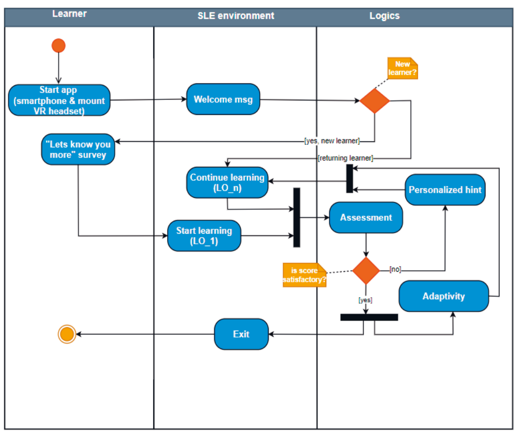


Fig. 3. Activity diagram of the smart learning environment: presenting interactions between each activity; the arrow shows the flow of communication from start to exit points

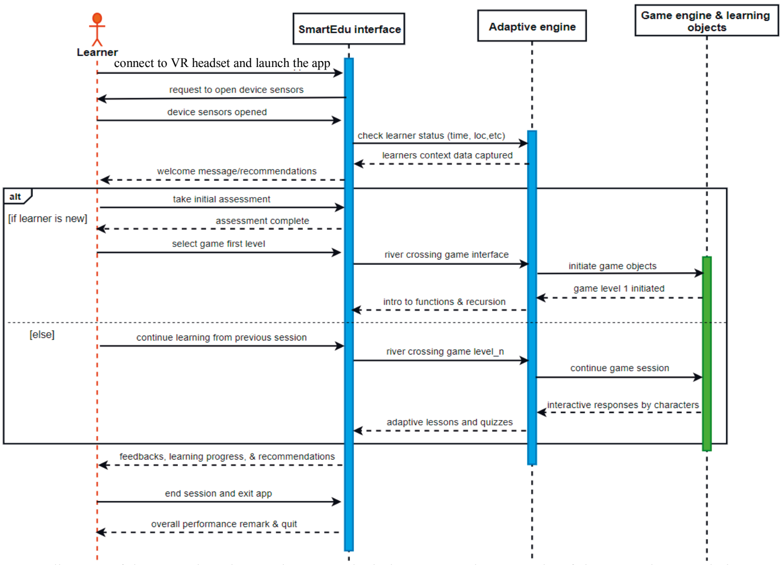


Fig. 4. Sequence diagram of the smart learning environment depicting a gameplay scenario of river crossing puzzle in a VR environment

TABLE I. CONCEPTUAL DESCRIPTION OF THE PUZZLE; GAME LEVELS AND BEHAVIOURS

Stepwise description of the river crossing puzzle
Step 1: the player selects a character (farmer) and at least three items as explained in each level of the game below.
Step 2: The player instantiates functions to collect each item into the boat from the departure terminal and dropoff at the destination terminal.
Step 3: The player calls the function to move each item across the river.
Step 4: move function is recursively called after a conditional statement is evaluated

Two levels description of the puzzle game
Our conceptual design of the game will be modelled using the 3D Unity game engine, where assets and objects are classified into three categories:
• first category is a universal set of animals containing subsets {goat, wolf}, {dog, cat}, & {lion, monkey}
• second category is a set of farm products {banana, watermelon}
• third category contains sets of farmers (river-crossers)

Level 1 rules: In no order, one grouped asset is picked from the first category, one asset is picked from the second, and one river-crosser is picked from the third category, respectively. Alternatively, the arrangement of the assets can be pre-defined within the game so that players do not need to select them. These options can be agreed upon during the co-design process of the game with prospective users.

Gameplay for level 1:
The river-crosser takes selected objects across the river with the function MOVE, which can be recursively called based on the following conditions.
i. if objects (animals) cannot co-habit on their own without causing harm to each other
ii. if an object (animal or farm product) remains to be moved across the river
The task is to move all the objects across the river with a limited number of calling the function MOVE.
After completing the level 1 task, the game rewards, give feedback/recommendations, and initiates the next level.

Level 2 rules: three objects (animals), two objects (farm produce), and one river-crosser are selected.

Gameplay for level 2:
Rules of the gameplay in level 1 apply. However, the number of objects is increased; therefore, the game becomes harder, and the task remains to limit as much as possible, the number of the recursive call to the function MOVE in order to complete the task with the best score

V. CONCLUSION AND FUTURE WORK

The goal of this ongoing study is to design a game prototype of a smart learning environment based on the proposed architecture. Currently, the study elicits requirements for the intended SLE, which are modelled using the UML. However, future work will involve a co-design process of the prototype with prospective users. After completion, evaluation of the prototype to find out the usefulness will be next. Despite the significant role that the smart learning environment plays in transforming teaching and learning at a higher education institution, not too many studies regarding the design of a smart learning environment with UML have been explored [27]. This paper tries to fill the gap by using the UML to design a game-based educational tool. The paper proposes the architecture of a VR-based SLE for teaching and learning of programming concepts and further designed the requirement for the system. A scenario of a puzzle game (River Closing) that teaches the concept of functions and recursion was also presented. This study contributes to the literature by presenting formal notations (UML) for designing a SLE, which allows for the representation that aids the understanding of researchers and other experts in this field. Besides, it also provides a guide for our ongoing research towards the implementation of a SLE platform for programming education, which is underway. Programming remains a challenging topic to teach in games, given the abstract nature of the topics it encompasses. We, however, believe that the visualization quality that VR technology provides will greatly help in communicating these concepts to learners. Via the game puzzle, we do not teach recursion and functions as abstract concepts, but we demonstrate and illustrate how these two pillars of programming are used to solve a practical (although imaginative) problem. We deviate from traditional learning games, where learners are usually asked questions throughout their gameplay to play further levels. Our approach relies on exploiting the qualities inherent to functions and recursion to play the game.

ACKNOWLEDGMENT

Work by the author, Solomon Sunday Oyelere was supported by the Jenny and Antti Wihuri Foundation.

REFERENCES

[1] S. S. Oyelere, J. Suhonen, G. M. Wajiga, and E. Sutinen. "Design, development, and evaluation of a mobile learning application for computing education," Education and Information Technologies, vol. 23, no. 1, pp. 467-495, 2018.

[2] C. Yongqiang, W. Xiaojun and Q. Chengbin, "Computer Programming Education for Primary School Students," In 2018 13th International Conference on Computer Science & Education (ICCSE), 2018.

[3] M. Ebert, "Increase active learning in programming courses," In 2017 IEEE Global Engineering Education Conference (EDUCON), 2017.

[4] S. S. Oyelere, F. J. Agbo, I. T. Sanusi, A. A. Yunusa and K. Sunday, "Impact of Puzzle-Based Learning Technique for Programming Education in Nigeria Context," In 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), 2019.

[5] W. Cazzola and D. M. Olivares, "Gradually learning programming supported by a growable programming language," IEEE Transactions on Emerging Topics in Computing, vol. 4, no. 3, pp. 404-415, 2015.

[6] O. O. Ortiz, J. A. P. Franco, P. M. A. Garau and R. H. Martin, "Innovative mobile robot method: improving the learning of programming languages in engineering degrees," IEEE Transactions on Education, vol. 60, no. 2, pp. 143-148, 2016.

[7] K. Sunday, P. Ocheja, S. Hussain, S. S. Oyelere, B. O. Samson and F. J. Agbo, "Analyzing Student Performance in Programming Education Using Classification Techniques," International Journal of Emerging Technologies in Learning (iJET), vol. 15, no. 2, pp. 127-144, 2020.

[8] A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin and J. Paterson, "A survey of literature on the teaching of introductory programming," SIGCSE Bulletin, vol. 39, no. 4, pp. 204-223, 2007.

[9] B. Law, " Introducing novice programmers to functions and recursion using computer games," In Academic Conferences and Publishing International Limited, ECGBL 2018 12th European Conference on Game-Based Learning, 2018, pp. 325-334.

[10] S. S. Oyelere, J. Suhonen and, T.H. Laine, "Integrating parson's programming puzzles into a game-based mobile learning application," In Proceedings of the 17th Koli Calling International Conference on Computing Education Research, pp. 158-162, 2017.

[11] J.T. Anyango, and H. Suleman, "Teaching Programming in Kenya and South Africa: What is difficult and is it universal?," In Proceedings of the 18th Koli Calling International Conference on Computing Education Research, pp. 1-2. 2018.

[12] P. B. Henderson and F. J. Romero, "eaching recursion as a problem-solving tool using standard ML," ACM SIGCSE Bulletin, vol. 21, no. 1, pp. 27-31, 1989.

[13] M. Piteira and C. Costa, "Learning computer programming: study of difficulties in learning programming," In Proceedings of the 2013 International Conference on Information Systems and Design of Communication, 2013.

[14] L. D., H. R. and W. M., "Contexts of Smart Learning Environments," In Smart Learning in Smart Cities, Singapore, Springer, 2017, pp. 15-29.

[15] H. Peng, S. Ma and J. M. Spector, "Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment," Smart Learning Environments, vol. 6, no. 1, pp. 9, 2019.

[16] R. Huang, J. Yang and Y. Hu, "From digital to smart: The evolution and trends of learning environment," Open Education Research, vol. 1, no. 1, pp. 75-84, 2012.

[17] F. J. Agbo, S. S. Oyelere, J. Suhonen and M. Tukiainen, "Smart learning environment for computing education: readiness for implementation in Nigeria," In EdMedia+ Innovate Learning, Amsterdam, 2019.

[18] J. M. Spector, "Conceptualizing the emerging field of smart learning environments," Smart learning environments, vol. 1, no. 1, p. 2, 2014.

[19] F. J. Agbo, S. S. Oyelere, J. Suhonen and T. M., "Identifying potential design features of a smart learning environment for programming education in Nigeria," International Journal of Learning Technology, vol. 14, no. 4, pp. 331-354, 2019.

[20] P. Johannesson and E. Perjons, A design science primer, CreateSpace Independent Publishing Platform, 2012.

[21] D. Silingas and B. Rimantas, "UML-intensive framework for modeling software requirements," In Proceedings of the 14th International Conference on Information and Software Technologies, 2008.

[22] K. Wiegers and J. Beatty, Software requirements, Pearson Education, 2013.

[23] A. Iordan and M. Panoiu, "Design using UML diagrams of an educational informatics system for the study of computational geometry elements," WSEAS Transactions on Computers, vol. 9, no. 9, pp. 960-970, 2010.

[24] J. N. Latta and D. J. Oberg, "A conceptual virtual reality model," IEEE Computer Graphics and Applications, vol. 14, no. 1, pp. 23-29, 1994.

[25] J. Radianti, T. A. Majchrzak, J. Fromm and I. Wohlgenannt, "A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda," Computers & Education, vol. 147, no. 103778, 2020.

[26] N. Bouali, E, Nygren, S. S. Oyelere, J. Suhonen and V. Cavalli-Sforza, "Imikode: A VR Game to Introduce OOP Concepts," In Proceedings of the 19th Koli Calling International Conference on Computing Education Research, pp. 1-2. 2019.

[27] G.-J. Hwang, "Definition, framework and research issues of smart learning environments-a context-aware ubiquitous learning perspective," Smart Learning Environments, vol. 1, no. 1, p. 8, 2014.

[28] R. P. De Lope and N. Medina-Medina, "Using UML to Model Educational Games," in 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), 2016.

[29] F. J. Agbo, S. S. Oyelere, J. Suhonen and S. Adewumi, "A Systematic Review of Computational Thinking Approach for Programming Education in Higher Education Institutions," In Proceedings of the 19th Koli Calling International Conference on Computing Education Research, Koli, 2019.

ARTICLE VI

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Laine T. H.
"Co-design of mini games for learning computational thinking in an online environment," *Education and Information Technologies, 26*(5), 5815–5849.
https://link.springer.com/content/pdf/10.1007/s10639-021-10515-1.pdf

Education and Information Technologies
https://doi.org/10.1007/s10639-021-10515-1

Co-design of mini games for learning computational thinking in an online environment

Friday Joseph Agbo[1] · **Solomon Sunday Oyelere**[2] · **Jarkko Suhonen**[1] · **Teemu H. Laine**[3]

Received: 27 January 2021 / Accepted: 21 March 2021

Abstract
Understanding the principles of computational thinking (CT), e.g., problem abstraction, decomposition, and recursion, is vital for computer science (CS) students. Unfortunately, these concepts can be difficult for novice students to understand. One way students can develop CT skills is to involve them in the design of an application to teach CT. This study focuses on co-designing mini games to support teaching and learning CT principles and concepts in an online environment. Online co-design (OCD) of mini games enhances students' understanding of problem-solving through a rigorous process of designing contextual educational games to aid their own learning. Given the current COVID-19 pandemic, where face-to-face co-designing between researchers and stakeholders could be difficult, OCD is a suitable option. CS students in a Nigerian higher education institution were recruited to co-design mini games with researchers. Mixed research methods comprising qualitative and quantitative strategies were employed in this study. Findings show that the participants gained relevant knowledge, for example, how to (i) create game scenarios and game elements related to CT, (ii) connect contextual storyline to mini games, (iii) collaborate in a group to create contextual low-fidelity mini game prototypes, and (iv) peer review each other's mini game concepts. In addition, students were motivated toward designing educational mini games in their future studies. This study also demonstrates how to conduct OCD with students, presents lesson learned, and provides recommendations based on the authors' experience.

