Integrating Parson's Programming Puzzles into a game-based mobile learning application

Solomon Sunday Oyelere, Jarkko Suhonen
School of Computing, University of Eastern Finland
Joensuu, Finland
{solomon.oyelere, jarkko.suhonen}@uef.fi

Teemu H. Laine
Department of Computer Science, Electrical and Space-Engineering, Luleå University of Technology,
Skellefteå, Sweden
teemu@ubilife.net

ABSTRACT

Understanding of elementary programming concepts, logic, and syntax is a vital part of learning to program. Unfortunately, learning programming is found to be difficult and boring, especially for novices. For example, drill type of exercises designed for learning elementary programming can be very tedious, making it hard to motivate learners. This study focused on the development of a mobile application, which integrates puzzle-based Parson's programming exercises into a strategy board game with the aim of facilitating the teaching and learning of programming. With the mobile application, learners play Ayo, a variant of the famous traditional African strategy board game Mancala. In each round of the game, the learners will solve a Parson's puzzle, which consist of small programming tasks where students are required to build programs by drag and drop, selection, indenting and ordering code fragments. The proposed solution of integrating the Parson's puzzles to Ayo game provides a new perspective on how to use mobile devices in programming education. The long-term aim of our work is to create a framework for integrating board games into computing education. The study reported in the article is the first step towards creating the framework.

CCS Concepts

•Social and professional topics → CS1;

Keywords

Parson's programming puzzles; Mobile learning; MobileEdu; Mobile learning application; Game-based learning; Board game; Computing education

1 INTRODUCTION

The use of computing related solutions in our daily activities has ubiquitous impact on computing education. The computing industry is increasingly demanding problem solving, analytical, logic, and programming skills from graduates. There is a world-wide trend on teaching computer science starting from the elementary school onwards, and this may affect tertiary education at some point [1]. In order to develop relevant skills, computing courses are designed with various hands-on features for students to practice. Unfortunately, novice learners find it hard to convert theoretical ideas into practical solutions. For example, according to Robins et al. [2], novice programmers are likely to concentrate only on shallow knowledge, lack conceptual modelling skill, and fail to incorporate pertinent knowledge required for solving problem. Novice programmers find it difficult to remember and correctly apply programming language vocabulary, logic, errors, syntax, semantics, and styles.

Particularly, learners' motivation and interest in programming course are dampen because the course is boring and unexciting [3]. One approach to improve learners' motivation, interest, and engagement in practicing basic programming skills in an entertaining way is the Parson's programming puzzles [4]. In Parson's problems, learners are given code fragments as exercise type and are required to reconstruct the program by ordering and selection. Parson's puzzles were originally designed to support learning of programming syntax and logic through repetition, and have been similarly applied to teach problem solving through algorithmic thinking [5] combined with visualization [6], and used to construct examination questions [7], [8], [9]. Another approach to address learners' motivation by making learning to program more exciting and engaging is use of games [3]. For instance, board games are considered simple to grasp, and interesting to motivate learning [13].

Quite recently, considerable attention has been given to the use of mobile touch screen devices to support learning. Previous studies indicate that these devices are suitable for both solving Parson's problems, Karavirta et al. [10] and learning to program through games, Hamid and Leong [11], Vinay et al. [12]. However, previous studies do not consider the potentials of integrating the Parson's puzzles to existing strategy board games, as a tool to support interactive learning, engagement, and motivation to novice programmers learning through mobile devices. The rules of strategy board games are typically simple and motivating for students [13], thus, suitable for integration into Parson puzzles on mobile touch devices.

Though the efficiency of building mobile learning application for Parson's puzzles has been improved in recent years, most of these improvements focused on providing automatic feedback mechanisms and integrating program visualizations [4], [6]. However, it is possible to improve its application further by developing novel approaches for incorporating Parson's problems into a game-based mobile learning environment. With this goal in mind, this work develops a new concept of deriving interaction, engagement, and motivation into learning to program by novices. Considering the challenges that programming pose to students, it is important to implement tools that will support novices to master programming concepts.

The article describes the integration process of Parson's programming puzzles into an African board game by presenting the design and development process of mobile learning application MobileEdu, the research framework used for building the new mobile learning application, followed by presenting the strategy board game under study and showing its integration process with Parson's puzzles. Finally, the article concludes with presentation of evaluation plans, and future direction towards additional contributions in our continuing quest for innovative learning platforms to aid novice programmers.