Keywords Online co-design · Computational thinking · Mini games · Virtual reality · Game-based learning · Smart learning environments · Nigeria

✉ Teemu H. Laine
teemu@ubilife.net

Extended author information available on the last page of the article

1 Introduction

Computational thinking (CT) is foundational knowledge for computer science (CS) students in introductory programming classes. CT is a fundamental step toward building problem-solving skills that can aid the understanding of programming (Agbo et al., 2019a). Studies have shown that understanding the characteristics and common practices of CT, such as problem decomposition, abstraction, algorithmic thinking, and problem-solving skills (Grover & Pea, 2013), are essential for students to excel in programming classes (Eguchi, 2016; Korkmaz et al., 2017). Introductory programming can be difficult for novice CS students (Malik et al., 2019). In the context of a developing country, e.g., Nigeria, this problem persists and has caused increasing failure rates among students who enroll in programming classes (Oyelere et al., 2018; Sunday et al., 2020). A previous study (Agbo et al., 2019a) recognized the potential of exploring CT approaches in higher education institutions (HEI) to allow students to gain the problem-solving skills required for advanced programming classes. Demonstrating CT competencies can be achieved through educational games designed to teach students basic CT concepts (Ch'ng et al., 2019; Mathew et al., 2019; Toivonen et al., 2020). In addition, games and game-based learning (GBL) promote interaction, engagement, and motivation for continuous learning (Al-Azawi et al., 2016).

Educational mini games deployed within a virtual reality (VR) environment can create better learning achievement through complete immersion (Chaves et al., 2019; Bouali et al., 2019). VR mini games allow learners to interact with real-world problems modeled as short mini games to deliver simple and tangible learning objectives (Bouali et al., 2019). Mini games are small, simple games that may exist within a bigger video game that can be played independently (Devisch et al., 2017). Research on transforming the traditional education environment into an immersive VR environment is receiving increasing attention (Freina & Ott, 2015; Virvou & Katsionis, 2008). A VR based mini game is an effective way to ensure that the learner is completely immersed, has a sense of presence, and interacts with objects in a learning environment to gain a better outcome and learning experience (Hickman & Akdere, 2018). In addition, VR has been widely used to support training and instructing students and professionals in different disciplines (Chittaro & Buttussi, 2015; Dias et al., 2019; Lindblom et al., 2021; Tobar-Muñoz et al., 2016). Therefore, a desirable approach to present CT concepts in HEI is to leverage VR technology and GBL (Chaves et al., 2019).

This study is a step toward designing a VR game-based smart learning environment (SLE) to support the understanding of CT. A learning environment is considered smart when it provides a high level of immersion, interactivity, personalization, and engagement to adapt to learners' needs and provide intelligent feedback based on learners' characteristics and learning progress (Agbo et al., 2019b, c, 2020a). Specifically, the current study attempts to demonstrate how co-designing in an online environment helps students design their own learning and develop CT skills. Some of the outputs from this study will form part of the mini games to be developed into a VR game based SLE to support students' further understanding of CT by giving them full immersion, rich interaction, personalized learning experience, presence, and engagement. In other words, the resulting artefact would promote learning and give the students a tangible learning object. A similar study that provides a VR environment to teach and learn CT concepts exists (Parmar et al., 2016); however, the approach used by Parmar et al., (2016) to design their VR game, i.e., Virtual Environment Interactions, which they refer to as VEnvI, to teach CT was not co-designed with CS students, which means it is not completely student-centered. Therefore, our study takes a different approach wherein mini games for learning CT were co-designed with CS students in an online setting. Therefore, the resulting output from this study could be refined into a student-centered solution to learn CT.

Learning elementary programming and CT through indigenous games and puzzles is not new in Nigeria (Oyelere, 2018). However, the design and implementation of indigenous games that engage students to innovatively co-design contextual mini games to support their understanding of CT and programming concepts remain insufficient. This study contributes to knowledge growth by designing and implementing an online co-design (OCD) process with CS students in a Nigerian HEI with the goal of supporting their understanding of CT principles. Specifically, this study aims to engage students in co-designing mini games that would improve their competency in problem abstraction, decomposition, algorithmic thinking, and recursive thinking though OCD. In this study, we refer to OCD as a process that involves researchers and participants in co-designing artifacts remotely by leveraging online platforms. A similar study that recently explored the use of an online environment to co-design a CS curriculum for teachers' professional development exists (Grover et al., 2020). As reported by Grover et al. (2020), teachers and researchers from four US States were recruited to participate in the study. Differing from that study, our study focuses on co-designing educational mini games with CS students to support their own learning through the OCD process. It has been reported that the OCD process itself creates opportunities for students to learn CT through collaborative co-designing activities (da Costa et al., 2017). Specifically, this study contributes to existing knowledge by showing how to co-design educational mini games in an online environment. The authors anticipate that co-designing in an online environment will become a future paradigm for conducting user-centered research, particularly if global challenges, such as the Coronavirus pandemic, persist. To the authors' best knowledge, a study to investigate the OCD process to create digital mini games in the context of Nigeria has not been conducted before. In addition, this study reports lessons learned from the OCD process and, based on the experience gained, provides recommendations that may be useful to educational game researchers. To achieve our objectives, we will provide answers to two research questions:

RQ1. How does OCD of contextual mini games with students in a Nigerian HEI work from the researchers' perspective?

RQ2. What are the experiences of students participating in an OCD of mini games to support their CT skills?

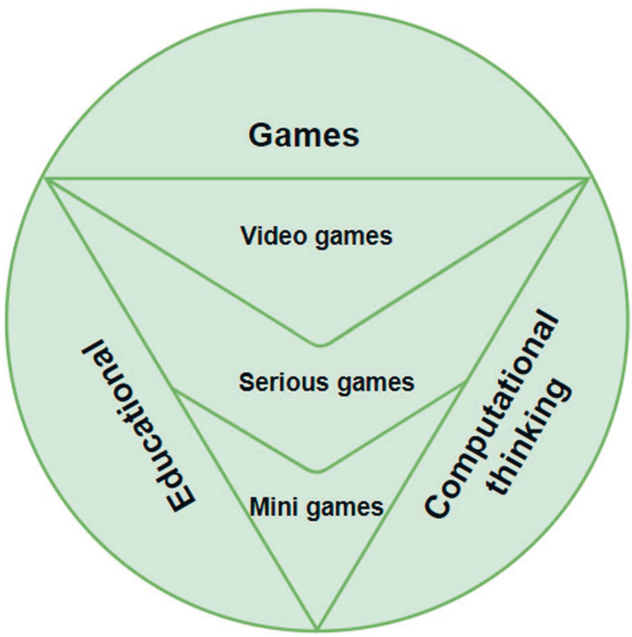
2 Background and theoretical framework

This section explains the major concepts and presents the contextual background of this study. In addition, we present an overview of relevant theories that support the design and development of SLEs for VR mini games to teach and learn CT and programming concepts.

2.1 Mini games and VR

Mini games have attracted increasing attention as educational tools that have the potential to teach difficult concepts to students (Asal et al., 2018). What is a mini game? We define educational mini games as short types of video games that are built within an educational application and that are independent in terms of game elements and mechanics, thus making them playable on their own (Fig. 1). Unlike serious games (Chittaro & Buttussi, 2015), mini games are flexible, simple, and easy to learn. These characteristics make it possible to achieve a small unit of learning, i.e., a mini game, similar to microlearning (Devisch et al., 2017). Moreover, learning in small chunks can provide a high level of interaction and aid learners' memory (Bruck et al., 2012). Designing mini games to support learning CT concepts can positively motivate students (Ch'ng et al., 2019) by structuring learning into smaller units (mini games) so that students can quickly grasp CT knowledge (Bakker, 2014). For example, to complement online learning, Arnab et al. (2020) conducted a study

Fig. 1 Operational definition of educational mini games



on designing mini games to support microlearning within an open educational resource context. That study found mini games to be a modular approach in terms of portability and flexibility, highly interactive, engaging, and connects learning objectives. Therefore, this study is motivated by these findings to create mini games to support understanding of CT concepts.

Currently, mini games deployed in VR environments are receiving considerable research attention (Parong & Mayer, 2020). Studies show that mini games played in VR environments present tremendous opportunities for adequate immersion, motivation, engagement, and interaction to enhance learners' learning experience (Chaves et al., 2019; Bouali et al., 2019). VR technology is not new, however, recently, its deployment and use in the field of education and training has increased dramatically (Zhou et al., 2018). Current technologies have made it possible to deploy VR applications on small devices, such as smartphones; thus, VR applications are accessible to many users, especially in developing countries. For instance, in Nigeria, the use of VR technology to support learning is possible since most students possess smartphones capable of deploying such learning applications (Agbo et al., 2019b, c). Another beneficial characteristic of VR technology in an educational context is the ability to intelligently detect and track head movement, hand gestures, and body movement using embedded sensors (Virvou & Katsionis, 2008). These features are useful in creating highly interactive educational mini games that engage learners for an enhanced learning experience. Furthermore, VR technology and devices are considerably more affordable than they were decades ago. For example, companies, such as Google and Facebook are currently producing affordable head-mounted displays (HMD). This affordability creates ample opportunities for the deployment of VR mini games in the context of developing countries, such as Nigeria, where university teachers and students can afford HMDs. Consequently, the authors are conducting research to support students to learn CT concepts through the OCD of mini games. In addition, the resulting artifacts that would emerge from the co-designing process, i.e., VR mini games, would provide enhanced CT learning experiences.

2.2 Learning theory: Constructivism, experiential, and participatory

According to scholars, GBL is generally connected to constructivism and experiential learning theories (Koivisto et al., 2017; Wu et al., 2012). It has been recognized that GBL provides effective learning outcomes in terms of immersion, motivation, and stimulation for continuous learning (Alamanda et al., 2019; Huizenga et al., 2019; Tokac et al., 2019). A recent study by Radianti et al. (2020) revealed that few studies on VR game-based educational applications connect the foundation for their research to any learning theories. Therefore, this section focuses on the fundamental learning theories that connect GBL, educational VR applications, and the co-design process. While this study does not dwell deeply on the learning theories, it connects the co-design process and GBL to the existing and relevant theories that support the aim of this study. The overall goal of connecting these relevant theories is to provide the foundation for designing a game-based VR interactive learning environment to support CT and programming education, which is the authors' long-term plan where the current study serves as input.

Generally, educational tools are expected to support teaching and learning processes in formal, nonformal, and informal settings (Pérez-Sanagustín et al., 2014). Currently, the use of games and GBL has become a strong approach for creating educational tools (Qian & Clark, 2016). In addition, GBL has been investigated to determine whether it can provide rich instructional content to learners in various disciplines, such as engineering and computing, health and medical science, art and design, languages, and mathematics (Chang & Hwang, 2019; Tokac et al., 2019). For example, previous studies have considered the use of games to gain more awareness and knowledge in sport and physical health education (Mubin et al., 2016; Regal et al., 2020), games to promote learning in the fields of arts, culture, and tourism (Cesário et al., 2019; Rinnert et al., 2019), and games focused on engineering and manufacturing education (Perini et al., 2018; Tobar-Muñoz et al., 2016).

The co-design process has become a popular method for designing a GBL tool to support students (Loos et al., 2019). The goal of the co-design process is to increase originality in the game in terms of meeting the requirements of the targeted players and to avoid biased assumptions arising from designers and developers of the game (Vetere, 2009). In addition, co-designing educational mini games with targeted stakeholders provides opportunities for designers and developers to uncover the differences in players' individual learning characteristics that can be modeled into the game to enhance personalization of the gameplay (Castro-Sánchez et al., 2019; Mariager et al., 2019; Thabrew et al., 2018).

Broadly, GBL and the co-design concept are grounded in three interwoven theories: constructivism theory (Jong et al., 2010), participatory design theory (Gomez et al., 2018), and experiential learning theory (Kolb, 2014). For example, participatory design theory is founded on constructivism theory (Spinuzzi, 2015). While constructivism theory postulates learning as an active, constructive process where learners create their own mental representation of learning objectives, participatory design theory deals with methodological approaches that ensure that users of technological artefacts are involved in the entire design process (co-design) of what affects them in order to create more efficient and usable systems (Bowen, 2010; Robertson & Simonsen, 2012; Rosenzweig, 2015). On the other hand, experiential learning theory views learning as a process whereby concepts are derived from and are continuously modified by experience, that is, "ideas are not fixed and immutable elements of thoughts but are formed and re-formed through experience" (Kolb, 2014, p. 26).

Figure 2 shows the relationship between these theories and how they are connected to provide the foundation for the design and development of VR game-based smart learning environments to teach CT and programming concepts.

These theories are relevant to this study because they provide the foundation for building a learner-friendly smart learning environment through the OCD process. According to Kommers (2003), combining experiential learning in a VR environment with constructivist concepts can provide a standard interface for an immersive learning experience that meets the expectation of future learning tools.

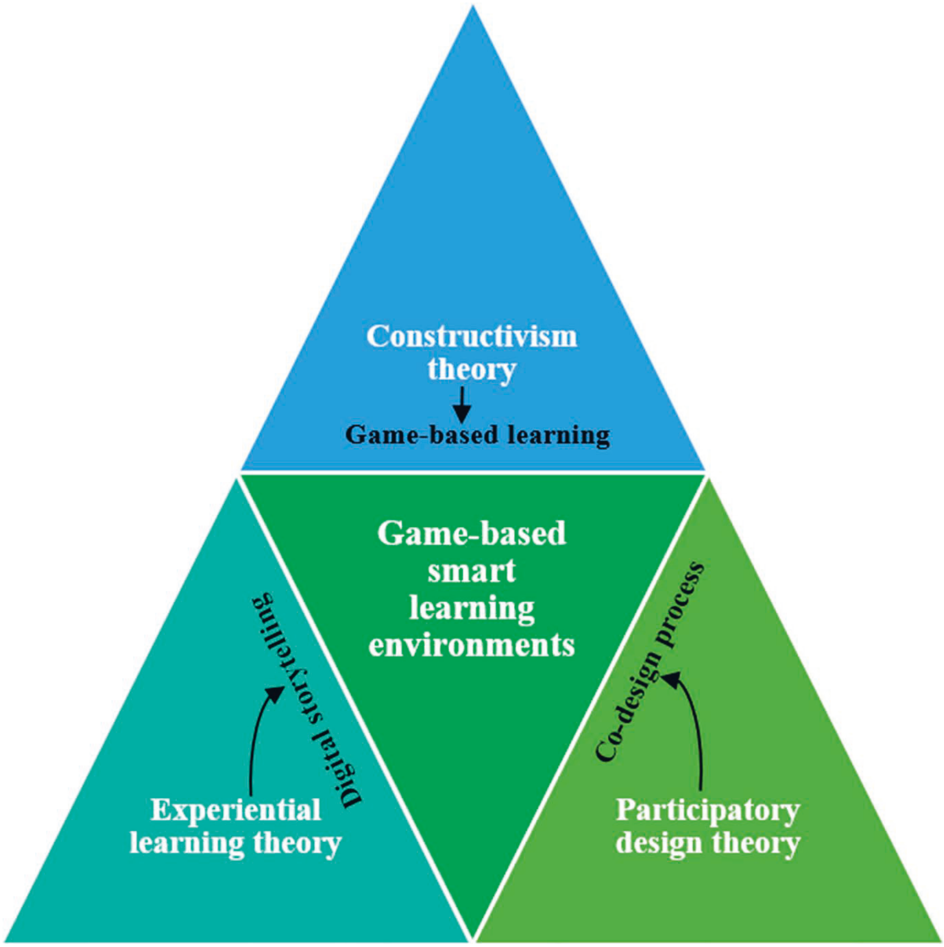


Fig. 2 Interrelationship of relevant theories of game-based smart learning environments

2.3 OCD of educational games

Different methodologies and techniques can be applied to design digital games to support learning. For example, Walsh et al. (2010) described a technique called “layered elaboration” as a new approach to co-design with children. According to the authors, this technique is a useful way to co-design since an initial idea by a co-designer can be built upon by another designer without modifying the original items. Several activities are involved in co-designing a digital game with stakeholders, such as brainstorming exercises, card sorting, sticky note exercises, group tasks, diagramming, and rapid paper prototyping (Ruiz et al., 2018). In our study, a co-design technique motivated by Walsh et al. (2010) was used since it sufficiently fit the context of educational game design to support students who, in this case, are the primary users of an artifact that would emanate from this co-design process (Havukainen et al., 2020).

The traditional face-to-face method is often used to co-design educational or commercial product. In this method, the researcher/facilitator meets with co-designers for a participatory process that leads to the creation of a new artifact (Bonsignore et al., 2016; Hjelmfors et al., 2018). However, conducting a co-design process in an online environment is an innovative approach that could be explored by researchers. This method provides a flexible way for researchers, facilitators, and stakeholders to participate in a co-design process through an online medium. During the co-design process all communication between stakeholders could occur with the help of an online platform. This type of OCD process is useful, particularly in situations where stakeholders are faced with various difficulties, such as distant geographical locations, time differences, and other circumstances that make it impossible to have a face-to-face meeting, such as a pandemic.

A reasonable number of studies connected to OCD process can be found in the literature. Some of these studies focused on co-designing commercial products to improve customers' motivation and engagement to buy and use a product (Dix et al., 2012). A few studies focused on co-designing online courses to support pedagogy and educators' professional development (Grover et al., 2020; Marín et al., 2018). To investigate co-designing an educational tool, Walsh et al., (2012) conducted an OCD process with children to design a prototype for a computer-based design tool—DisCo. This tool was designed to facilitate collaborative work through drawing and annotation of objects among geographically distributed participants. Their study revealed some limitations, including the inability for the participants to draw conveniently on a computer screen (Walsh et al., 2012). In addition, Friedrich (Friedrich, 2013) conducted a study on how social media platforms can support the participation of stakeholders in an online co-designing process. The output of this study was the online collaboration tool called Open Web Lab, which provides social media elements to facilitate the co-design process (Friedrich, 2013).

The recent global pandemic, which has affected researchers from all fields, including education, has created the opportunity to develop innovative ways of co-designing artefacts with users in an online environment. Our idea of an OCD process entails a situation where the researchers and students implement the co-design remotely through online platforms. The students worked remotely within a closed online group to collaborate and create their mini game prototypes. However, students could choose to have a physical meeting without the researchers to implement their collaborative tasks because, when the study was conducted, physical meetings were still possible in the country where the students reside. This OCD process creates flexibility for students to effectively co-design with researchers. There are a few studies on co-designing educational tools in an online environment; however, to the best of our knowledge, no studies focus on co-designing educational mini games to support CT skills in an online environment, specifically in the context of Nigeria. Therefore, among other objectives, this study contributes to the body of knowledge in this regard.

2.4 Educational games and interventions for CT in the Nigerian context

This section presents an overview of concepts and interventions to support CT-related skills in Nigeria. To start with, the game approach, simulations, and multimedia tools to support CT skills in Nigeria have been investigated and found to be useful in terms of engaging and enhancing students' learning outcomes (Adetunji et al., 2013). Scholars are making efforts to promote teaching and learning through the design and implementation of games to complement the traditional approach of textual materials in Nigeria (Ogunsile & Ogundele, 2016; Oyelere, 2018; Bassey et al., 2020). For example, in the field of health and medicine, the use of educational games is gaining momentum. Fisher (2020) recently explored the potential of an educational game to facilitate civic learning in the context of Nigeria. Fisher's study examined how game approaches provide opportunities for civic engagement through participatory learning in a developing country. The author revealed that the use of games for civic education could facilitate community discussion and democratic deliberation through participatory learning. In addition, Bassey et al. (2020), designed a board game, Worm and Ladders, to promote education on good hygiene practices to control soil-transmitted helminthiasis (parasitic worm infection) in southwestern Nigeria. This study revealed the potential for teaching and promoting effective hygiene behavior among young people through the use of board games to complement other teaching methods (Bassey et al., 2020). Furthermore, a study on nutrition education among adolescents was conducted using an educational game to complement teaching and learning about how to practice healthy eating (Ogunsile & Ogundele, 2016). The findings from Ogunsile and Ogundele's study (2016) indicate that the use of the game for nutrition education is an effective approach to enhance adolescents' knowledge, attitudes, and healthy eating practices in southwestern Nigeria.

In the context of science, technology, engineering, and mathematics (STEM), a few studies were seen to set the foundations for the teaching and learning of CT in Nigeria. For example, recent studies have focused on developing the learning and teaching framework to build teacher's capacity to support CT education (Emembolu et al., 2019; Ramin et al., 2020). Through a concept of TeachAKid-2Code, Emembolu et al. (2019) recruited educators across nine Nigerian States to provide training and capacity building in order to increase the number of STEM educators in Nigeria. In another setting, Talib et al. (2019) conducted a study on enhancing students' critical thinking and CT skills using graphic calculator (GC) technology. This study showed that GC can be maximized as a pedagogical tool to benefit students' CT skills. Similarly, Adetunji et al. (2013) revealed that students from southern Nigeria who learned mathematics through a digital game performed better in problem-solving than those exposed to the traditional method.

In the context of HEI, not too many studies that addressed CT and programming education in Nigeria are available. However, it is worthy to note that some studies have been conducted to facilitate CT and basic programming education in the context. For example, Oyelereet al. (2018) designed and developed a mobile learning application to facilitate elementary programming by integrating Parsons programming puzzles into a traditional board game. Evaluation of the mobile learning application with Nigerian students reported that the tool promotes teaching and learning of programming by engaging the students in problem-solving through an indigenous game (Oyelere et al., 2019). It is also interesting to see how scholars from the university where the participants of this study emanate are making contributions toward developing students' CT and programming skills (Oladipo & Ibrahim, 2018). For instance, an intervention to support students on problem-solving and self-learning (CodeEazee) has been developed using Python (Oladipo & Ibrahim, 2018). In another study, Oladipo et al. (2017) developed a tool called FULangS using C language for the purpose of guiding the teaching of scripting through a command-line interface. All the studies presented from the HEI context have in some ways contributed toward developing CT skills among students in Nigeria. However, the focus has not been on the core concepts of CT, such as algorithmic thinking, problem decomposition, and recursive thinking. Therefore, there is a need for more research focusing on establishing teaching and learning of CT concepts in all levels of education including HEI. In this sense, the current study engages students to design contextual mini games through which they can gain CT knowledge and to develop the resulting mini games into a VR game based SLE to further support students in understanding programming concepts. As established earlier, an average Nigerian HEI student can afford a simple VR headset of about 5 US Dollars. In addition, teachers, and most of the students in Nigeria already possess smartphones that can support VR applications (Agbo et a., 2019c).

3 Methodology

This section explains the research procedures, participants, and description of the participatory student-centered design method (Bonsignore et al., 2016; Gomez et al., 2018) implemented through an OCD process and shows how a series of activities were carried out, which are explained in the subsections.

3.1 Participants and ethical consideration

This section provides information about the student participants, the research team, and the students' coordinator involved in the OCD process. The research team included a doctoral researcher, who is also the software developer, and a postdoctoral researcher, who is also one of the doctoral researcher's supervisors. Twelve CS students (eight males, four females) studying at a university located in the north-central region of Nigeria were recruited to participate in the study. Although the students were off campus during their participation, some were living in the city where the university is located. The study was planned to be completely online based. However, we discovered that some of the students had weak internet connections and would not be able to fully participate as a result. For students living close to the university who did not have a strong internet connection, the students' coordinator made a provision to allow them to use the university's CS lab. This alternative provision was made to allow the students in this category to participate in the first meeting that provided important information regarding how to participate in the study.

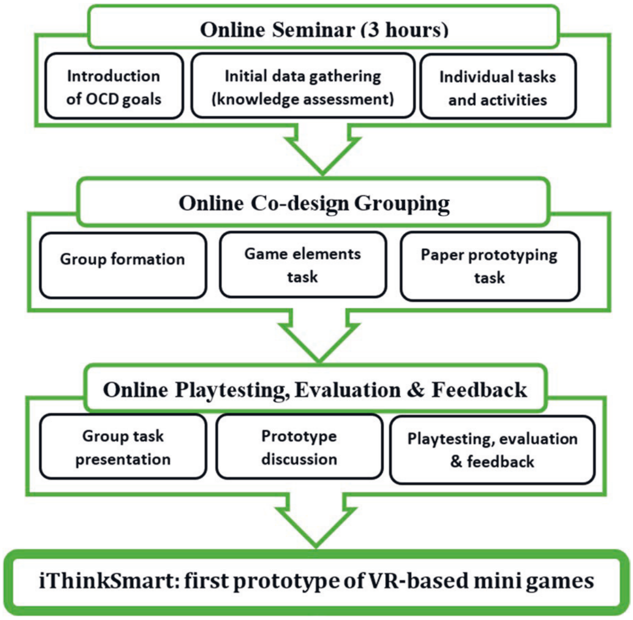
The students who participated in this OCD process were at different years/levels in their study. Four students were in their second year (200 level), four were in the third year (300 level), and four were final year students (400 level). The purpose of having students from various levels (years) of study to participate in the OCD process was to obtain an inclusive experience in terms of creativity and design perspective from all stakeholders. All the students who participated in the study had completed an introduction to computer science (CSC 101) course. Based on the targeted university's CS curriculum for year one, the participants should have been introduced to the basics of computer science, introduction to problem solving methods, and algorithm development. This prior knowledge was necessary to make each participant contribute meaningfully to the co-design activities. The student's specific data were obtained during a seminar while each person introduced themselves. Because the students are studying CS at the same university, they would probably be familiar with one another.

This study conforms to the ethical principles and guidelines of the Finnish national board on research integrity regarding responsible conduct of research (RCR). Prior to the seminar, the researchers obtained informed consent from each participant electronically. Some students were contacted to obtain their consent via phone calls. During the recruitment process, the aim of the study was explained to the students. It was mentioned in the consent messages that "the aim is to engage students in co-designing mini games that would improve their CT skills and provide state-of-the-art information regarding the use of recent technology such as VR in the educational field." The students were informed of their right to stop participating in the OCD process at any stage. The students' right to withdraw from participating in the OCD process was repeated during the seminar. Consequently, the rights and interests of the students were fully respected throughout the OCD process. Students also consented to allow the data collected during the OCD process, including their photos, drawn images, and text, to be used anonymously for research purposes.

3.2 Participatory student-centered co-design process in an online environment

To conduct this study, the researchers opted for OCD due to widely imposed pandemic-related travel restrictions. The travel restrictions made it impossible for the researchers and participants, who were located in different countries, to conduct a face-to-face co-design process. In such a situation, the OCD process became the necessary and available option (Grover et al., 2020; Walsh et al., 2012). In addition, it is worth mentioning that at the time this research

Fig. 3 Screenshots showing the Zoom meeting between researchers and students participating in an online seminar (first session of the OCD process) where some students connected from different locations and others were gathered and connected from a university CS lab



was conducted only two Nigerian states, Lagos and Abuja, had reported Coronavirus cases, with a very low number of infections. Therefore, Nigerian authorities had not banned gatherings; Thus, people were allowed to carry on with their normal activities. Since the students could be reached remotely, the researchers primarily leveraged the Zoom[1] and WhatsApp[2] platforms to conduct the entire four-week OCD process. Research on the use of social media, such as WhatsApp, to facilitate learning has been conducted in the same context (Agbo et al., 2020b). Students participating in the OCD process required an internet connection and Zoom software installed on a computer or smartphone. Because some of the students could not access the internet from their locations, we arranged for them to use a university computer lab where a few students gathered to attend the first meeting. As shown in the screenshots in Fig. 3, the first meeting was held virtually.

During the first meeting, a three-hour seminar was held to provide an extensive introduction about the objectives of the co-design process, activities, tasks to be completed, and expected outcomes. Subsequently, the meeting continued at a group level with four students in each group created on the WhatsApp platform. The WhatsApp groups were created to facilitate collaborative work in the co-design activities. Group WhatsApp activities lasted for three weeks, and the meeting schedule and strategies in each group were defined independently based on what suited the members. However, to ensure effective collaboration among the group members on the assigned tasks, a researcher was assigned to each group to provide guidance and motivation to the participating students. Since the students could communicate outside the

1 https://zoom.us/

2 https://www.whatsapp.com/?lang=en

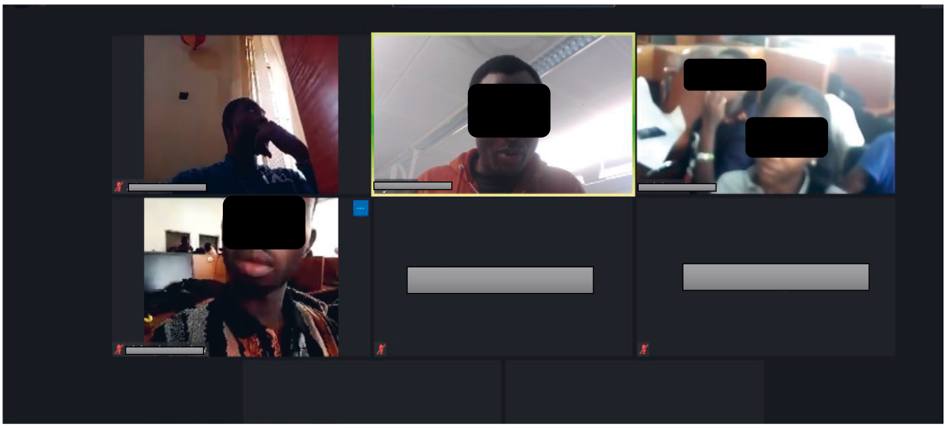


Fig. 4 Implementation flow of the online co-design methodology

WhatsApp platform, other forms of collaboration among the students at the group level, including physical meetings, were possible. The researchers may be unaware of such meetings. After three weeks of group collaboration, each group submitted their completed tasks, which included all materials used in the conceptual design of the mini game, story and scenarios, puzzles, and problem-based challenges. The students were expected to connect their mini games and puzzles to at least one aspect of CT (problem-solving, abstraction, pattern recognition, and recursion).

The OCD process includes: (i) an online seminar (3 h) (ii) online co-design grouping, and (iii) online playtesting, evaluation, and feedback. The OCD methodology, as shown in Fig. 4, was implemented to reflect the interconnection of the relevant theories identified in Section 2.2 (Fig. 2). The first stage of the OCD process, a three-hour online seminar, introduced the objectives of OCD, emphasized the goal for co-designing, and described the types of activities that the students would undertake during the entire OCD process. For example, it was explained that the OCD goal was to conceptualize contextual mini games that teach basic CT concepts and that students would be able to understand CT through the conceptualization and design of mini games. These mini games would serve as input for the authors' long-term plan to develop a VR application containing mini games to support CT skills. Therefore, students were briefly introduced to VR technology and the concept of smart learning environments during the seminar.

During the seminar session, activities were recorded using Zoom's live recording option to allow the researchers to play back when necessary. In addition, during the first session, we took photos of the participants in the physical lab and screenshots of those online and connected through Zoom. The Zoom chat board was used to collect data from students who were online and paper-based exercises, such as sticky notes and flipcharts, were used to collect data from students in the university lab. As explained earlier, this approach of online activities creates a flexible way of co-designing where some students were connected to the virtual meeting from a physical location in a university lab and other students, including the researchers, were connected from different locations. The first activity in the seminar was self-introduction.

Afterwards, an initial assessment of the participants was conducted using an online survey form. The motivation behind the initial assessment was to understand the participants' background and prior knowledge in terms of discipline and level of experience of games and game elements. Furthermore, for students who were physically at the computer lab and could not use the online survey form, an individual sticky note exercise was used to elicit the same information gathered from the online survey.

The second stage of the OCD process was the online co-design grouping. We were guided by the students' year of study and their gender to form the group. For instance, group one consisted of students from year two to year four. The researchers ensured that there was one female in each of three groups. The researchers relied on the information already gathered from the students to select who should belong to what group. Details of the participants are provided in Section 3.1. In addition, the grouping was done such that every participant could meaningfully contribute (Gomez et al., 2018). A three-week agenda for the OCD activities was set by each group. A group leader was nominated by the group members. Activities within the group commenced immediately at the seminar and continued online for three weeks.

In addition, during the seminar, activities such as developing a game element wish list were completed on a group basis. Students discussed which game elements they would prefer to see in their conceptual mini games and made a wish list of such game elements. Then, students participated in a breakout session in groups to brainstorm on concrete problems they will identify to solve with mini games and how they want to solve those problems. The outcome from this brainstorming session was harmonized, written on a flipchart or the Zoom chat board, and presented upon returning to the main Zoom meeting room. The next group task was game ideation and prototyping. The researcher provided guidelines for the prototype exercise: (i) prototypes should consider two separate ideas of contextual mini games, (ii) the games should present CT problems which they intend to solve, (iii) the games should comply with the fundamentals of game elements (space, components, mechanics, goals, and rules) (E-lineMedia, 2011), and (iv) a contextual storyline for the conceived mini games should be created. The participants were encouraged to freely discuss all ideas and be creative in their design and prototyping (Jong et al., 2010). The context for their storyline could be anything that appeals to them from their experience (Kolb, 2014). For example, they could choose to create a storyline about the Nigerian context, such as ethnicities, government, politics, and the environment.

The final stage of the OCD process focused on online playtesting, evaluation, and feedback. Playtesting a co-designed mini game provides an opportunity for end users to evaluate their co-designed games (Eckardt & Robra-Bissantz, 2018; Gerling & Masuch, 2011). This stage required each group to submit their output and make a presentation. During the presentation, the groups discussed their ideas and evaluated each other. After the three weeks of collaborative group co-design, group leaders submitted all documents they had prepared during the task implementation, including paper prototypes, sketches, voice notes, videos, mock-ups, and PowerPoint presentations. All groups tried to model user-centered designs with digital prototyping using wireframes, e.g., using Corel Draw (Agboet al., 2019a, 2020b; Laine et al., 2020).

3.3 Data collection

In this study, the authors collected both quantitative and qualitative data (see Appendix). At the beginning of the seminar, the researchers administered a short online survey (Anyango & Suleman, 2018) where participants gave their responses regarding their prior knowledge and experience about co-design, games and game elements, and expectations from participating in the study. The survey instruments were administered on a five-point Likert scale (1 = strongly disagree; 5 = strongly agree). Items in the questionnaire were designed by the researchers based on the context of this study and were validated by three CS or educational technology experts prior to being administered (Anyango & Suleman, 2018).

Aside from the quantitative data collected during the seminar, the authors used different approaches to collect qualitative data. For example, qualitative data were obtained from sticky note exercises, Zoom chat content, and recordings collected during the seminar (da Costa et al., 2017). In addition, voice notes, paper designs, and prototypes were collected asynchronously during the group co-design activities (Spencer et al., 2019) (Fig. 5). In addition, at the end of the OCD process, a semi-structured interview was conducted with a single randomly selected student from each group. The reason for conducting an interview instead of administering a post-questionnaire survey was to gather more specific responses from the students

Fig. 5 Portion of the sticky notes from a data gathering exercise

regarding their experience after participating in the OCD process. Besides, the sample size for the participants is small; thus, we considered that an interview would be more meaningful. The interviews were conducted through the Zoom platform.

The interviews were recorded and transcribed. The transcribed interviews were coded, and the guidelines provided by Moser & Korstjens (2018) were followed to present a content analysis of the coded transcript. In addition, the quantitative data were analyzed using descriptive statistics (mean and standard deviation). Analysis of the data collected from quantitative and qualitative method are presented in the findings section to complement our findings in terms of validity and reliability from the mixed-method approach (Natow, 2020).

4 Findings

The early part of this section presents the results that are focused on addressing the first research question. Specifically, we show the findings from the implementation of the student-centered OCD process including the analysis of seminar exercises, designs, and other data collected during the process. The remaining part of the section presents the results that address the second research question, i.e., participants' experiences after undertaking the OCD process of co-designing mini games.

To answer the research question (RQ1) *"How does OCD of contextual mini games with students in a Nigerian HEI work from the researchers' perspective?"* we begin by presenting the background information of the participants' in Section 4.1 and proceed to analyze the data collected during the OCD process. The analysis of the participants prior experience is necessary for the study as it helps to find out their post OCD experience.

4.1 Descriptive analysis of participants experience prior to participating in the OCD process

This study revealed several information items regarding the participants' prior knowledge and experiences in terms of the co-design process, games, and game elements. As shown in Table 1, the majority ($\mu = 3.82$, $\sigma = 1.17$) of the participants indicated that they had participated in an online seminar. Surprisingly, a slight majority of the students ($\mu = 3.64$, $\sigma = 1.12$) indicated that they had participated in an OCD process. Since this response was given before the actual seminar, students may have misunderstood the question as to what exactly an OCD process means.

As shown in Table 2, a majority of the students are active game players, primarily for fun ($\mu = 3.91$, $\sigma = 0.83$). However, a slightly smaller number of students ($\mu = 3.73$,

Table 1 Prior participation in seminars

Items	μ	σ
I have participated in an online seminar	3.82	1.17
I have participated in an online co-design process	3.64	1.12

Table 2 Prior knowledge of and experience with games and game design

Items	μ	σ
I am an active player of games	3.82	0.75
I play games to gain more fun	3.91	0.83
I play games to gain new knowledge or learn new things	3.73	0.79
I have frequently participated in game design	3.55	1.03
I am familiar with the elements of games	3.64	0.81
I am aware of how game elements operate	3.55	0.69

$\sigma = 0.79$) play games to acquire new knowledge. A handful of students ($\mu = 3.55$, $\sigma = 1.03$) indicated that they had participated in a game design process; however, we observed that the data points of respondents are spread out over a wider range of values. A slight majority of students ($\mu = 3.64$, $\sigma = 0.81$) claimed to be familiar with game elements.

In addition, the study investigated the students' expectations regarding what they aimed to gain from participating in the OCD process. The results revealed that participants who attended the OCD seminar had high expectations. For example, the data presented in Table 3 indicates that most of the participants ($\mu = 4.36$, $\sigma = 0.81$) were eager to participate in the OCD seminar because they understood and welcomed the purpose of the seminar.

While most items indicating students' expectations in Table 3 show high scores, the question of whether the seminar will help the students identify a new CS area got a low score ($\mu = 2.45$, $\sigma = 1.57$). This result revealed that since students were already familiar with the seminar's purpose, as indicated in the invitation notice, they may not be expecting something new. Besides, all participants had passed an introductory programming course; thus, the topic would not be new to them.

In addition, the study analyzed the participants' responses regarding their personal experiences in understanding programming topics, i.e., what topics they found easy or difficult in their programming classes. The reason is to concretize the need for the current study by ensuring that student-centered game prototype is being designed and to provide useful information for future study. Although the sticky note exercise that was conducted during the seminar revealed certain information in this regard (Fig. 5), the analysis shown in Fig. 6 provides a clearer picture of the programming topics that the CS students found difficult to understand. The results shows that 55% of the students ranked "*recursions*" as very difficult to understand while 45% indicated that "*file and exception handling*" and "*methods*" were very difficult programming topics.

Table 3 Participants' expectations of the OCD process

Items	μ	σ
I am eager in participating in the seminar because the purpose in the invitation notice was clear	4.36	0.81
I am eager in participating in the co-design process because I like to collaborate and share knowledge	4.18	0.87
I expect to learn new things in the co-design seminar	4.45	0.93
I am hoping that the co-design seminar will help me identify new areas of computer science	2.45	1.57
The seminar will provide me the opportunity to design my own game	4.45	0.93

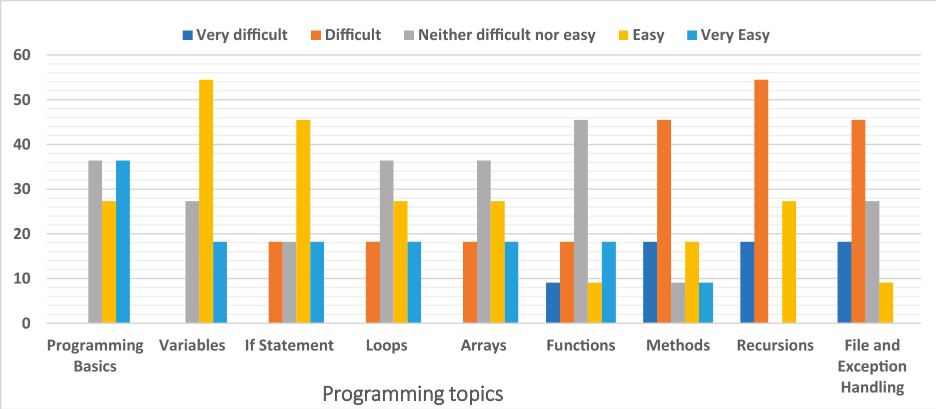


Fig. 6 Participants' experience, ease/difficulty in understanding programming topics

4.2 Researchers' analysis of conceptual and contextual mini games co-designed by HEI students in Nigeria

This section presents the analysis of conceptual and contextual mini games co-designed by CS students in Nigeria following participatory design theory (Gomez et al., 2018). The analysis is based on researchers' perspective about concepts, activities, and student-centered approach followed by participants in the OCD process to create contextual mini games in an online environment. Here, the focus is on brainstorming with students on game features, game elements, and paper prototyping.

4.2.1 Activities during the seminar: Sticky notes, selection of game elements, and game prototyping (paper and mock-up)

The sticky notes activity was designed to motivate students to begin to prepare their minds and refresh their memories about experiences they have obtained from playing any kind of game. Related to our discussion of various types of learning theory (Section 2.2), these experiences provide opportunities for students to actively contribute in terms of what functions they really wish to have in their mini games. The desires of each participants expressed in the resulting wish list indicate that they have a certain level of game experiences, as shown in Table 4.

Furthermore, in their groups, the students independently designed some conceptual mini games and puzzles using a paper prototype and later a wireframe. One of the researchers was added to each WhatsApp group to monitor the progress of the students' co-design. Figure 7 lists game elements and paper prototypes of mini games from the group OCD activities.

In addition, the researchers provided information on how each group could present their idea in digital form, e.g., by using a wireframe. Some of the groups were able to use software, such as Corel Draw, to transform their paper prototype into a mock-up design, as shown in Fig. 8a and b.

Table 4 Classification of co-designers wish list of game elements

Game elements	Wish list classification
Player	auto playing, character, avatar,
Environment	user interface, interaction, experience, accessibility, easy to play, user friendly, interactive space
Input/output	navigation, buttons, clear instructions, movements, gesture detection
Rule & Challenge	competitive, high challenge, constraints
Rewards & badges	scores, rewards, win, goals, winning sounds, leagues, cups,
Social element	collaboration, teamwork
Aesthetics	graphics, animation, colors, background, sounds, themes

Worthy of note is the contextualization of ideas of game scenario, storylines, and game puzzles for teaching and learning of CT concepts (Malik et al., 2019) by the students who created themes around the context. For example, one of the mini games was named *Mount Patti treasure hunt* (Fig. 8a).

The storyline for this game was connected to a popular mountain located in the same city where the students reside. *Mount Patti* is 1503 feet tall above sea level and had long served as a tourist center in the locality. *Patti* is a native word meaning "hill." This mountain has many historical relics, such as the government house of Nigeria's first capital city. The co-designers created their storyline around this history and gamified climbing the mountain. The simple rule for exploring the mountain is to keep dodging falling rocks by correctly answering

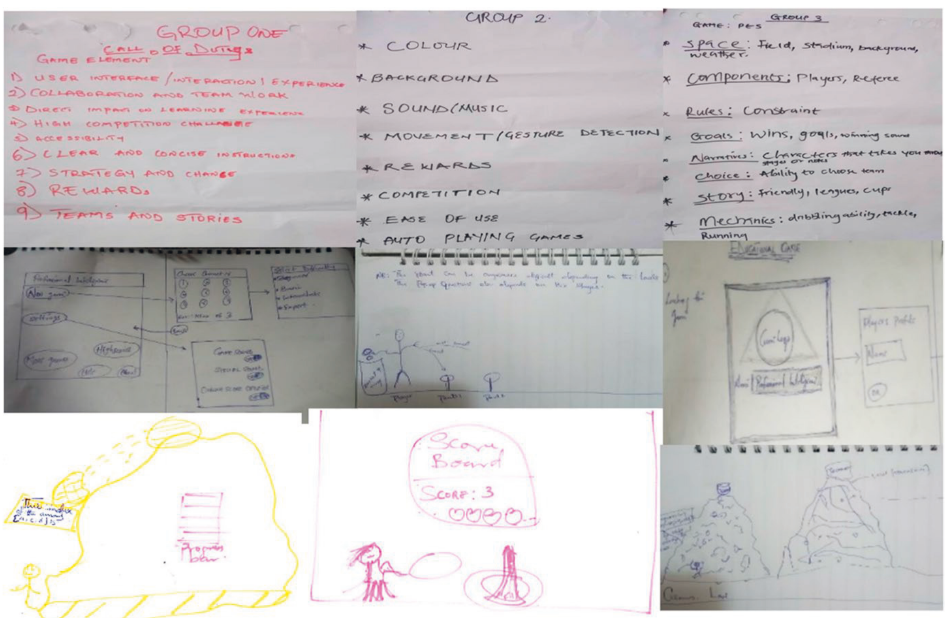


Fig. 7 Outputs from group tasks on selection of game elements and paper prototypes

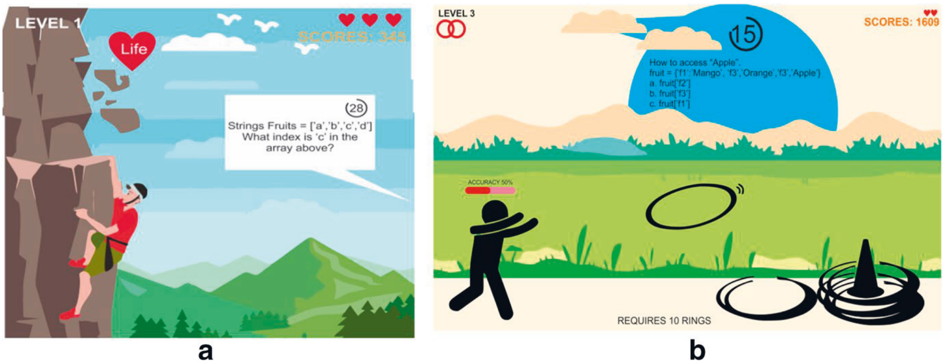


Fig. 8 Screenshots of mock-up prototype of co-designed mini games. A. the "*Mount Patti Treasure Hunt"* mini game; B. the "*Targeted Throws"* mini game

some CT puzzles. If players answer a puzzle incorrectly, they will be crushed by the falling rock and pushed down a step.

Another example of a contextual mini game conceived by the students is the "Targeted Throws" (Fig. 8b). The idea behind this mini game connects to the usual practice in the context where youth compete in harvesting fruit by throwing objects at ripe fruit, such as mangos, oranges, and apples. The player with the highest number of harvested fruits wins the game. Although the rules and constraints for playing this game were not stated by the students, they did connect their game to programming puzzles where a player earns a score for each correctly answered puzzle question, which implies a successfully harvested fruit.

4.2.2 Playtesting co-designed mini games, prototype evaluation, and feedback

The playtest process began by asking each group to submit their co-designed concepts to the researchers after three weeks of OCD activities. Their submissions included designs, game scenarios, puzzles, and prototypes. These submissions were first blinded (anonymized) and shared among the groups for playtesting and peer reviewing. In other words, the groups peer-reviewed each other based on the guidelines provided by the researchers regarding the goals for playtesting and peer reviewing. For example, the participants were asked to focus on the clarity of the game scenario, contextual storyline, precise information on rewarding the player, appropriate use of game elements, motivating features, and educational content embedded in the game. Some of playtesters' remarks are given in Table 5.

Feedback from the peer reviews provided insight into the depth of knowledge these participants obtained and their expectation from each other regarding the design of a contextual educational mini game. In addition, the students were able to learn from one another at different stages of the co-design process, particularly from the peer review process.

Table 5 Peer review of OCD prototype and evaluation by co-designers

Group number (Gn)	Peer reviewers	Sample of evaluation remarks by student reviewers
Gn1	Gn3	"I don't understand what you meant by the question point and the puzzle point." "…in the first game I did not notice any form of reward for the player." **[Gn3]**
Gn2	Gn1	"very good idea" "This is an incredibly brilliant game Idea. I love it." **[Gn1]**
Gn3	Gn2	"… not able to get the info on time" "The game was not well explained." "The diagrams were not explaining the game, rather explaining the work through the game." "The write-up was not properly structured…" **[Gn2]**

4.3 Students' experiences of OCD process for co-designing mini games

To answer the research question (RQ2) *"What are the experiences of students participating in an OCD of mini games to support their CT skills?"* we follow Moser and Korstjens' guidelines (2018) to present a content analysis of participants' responses to the interview conducted after completing the OCD process. The responses are the students' reflections regarding the knowledge gained and their experiences during and after the OCD process.

The students' reflections suggest that they gained several pieces of knowledge that made the exercise rewarding, although challenging, as claimed by some of the participants. For example, the following responses illustrate mixed feelings.

> the seminar was more of brainstorming especially the aspect of thinking of designing a game that helps one to understand programming… the experience was kind of challenging but was nice. **[P2]**
> **…**it's challenging to create games for education… **[P3]**
> My experience from the seminar was good. The seminar provided the opportunity for me to learn that games are not just all about the fun alone…but there's always educational knowledge that games usually pass onto the player…. **[P4]**

Regarding the expectation of the participants before participating in the OCD process and whether their expectations were met, their responses revealed that some participants did not know exactly what to expect but hoped to gain knowledge from the OCD process. For example, some participant responses were as follows.

> I don't actually know what to expect, but during the seminar, it was educative. I learned many things that I was not initially expecting. **[P1]**
> I thought that the seminar will be more of coding games, more of programming… but when we started, I discovered that it was more of games and

> co-design of games, then I felt excited since I do not know much of programming. **[P2]**
> Yes, my expectation for the seminar was to learn more about games because this was the idea the student coordinator gave to me…I thought we were going to be playing some games, but we ended up creating games. Its challenging to create games for education but I learned it during the activities in the seminar. **[P3]**

The interview specifically sought to know what learning objectives students gained from the OCD process. In other words, we asked the participants to describe a concrete take-home lesson from the OCD process. While some acquired collaborative skills, some mentioned learning about game elements and scenarios that seem unfamiliar to them before attending the seminar. Some responses from students in this regard are as follows.

> I learned that you can actually use games to teach many things because people learn differently and people learn faster using images and sounds, which is a better way to communicate certain concepts to students… this approach makes things stick faster in our minds and memory. **[P1]**
> I learned how to create smart learning system such as educational games that is easy to play and to teach students something. I also learned about elements of games… **[P2]**
> I learned about game scenarios, game elements, which I never knew, also, some platform used for creating games for different platforms **[P3]**
> Basically, I learned how to co-design games with fellow students…we brainstormed on ideas and collaborated in many ways to combine ideas for our games **[P4]**

The responses from interviewees shows that students had positively improved their co-design and collaborative skills after the OCD process. For example, one student asserted:

> even though some people in the group did not give their best input, which makes the workloads of group task to fall on a few… but it help me to learn a lot **[P4].**

Students generally expressed that the OCD process has positively affected their interest in educational games. Aside from the fun and excitement that students derive from games, they have been spurred through OCD process to design their own educational games. In fact, some anticipated designing educational games during their final year project.

5 Discussion

Previous studies have shown that co-designing digital mini games for educational purpose has proven to be a sound approach toward creating a learner-centered artefact (Havukainen et al., 2020). However, designing mini games through a co-design process is usually done in a face-to-face setting where the researcher meets with end users to participate in co-designing a product or artefact (Bonsignore et al., 2016). For example, Havukainen et al. (2020) recently explored a face-to-face approach for co-designing digital games with older adults and children. Similarly, Hjelmfors et al. (2018) co-designed with patients and health care professionals through a blended web-based and face-to-face approach to develop an intervention to improve communication health failure communication. This study designed and implemented an OCD process with CS students to create ideas, scenarios, and mini game prototypes that would later be developed into VR mini games to support CT and programming education. The study demonstrated how an OCD methodology was applied. The co-design process of mini games engaged twelve CS students at different levels of study in a Nigerian university. The commonality of the recruited students, i.e., they were all studying at the same university, helps make group collaboration easier. In addition, selecting students at different levels of study was a deliberate attempt to achieve an inclusive OCD process (Havukainen et al., 2020). Findings from this study are discussed in this section.

5.1 *How does OCD of contextual mini games with students in a Nigerian HEI work from the researchers' perspective?* (RQ1)

The descriptive analysis of the pre-OCD participation revealed several things. First, it was shown that most of the students who participated in the study had experience with online seminars and had even participated in an OCD activity before enrolling for this current study. This finding is surprising and makes the researchers wonder whether the students truly understood the meaning of the term "online co-design." The researchers had anticipated that since there a few OCD studies, most of the students would have little or no experience with it. It is possible that the students understood the term "online co-design" to mean "online collaboration," which probably is more familiar to them. This misunderstanding could be possible since their response was given before the actual seminar commenced. The researchers only provided a detailed explanation of the term "online co-design" and its objectives during the online seminar. The OCD objectives entail thinking, conceiving, and creating contextual mini game prototypes to assist students in acquiring CT skills.

As shown Section 4.1, Table 2, most of the students were used to playing games, and their experiences with different games were useful in making meaningful contributions toward co-designing contextual digital mini games. This finding aligned with Grover et al. (2020) where participants in the OCD for teacher's professional development were already familiar with the CS topic for which its curriculum was co-designed. In our study, the participants had the opportunity to brainstorm iteratively, discuss objectively, and negotiate their wish lists of game elements and features during the OCD process (Laine et al., 2020). The game elements wish list activity was initiated by the individual students and later extended to the group to allow for exhaustive deliberations on what users consider suitable for the mini games.

In addition, students were very eager to participate in the OCD process, as revealed by the results. Although their expectations before or during the OCD process were not explicitly known, their responses to the survey administered prior to the seminar showed that they had a strong motivation and were eager to participate in the OCD process. This result is expected since students voluntarily gave their consent to participate in the study and actively responded to our requests to participate in this study. While the students were hoping to learn new things during the OCD process, they did not expect to identify new areas in CS. This finding suggests that the students were probably familiar with CT topics and may have been taught the principles of CT in earlier completed courses.

Our investigation showed that students found "recursions" to be a very difficult programming topic to understand. This finding aligned with that of Anyango & Suleman (2018) who revealed that recursion and arrays are difficult programming topics for many novices. In addition, this finding provides useful insight in terms of supporting the authors' intention to provide SLEs to aid students in understanding CT concepts, including recursion, in the context of Nigeria. In other words, when designing a smart learning environment to teach basic CT principles to improve students understanding of programming topics, the authors would ensure that part of the learning objective would include teaching and learning about problem abstraction and recursion concepts, which is lacking in previous VR application to gain CT skills (Parmar et al., 2016).

In addition, the output from the OCD, as discussed in Section 4, shows that the students learned CT by thinking of conceptual and contextual game scenarios and stories. This method of teaching has been acknowledged to make learning through experience useful to students (Kolb, 2014). Besides, students used familiar stories within their context to create mini games (Eckardt & Robra-Bissantz, 2018). In addition, OCD activities, such as playtesting of co-designed mini games, prototype evaluation, and feedback, shows potential to allow students to gain creative and constructive ideas for designing educational mini games (Jong et al., 2010). Playtesting would ensure that issues arising from the designed prototype that did not fit the desire of end users could be discovered (Bonsignore et al., 2016). This discovery provides feedback to improve the mini games. In this study, the playtesting conducted at this stage of the design was minimal since the prototypes designed were still at a very low fidelity. Thus, the playtesting was intended to provide a general evaluation of what users perceive fits their expectations rather than to engage users in serious gameplay or deep interaction with game elements.

5.2 *What are the experiences of students participating in an OCD of mini games to support their CT skills?* (RQ2)

The study generally revealed that students gained CT skills during the OCD activities, even though they indicated that the process was challenging (Walsh et al., 2012). The students felt that the OCD activities were challenging because they made them "think" and develop a game scenario. For example, the two major outputs of the co-design process in the online environment shows how the students learned to conceptualize problems and connect them to contextual scenarios to create mini games. Therefore, the students were able to gain CT skills (problem identification, abstraction, and algorithmic thinking) through the OCD process, which is in line with the findings of a previous study (Malik et al., 2019).

Moreover, the result from the interviews conducted after the OCD process shows that the students gained new experience, such as game elements and how to connect those elements to create mini games. In addition, the students developed more interest in educational games and expressed their interest in developing mini games in their study projects. Different from the work of Grover et al. (2020), the findings from this study regarding its impact on the students' learning outcomes suggest that engaging students in designing something that is useful to their learning can improve learning achievement and provide inputs for creating a student-centered learning environment. Therefore, a student-centered OCD approach to developing educational tools to teach CS topics can improve students' learning experience than a teacher-centered OCD approach.

5.3 Lesson learned from co-designing mini games in an online environment in the Nigeria context

In this section, we discuss the authors' experiences, lessons learned, and provide recommendations for educational game designers and researchers adopting the OCD method. The methodology deployed in this study provided insights regarding the feasibility and suitability of an OCD process within the context of a developing country. Implementation of OCD is an important step toward creating an alternative co-design process for designers and researchers whose stakeholders are in "difficult to reach" locations owing to certain circumstances. Particularly in the African context, it could be assumed that OCD is rarely feasible considering infrastructure challenges, such as the cost of internet bandwidth, uncertain electricity supply to power the devices used for OCD, and limitations in terms of students' willingness to participate in week-long activities in the online environment. However, our experience shows that a methodological approach that is well-planned and defined can be suitable for such a context. Especially in this current era of efficient, easy to use, and even free online collaborative tools, such as Zoom, Google Meet, and WhatsApp, user-centered co-design processes can be conducted anywhere in the world.

While implementing the OCD methodology (Fig. 3), several noteworthy lessons were learned. These lessons could be useful to researchers, designers, educators, and other stakeholders interested in conducting a similar study in a contextual situation similar to this study. We discuss these lessons in five stages, which include (i) planning and engaging, (ii) exploring, (iii) designing, (iv) discussing and deciding, and

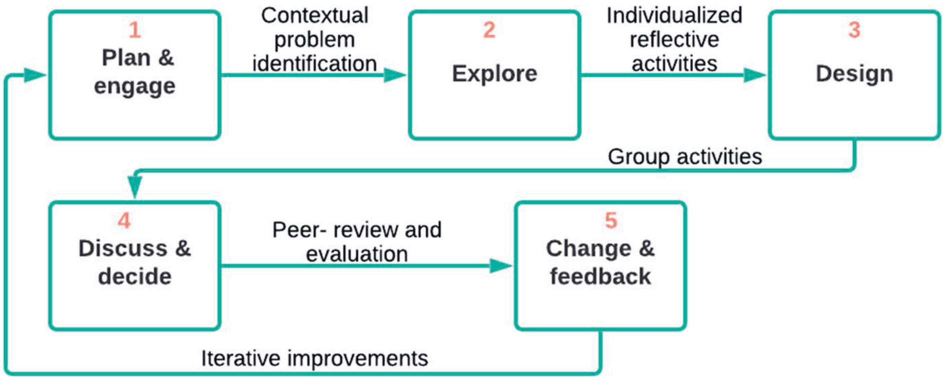


Fig. 9 Process flow of the five-stage implementation of OCD process

(v) changes and feedback stages. To better present these stages, we have provided Fig. 9 as a process flow and connects it to the lessons learned from the implementation of the OCD methodology. That is, the activities at each stage serve as input to the next stage. The stages presented in Fig. 9 is followed to explain researchers' experiences from implementation of the OCD process, lessons learned, and recommendations that may be useful to educational game designers and researchers who may adopt the OCD method in future research.

Planning and engaging stage From the researchers' experience, the applied OCD process was successful in the Nigerian context. However, problems regarding expensive internet bandwidth, inconsistent electricity supply, and lack of full commitment by some participants some of the contextual issues experienced. We mitigated these challenges by providing internet connectivity and electricity to motivate the participants, specifically during the online seminar. Hence, we recommend that during the planning and engaging stage, researchers should identify and recruit co-designers who are willing to fully participate in the OCD process. We also recommend that researchers should make provisions for basic facilities, such as internet connection that participants might need during the OCD implementation.

Exploring stage During the exploring stage, the basic equipment and technology required to conduct the OCD process were arranged, tested, and confirmed to function effectively. This initial confirmation was possible by running a test session between the OCD facilitators and selected participants using the Zoom platform. Although a few challenges that could interfere with online collaboration were identified at exploring stage, steps were address them before the main OCD process began. Therefore, we recommend that researchers should conduct an initial needs assessment to determine the type of equipment/platforms that work well (Walsh et al., 2012). For instance, the internet strength and the online communication platform must be simulated to ensure they are fit for OCD implementation.

Designing stage To make the designing stage a collaborative experience, the participants were grouped (Huizenga et al., 2019). Each group was limited to four members. Group activities within the designing stage included brainstorming on game elements wish list, ideation about contextual game scenarios and stories, creation of education mini games (puzzles), paper and mock-up designs, and presentation of concepts. Our experience shows that small groups can achieve quality collaboration. Inclusive collaboration can be enhanced if every member contributes to the group tasks. Hence, we recommend that researchers should begin by conducting a brief seminar where participants are introduced to the concept, goals, and objectives of the activities. Afterwards, participants should be grouped to allow for effective collaboration (Bonsignore et al., 2016).

Discussing and deciding stage During the discussing and deciding stage, co-designers presented their concepts at the group level. Each group evaluated another group's design and provided feedback based on the guideline that researchers had provided. The goal of the peer review is to obtain a users' perspective regarding what they considered suitable within the mini games by playtesting the paper prototype (Eckardt & Robra-Bissantz, 2018). Our experience shows that students learn from each other's concepts by providing comments (appraisals) based on their expectations of educational mini games. Therefore, it is recommended that OCD study allow co-designers to peer review themselves at group level based on their expectations (da Costa et al., 2017). This way, they could learn more from one another's ideas and contribute by presenting their individual point of view. We recommend that playtesting of prototypes should involve peer reviews.

Changes and feedback stage In the changing and feedbacking stage, comments from the co-designers (Table 5) formed part of the requirements for the ongoing second phase of the design process. Our experience shows that changes and feedback are an essential component of OCD where unforeseen situations may be accommodated at any stage of the OCD implementation. The OCD process allows for feedback at any stage. Although the output from one stage could serve as the input of the next stage, we recommend that implementation of the OCD process should be flexible enough to allow for scalability and changes that may arise.

5.4 Study limitations

This study was not without limitations. In terms of the procedure and methodology used, OCD did not give researchers the opportunity to completely monitor participants' activities remotely, which defeats the aim of mentorship and creates a barrier for a supervised collaborative design. Some potential solutions future studies could provide to alleviate this issue is to, for instance, ask the students to record all co-design activities or use a web-based co-design tool that automatically records all activities. The number of participants recruited for this study was small. This small sample size limits the extent to which our results can be generalized. While we tried to mitigate this limitation by adopting the interview method to gather more in-depth data, future studies should consider recruiting more students. In addition, issues of scalability might occur if the same OCD process is applied with more than 100 participants. Another limitation is the way playtesting of the designed mini games was conducted. Users only evaluated the low-fidelity paper and mock-up prototypes. They could not use the complete digital version of the prototype. This creates a lack of interaction with the game elements and features, which limits the user's experience.

6 Conclusion and future work

This study employed an innovative approach of OCD process to develop educational mini games through which students can gain CT skills, and the resulting prototypes are being developed into a VR based mini games environment to support the understanding of CT concepts. This intended VR application is aimed at supporting teaching and learning of CT at HEI in the context of Nigeria; however, it is envisaged to be relevant to students in all contexts since understanding of CT concepts, such as recursion, is still considered problematic for novices globally. The OCD process was developed to suit the situation where a face-to-face meeting between researchers and other stakeholders involved in co-designing artefacts became difficult. This study provided a thorough explanation of how the OCD process was followed to design contextual mini games prototypes (paper and mock-up based). Online platforms, such as the Zoom and WhatsApp, played the major role in implementing the OCD process. Analysis of participants' pre- and post-experience regarding seminar participation, co-design experience, gameplay, and games elements are presented in the study. One of the learning points for the researchers is how contextual games can emanate from the students being the co-designers and users of the mini games. This study highlighted five main stages for implementing the OCD process based on the researchers' experience. These stages are part of the researchers' contribution to the existing knowledge in terms of methodology and practice of co-designing mini games in online spaces.

This study reported part of the steps of authors' long-term plan of developing a learning environment to support CT skills in the context of Nigeria. The learning environment would provide a gamified approach for learning abstract programming concepts, such as recursion, through immersion in a VR environment. In addition, the study contributes to existing research in terms of co-designing educational mini games in several ways. First, the study established a theoretical foundation for designing a GBL application and connects these theories to guide the researchers' future study. Second, a methodological approach for conducting an OCD process was designed and implemented. Third, the results gave insight into how students conceived contextual mini games to solve CT and programming concepts. Fourth, the study outlined five stages of implementing the OCD process as a way of explaining the researchers' experiences and lessons learned. The recommendations provided in this study were based on these lessons and can serve as a guide to researchers who are carrying out similar studies in the future.

Future work will attempt to address the limitations highlighted earlier by investigating how to teach CT through the co-design process with a scaled number of participants in face-to-face and online settings. This future direction will provide the opportunity to conduct a comparative study to gain more insight into what works and what does not work within the context. Currently, we are finalizing the implementation of a VR application prototype, making it a collection of high-fidelity VR mini games to support students understanding of CT. A screenshot of this VR application of mini games is shown in the Appendix. Once finished, the authors would proceed to experiment and evaluate this prototype with CS students. In addition, the authors would try to investigate students' opinions about the perceived difficulty of CT and programming concepts after participating in the experiment.

Appendix

Screenshot of a VR-based *Mount Patti Treasure Hunt* mini game co-designed by the students. Left, the climbing progress made by the player and a falling stone that could crush the player. Right, a puzzle the player must solve to avoid being crushed by the falling stone and progress to the top of the mountain.

Questionnaire items administered prior to OCD activities.

Course of study:

Computer Science	☐	Computer Engineering	☐	Information Science	☐
Library Science	☐	Mathematics	☐	Others	☐

Questions on prior knowledge and experience in participating in seminars

S/N	Items
1	I have participated in an online seminar
2	I have participated in an online co-design process
	Questions on game and game design
3	I am an active player of games
4	I play game to gain more fun
5	I play game to gain new knowledge or learn new things
6	I have frequently participated in game design
7	I am familiar with the elements of game
8	I am aware of how game elements operate
	Questions on expectation from participating in the OCD process
9	I am eager in participating in the seminar because the purpose in the invitation notice was clear
10	I am eager in participating in the co-design process because I like to collaborate and share knowledge
11	I expect to learn new things in the co-design seminar
12	I am hoping that the co-design seminar will help me identify new areas of computer science
13	The seminar will provide me the opportunity to design my own game
	Question on ease/difficulty in understanding programming topics
14	In each of the programming topics, choose how easy or difficult they are based on your experience: 1. Programming Basics 2. Variables 3. If Statement 4. Loops 5. Functions 6. Methods 7. Recursions 8. File and Exception Handling

Interview instrument for participants of online co-design process

1. Tell me about your experience during the workshop.
2. What was your expectation before the workshop, and was this expectation met?
3. What is the objective of the workshop (what exactly did you learn)?
4. Is your knowledge of co-design improved in any way? How?
5. Tell me about your experience in co-designing within the group.
6. Did you contribute in any way? Share with me how you contributed, and lesson learned?
7. What other things have you gained from the co-design process with your colleagues, for example, brainstorming about the ideas on games, designs, and paper prototypes?
8. How has this exercise of the co-design process affected your interest in educational games?
9. How has your knowledge of VR improved?
10. Amongst the game elements, you have outlined from the group task, which is most important to you and why?

Funding This work was partly supported by the New Faculty Research Fund of Ajou University.

Data Availability Not applicable.

Code availability Not applicable.

Declarations

Competing interests The authors declare that they have no competing interests.

References

Adetunji, A. A., Bamidele, E. F., & Awodele, B. A. (2013). Effects of historical simulations as narrative and graphic advance organizers on Nigerian junior secondary school students' learning outcomes in basic science. *Mediterranean Journal of Social Sciences, 4*(2), 743–752.

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Adewumi, S. (2019a). A systematic review of computational thinking approach for programming education in higher education institutions. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1–10). ACM.

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Tukiainen, M. (2019b). Identifying potential design features of a smart learning environment for programming education in Nigeria. *International Journal of Learning Technology, 14*(4), 331–354.

Agbo, F., Oyelere, S., Suhonen, J., & M., T. (2019c). Smart learning environment for computing education: readiness for implementation in Nigeria. In *EdMedia+ Innovate Learning* (pp. 1364–1373). AACE.

Agbo, F. J., Oyelere, S. S., & Bouali, N. (2020a). A UML approach for designing a VR-based smart learning environment for programming education. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1–5). IEEE.

Agbo, F. J., Olawumi, O., Oyelere, S. S., Kolog, E. A., Olaleye, S. A., Agjei, R. O., Ukpabi, D. C., Yunusa, A. A., Gbadegeshin, S. A., Awoniyi, L., & Erinle, K. O. (2020b). Social media usage for computing education: The effect of tie strength and group communication on perceived learning outcome. *International Journal of Education and Development using Information and Communication Technology, 16*(1), 5–26.

Al-Azawi, R., Al-Faliti, F., & Al-Blushi, M. (2016). Educational gamification vs game based learning: Comparative study. *International Journal of Innovation, Management and Technology, 7*(4), 132–136.

Alamanda, D. T., Anggadwita, G., Ramdhani, A., Putri, M. K., & Susilawati, W. (2019). Kahoot!: A game-based learning tool as an effective medium to improve students' achievement in rural areas. In *Opening up education for inclusivity across digital economies and societies* (pp. 191–208). IGI Global.

Anyango, J. T., & Suleman, H. (2018). Teaching Programming in Kenya and South Africa: What is difficult and is it universal. *Proceedings of the 18th Koli Calling International Conference on Computing Education Research* (pp. 1–2). Koli: ACM.

Arnab, S., Walaszczyk, L., Lewis, M., Kernaghan-Andrews, S., Masters, A., Calderwood, J., Clarke, S. & Loizou, M. (2020). Designing Mini-Games for Micro-Learning: Open Educational Resources on Cultural Risks in Multi-Cultural Organisations. *ECGBL 2020 14th European Conference on Game-Based Learning* (p. 21). Academic Conferences limited.

Asal, V., Jahanbani, N., Lee, D., & Ren, J. (2018). Mini-games for teaching Political Science methodology. *Political Science & Politics, 51*(4), 838–841.

Bakker, M. (2014). *Using mini-games for learning multiplication and division: A longitudinal effect study*. Utrecht University.

Bassey, D., Mogaji, H., Dedeke, G., Akeredolu-Ale, B., Abe, E., Oluwole, A., Olagunju A., Chiedu F., & Ekpo, U. (2020). The impact of worms and ladders, an innovative health educational board game on soil-transmitted helminthiasis control in Abeokuta, Southwest Nigeria. *PLoS Neglected Tropical Diseases, 14*(9), 1–17.

Bonsignore, E., Hansen, D., Pellicone, A., Ahn, J., Kraus, K., Shumway, S., Kaczmarek, K., Parkin, J., Cardon, J., Sheets, J., & Holl-Jensen, J. (2016). Traversing transmedia together: Co-designing an educational alternate reality game for teens, with teens. *Proceedings of the 15th International Conference on Interaction Design and Children* (pp. 11–24). ACM.

Bouali, N., Nygren, E., Oyelere, S. S., Suhonen, J., & Cavalli-Sforza, V. (2019). Imikode: A VR Game to Introduce OOP Concepts. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1–2). ACM.

Bowen, S. (2010). Critical theory and participatory design. *Proceedings of CHI*, (pp. 10–15)

Bruck, P. A., Motiwalla, L., & Foerster, F. (2012). Mobile Learning with Micro-content: A Framework and Evaluation. *BLED 2012 Proceedings*. 25, pp. 527–543. AIS eLibrary.

Castro-Sánchez, E., Sood, A., Rawson, T. M., Firth, J., & Holmes, A. H. (2019). Forecasting implementation, adoption, and evaluation challenges for an electronic game–based antimicrobial stewardship intervention: Co-design workshop with multidisciplinary stakeholders. *Journal of Medical Internet Research*, 21(6), 1–10.

Cesário, V., Coelho, A., & Nisi, V. (2019). Co-designing gaming experiences for museums with teenagers. In A. Brooks, E. Brooks & C. Sylla C (Eds.), *Interactivity, game creation, design, learning, and innovation. ArtsIT 2018, DLI 2018. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* (Vol. 265). Springer. https://doi.org/10.1007/978-3-030-06134-0_5.

Ch'ng, S. I., Low, Y. C., Lee, Y. L., Chia, W. C., & Yeong, L. S. (2019). Video games: A potential vehicle for teaching computational thinking. In S. C. Kong & H. Abelson (Eds.), *Computational thinking education*. Springer. https://doi.org/10.1007/978-981-13-6528-7_14.

Chang, C. Y., & Hwang, G. J. (2019). Trends in digital game-based learning in the mobile era: A systematic review of journal publications from 2007 to 2016. *International Journal of Mobile Learning and Organisation, 13*(1), 68–90.

Chaves, E. M., Braga, P. B., Montenegro, Y. F., Rodrigues, V. B., Munguba, M. C., & de Albuquerque, V. H. (2019). Toc-Tum mini-games: An educational game accessible for deaf culture based on virtual reality. *Expert Systems*, e12470, 1-23.

Chittaro, L., & Buttussi, F. (2015). Assessing knowledge retention of an immersive serious game vs a traditional education. *IEEE Transactions on Visualization and Computer Graphics, 21*(4), 529–538.

da Costa, A. C., Rebelo, F., & Rodrigues, A. (2017). Co-designing a civic educational online game with children. *International Conference of Design, User Experience, and Usability* (pp. 377–386). Springer.

Devisch, O., Gugerell, K., Diephuis, J., Constantinescu, T., Ampatzidou, C., & Jauschneg, M. (2017). Mini is beautiful Playing serious mini-games to facilitate collective learning on complex urban processes. *Interaction Design and Architecture (s) Journal, 35*, 141–157.

Dias, P., Silva, R., Amorim, P., Lains, J., Roque, E., Serôdio, I., & Potel, M. (2019). Using virtual reality to increase motivation in poststroke rehabilitation. *IEEE Computer Graphics and Applications, 39*(1), 64–70.

Dix, S., Ferguson, G., Son, J., Sadachar, A., Manchiraju, S., Fiore, A., & Niehm, L. (2012). Consumer adoption of online collaborative customer co - design. *Journal of Research in Interactive Marketing, 6*(3), 180–197.

Eckardt, L., & Robra-Bissantz, S. (2018). Playtesting for a better gaming experience: importance of an iterative design process for educational games. *Twenty-Second Pacific Asia Conference on Information Systems.* Japan.

Eguchi, A. (2016). Computational thinking with educational robotics. *Society for Information Technology & Teacher Education International Conference* (pp. 79–84). Association for the Advancement of Computing in Education (AACE).

E-lineMedia. (2011). Gamestar mechanic learning guide. Retrieved 10 05, 2020, from https://sites.google.com/a/elinemedia.com/gsmlearningguide/game-design-101/lesson-2-core-design-elements

Emembolu, I. C., Emembolu, C., Umechukwu, K., Sulaiman, A., & Aderinwale, O. (2019). Building a community of STEM educators in Nigeria using the TeachAKid2Code program. In International Conference on Interactive Collaborative Learning (pp. 536–545). Springer.

Fisher, J. (2020). Digital games, developing democracies, and civic engagement: a study of games in Kenya and Nigeria. *Media, Culture & Society, 42*(7), 1309–1325.

Freina, L., & Ott, M. (2015). A literature review on immersive virtual reality in education state of the art and perspectives. *The international scientific conference e-learning and software for education, 1–8*, 10.

Friedrich, P. (2013). *Web-based co-design: Social media tools to enhance user-centred design and innovation processes*. VTT Science.

Gerling, K., & Masuch, M. (2011). When gaming is not suitable for everyone: playtesting wii games with frail elderly. *1st Workshop on Game Accessibility. Gerling, K.*, & Masuch, M.: ACM.

Gomez, K., Kyza, E. A., & Mancevice, N. (2018). Participatory design and the learning Sciences (1st ed.). In *International handbook of the learning sciences*. Routledge.
Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43.
Grover, S., Cateté, V., Barnes, T., Hill, M., Ledeczi, A., & Broll, B. (2020). FIRST principles to design for online, synchronous high school CS teacher training and curriculum co-design. *In Koli Calling'20: Proceedings of the 20th Koli Calling International Conference on Computing Education Research* (pp. 1–5). ACM.
Havukainen, M., Laine, T. H., Martikainen, T., & Sutinen, E. (2020). A case study on co-designing digital games with older adults and children: game elements, assets, and challenges. *The Computer Games Journal, 9*, 163–188.
Hickman, L., & Akdere, M. (2018). Developing intercultural competencies through virtual reality: Internet of Things applications in education and learning. *15th learning and technology conference (L&T)* (pp. 24–28). IEEE.
Hjelmfors, L., Strömberg, A., Friedrichsen, M., Sandgren, A., Mårtensson, J., & Jaarsma, T. (2018). Using co-design to develop an intervention to improve communication about the heart failure trajectory and end-of-life care. *BMC Palliative Care, 17*(1), 85.
Huizenga, J., Admiraal, W., Ten Dam, G., & Voogt, J. (2019). Mobile game-based learning in secondary education: Students' immersion, game activities, team performance and learning outcomes. *Computers in Human Behavior, 99*, 137–143.
Jong, M. S., Shang, J., Lee, F. L., & Lee, J. H. (2010). VISOLE: A constructivist pedagogical approach to game-based learning. Collective intelligence and e-learning 2.0: *Implications of web-based communities and networking* (pp. 185–20). IGI Global.
Koivisto, J. M., Niemi, H., Multisilta, J., & Eriksson, E. (2017). Nursing students' experiential learning processes using an online 3D simulation game. *Education and Information Technologies, 22*(1), 383–398.
Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. FT press.
Kommers, P. (2003). Experiential learning through constructivist learning tools. *International Journal of Computers and Applications, 25*(1), 72–83.
Korkmaz, Ö., Cakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior, 72*, 558–569.
Laine, T., Normark, J., Lindvall, H., Lindqvist, A. K., & Rutberg, S. (2020). A distributed multiplayer game to promote active transport at workplaces: User-centred design, implementation and lessons learned. *IEEE Transactions on Games, 12*(4), 386–397
Lindblom, A., Laine, T. H. and Souza Rossi, H. (2021). Investigating network performance of a multiuser virtual reality environment for mining education. *Proceedings of the International Conference on Ubiquitous Information Management and Communication*, Seoul, Republic of Korea.
Loos, E., de la Hera, T., Simons, M., & Gevers, D. (2019). Setting up and conducting the co-design of an intergenerational digital game: A state-of-the-art literature review. *International Conference on Human-Computer Interaction* (pp. 56–69). Springer.
Malik, S. I., Shakir, M., Eldow, A., & Ashfaque, M. W. (2019). Promoting algorithmic thinking in an introductory programming course. *International Journal of Emerging Technologies in Learning (iJET), 14*(1), 84–94.
Mariager, C. S., Fischer, D. K., Kristiansen, J., & Rehm, M. (2019). Co-designing and field-testing adaptable robots for triggering positive social interactions for adolescents with cerebral palsy. *28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)* (pp. 1–6). IEEE.
Marín, V., Asensio-Pérez, J., Villagrá-Sobrino, S., Hernández-Leo, D., & García-Sastre, S. (2018). Supporting online collaborative design for teacher professional development. *Technology, Pedagogy and Education, 27*(5), 571–587.
Mathew, R., Malik, S. I., & Tawafak, R. M. (2019). Teaching problem solving skills using an educational game in a computer programming course. *Informatics in Education, 18*(2), 359–373.
Moser, A., & Korstjens, I. (2018). Series: Practical guidance to qualitative research. Part 3: Sampling, data collection and analysis. *European Journal of General Practice, 24*(1), 9–18.
Mubin, O., Novoa, M., Forrester, J., & Iqbal, R. (2016). GiggleBat: Enhancing playing and outdoor culture in Australian children. *Proceedings of the 30th International BCS Human Computer Interaction Conference*, (pp. 1–10).

Natow, R. S. (2020). The use of triangulation in qualitative studies employing elite interviews. *Qualitative Research, 20*(2), 160–173.
Ogunsile, S. E., & Ogundele, B. O. (2016). Effect of game-enhanced nutrition education on knowledge, attitude and practice of healthy eating among adolescents in Ibadan, Nigeria. *International Journal of Health Promotion and Education, 54*(5), 207–216.
Oladipo, F. O., & Ibrahim, M. A. (2018). The CodeEazee tool support for computational thinking in python. *European Journal of Engineering and Technology Research, 3*(3), 12–20.
Oladipo, F. O., Ibrahim, M. A., Obansa, A. A. U., & Jatto, A. A. (2017). FULangS: A capstone scripting tool. *International Journal of Engineering and Applied Sciences, 4*(9), 44–50.
Oyelere, S. (2018). *Design and development of a mobile learning system for computer science education in Nigerian higher education context*. Itä-Suomen yliopisto.
Oyelere, S. S., Suhonen, J., Wajiga, G. M., & Sutinen, E. (2018). Design, development, and evaluation of a mobile learning application for computing education. *Education and Information Technologies, 23*(1), 467–495.
Oyelere, S. S., Agbo, F. J., Sanusi, I. T., Yunusa, A. A., & Sunday, K. (2019). Impact of puzzle-based learning technique for programming education in Nigeria context. In *19th International Conference on Advanced Learning Technologies (ICALT)* (2161, pp. 239–241). IEEE.
Parmar, D., Isaac, J., Babu, S., D'Souza, N., Leonard, A., Jörg, S., Gundersen, K., & Daily, S. (2016). Programming moves: Design and evaluation of applying embodied interaction in virtual environments to enhance computational thinking in middle school students. *IEEE Virtual Reality (VR)* (pp. 131–140). IEEE.
Parong, J., & Mayer, R. E. (2020). Cognitive consequences of playing brain-training games in immersive virtual reality. *Applied Cognitive Psychology, 31*(1), 29–36.
Pérez-Sanagustín, M., Hernández-Leo, D., Santos, P., Kloos, C. D., & Blat, J. (2014). Augmenting reality and formality of informal and nonformal settings to enhance blended learning. *EEE Transactions on Learning Technologies, 7*(2), 118–131.
Perini, S., Oliveira, M., Margoudi, M., & Taisch, M. (2018). The use of digital game based learning in manufacturing education–a case study. *International Conference on Learning and Collaboration Technologies* (pp. 185–199). Springer.
Qian, M., & Clark, K. R. (2016). Game-based learning and 21st century skills: A review of recent research. *Computers in Human Behavior, 63*, 50–58.
Radianti, J., Majchrzak, T. A., Fromm, J., & Wohlgenannt, I. (2020). A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education, 147*, 103778.
Regal, G., Sellitsch, D., Kriglstein, S., Kollienz, S., & Tscheligi, M. (2020). Be active! participatory design of accessible movement-based games. *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*, (pp. 179–192).
Ramin, N., Talib, C. A., & Aliyu, H. (2020). Science teacher's self-confidence on integrating computational thinking into classroom pedagogies for teaching and learning. *Learning Science and Mathematics, 15*, 56–70.
Rinnert, G. C., Martens, M., Campana, K., & Kociubuk, J. (2019). Playing with handwriting: creating a game-based app for learning cursive. *European Conference on Games Based Learning* (pp. 569–570). Academic Conferences International Limited.
Robertson, T., & Simonsen, J. (2012). *Routledge international handbook of participatory design* (Simonsen, J., & Robertson, T. ed.). Routledge.
Rosenzweig, E. (2015). *Successful user experience: Strategies and roadmaps* (Morgan Kaufmann ed.). Elsevier.
Ruiz, A., Giraldo, W. J., & Arciniegas, J. L. (2018). Participatory design method: Co-creating user interfaces for an educational interactive system. *International Conference on Human Computer Interaction* (pp. 1–8). ACM.
Spencer, T., Rademaker, L., Williams, P., & Loubier, C. (2019). *Online, Asynchronous Data Collection in Qualitative Research*. The Oxford Handbook of Methods for Public Scholarship.
Spinuzzi, C. (2015). The methodology of participatory design. *Technical Communication, 52*(2), 163–174.
Sunday, K., Ocheja, P., Hussain, S., Oyelere, S., Samson, B., & Agbo, F. (2020). Analyzing student performance in programming education using classification techniques. *International Journal of Emerging Technologies in Learning (iJET), 15*(2), 127–144.

Talib, C. A., Aliyu, H., Zawadzki, R., & Ali, M. (2019). Developing student's computational thinking through graphic calculator in STEAM education. *In AIP Conference Proceedings (*pp. 030003). AIP Publishing LLC.
Thabrew, H., Fleming, T., Hetrick, S., & Merry, S. (2018). Co-design of eHealth interventions with children and young people. *Frontiers in psychiatry, 9*, 481.
Tobar-Muñoz, H., Baldiris, S., & Fabregat, R. (2016). Co design of augmented reality game-based learning games with teachers using co-CreaARGBL method. *16th International Conference on Advanced Learning Technologies (ICALT)* (pp. 120–122). IEEE.
Toivonen, T., Jormanainen, I., Kahila, J., Tedre, M., Valtonen, T., & Vartiainen, H. (2020). Co-designing machine learning apps in K–12 with primary school children. *In 2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT)* (pp. 308–310). IEEE.
Tokac, U., Novak, E., & Thompson, C. G. (2019). Effects of game-based learning on students' Mathematics achievement: A meta-analysis. *Journal of Computer Assisted Learning, 35*(3), 407–420.
Vetere, F. D. (2009). The magic box and collage: Responding to the challenge of distributed intergenerational play. *International Journal of Human-Computer Studies, 67*(2), 165–178.
Virvou, M., & Katsionis, G. (2008). On the usability and likeability of virtual reality games for education: The case of VR-ENGAGE. *Computers & Education, 50*(1), 154–178.
Walsh, G., Druin, A., Guha, M. L., Bonsignore, E., Foss, E., Yip, J. C., Golub, E., Clegg, T., Brown, Q., Brewer, R., & Joshi, A. (2012). DisCo: a co-design online tool for asynchronous distributed child and adult design partners. *Proceedings of the 11th International Conference on Interaction Design and Children* (pp. 11–19). ACM.
Walsh, G., Druin, A., Guha, M. L., Foss, E., Golub, E., Hatley, L., Bonsignore, E., & Franckel, S. (2010). Layered elaboration: a new technique for co-design with children. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1237–1240). ACM.
Wu, W. H., Hsiao, H. C., Wu, P. L., Lin, C. H., & Huang, S. H. (2012). Investigating the learning-theory foundations of game-based learning: a meta-analysis. *Journal of Computer Assisted Learning, 28*(3), 265–279.
Zhou, Y., Ji, S., Xu, T., & Wang, Z. (2018). Promoting knowledge construction: A model for using virtual reality interaction to enhance learning. *Procedia Computer Science, 130*, 239–246.

Authors and Affiliations

Friday Joseph Agbo[1] · Solomon Sunday Oyelere[2] · Jarkko Suhonen[1] · Teemu H. Laine[3]

Friday Joseph Agbo
friday.agbo@uef.fi

Solomon Sunday Oyelere
solomon.oyelere@ltu.se

Jarkko Suhonen
jarkko.suhonen@uef.fi

[1] School of Computing, University of Eastern Finland, P.O. Box 111, FIN-80101 Joensuu, Finland

[2] Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, SE-93177 Skellefteå, Sweden

[3] Department of Digital Media, Ajou University, 16499 Suwon, Republic of Korea

ARTICLE VII

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Tukiainen, M. (2021). Design, development, and evaluation of a virtual reality game-based application to support computational thinking. *Submitted to Educational Technology Research and Development*.

FRIDAY JOSEPH AGBO

Due to how portable and accessible technology has become, developing smart learning environments with advanced technology such as virtual reality can mediate teaching and learning in several ways. To develop a state-of-the-art smart learning environment that facilitates 21st-century skills, including computational thinking and problem-solving, this dissertation co-designed a game-based virtual reality application with students from sub-Saharan Africa to enhance their learning experience. The virtual reality application aims to support novices with no programming experience.

UNIVERSITY OF
EASTERN FINLAND

uef.fi

**PUBLICATIONS OF
THE UNIVERSITY OF EASTERN FINLAND**
Dissertations in Forestry and Natural Sciences

ISBN 978-952-61-4468-9
ISSN 1798-5668