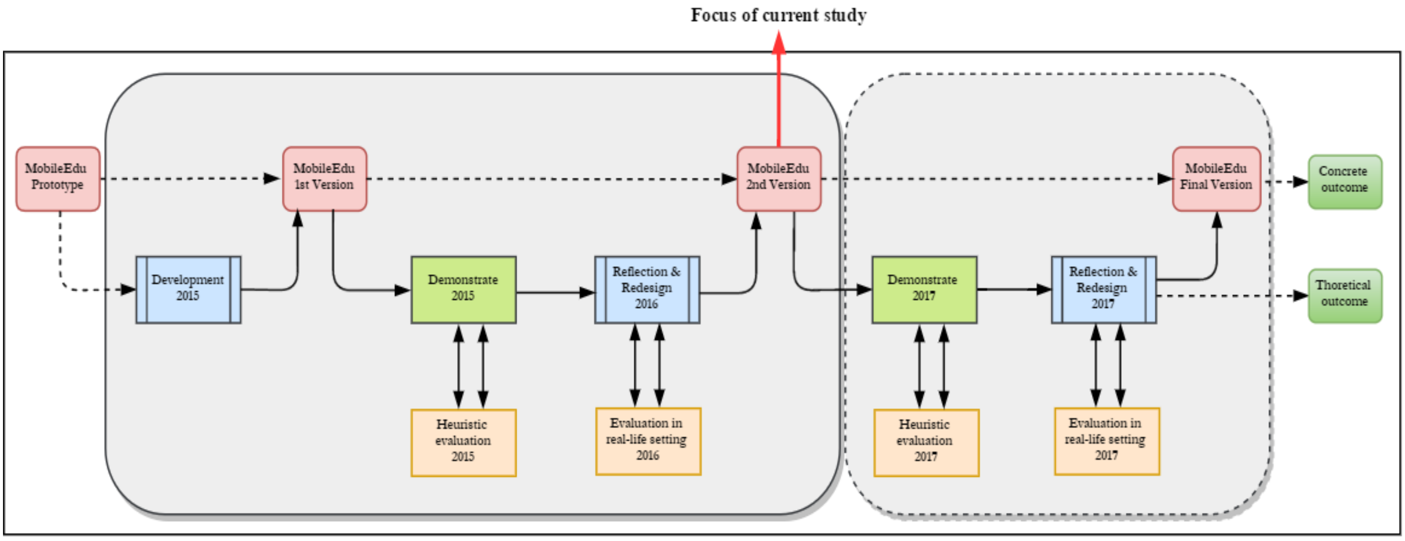


Figure 2: DSR development process of MobileEdu.

2 INTEGRATION OF PARSON’S PROGRAMMING PUZZLES INTO AN AFRICAN BOARD GAME AYO

In this section, we first introduce a mobile learning application MobileEdu and briefly summarize its development process. We then describe how Ayo game has been implemented in the MobileEdu application. Finally, we present how Parson’s programming puzzles are integrated into the mobile version of the Ayo board game.

2.1 MobileEdu – a mobile learning application for computing education

MobileEdu is an Android-based mobile learning application developed to support the learning of computing courses on mobile devices in Nigerian higher education context. It consists of seven tabs to support various learning functions such as easy access to course materials, quiz, messaging, teamwork, collaboration, receiving course updates and information [14]. For example, the group tab on the application is presented in Fig. 1. In the group tab, learners are placed into groups to complete tasks together to support teamwork. Further details about MobileEdu are presented in an earlier study [14]. The design process of MobileEdu is based on the d*esign science research* (DSR) framework proposed by Johannesson and Perjons [15]. DSR has been widely applied in information systems and educational technology research [15-16], and the method is particularly suitable for mobile learning [17]. DSR is a holistic research approach with dual aims. First, a concrete solution (artifact) is developed to solve a practical problem, fulfill a need, or requirement related to a real-life context. Second, DSR expands the existing knowledge base related to the research area.

So far, the MobileEdu development and research process comprised two iterations and one evaluation [14]. In consonance with DSR framework, Johannesson and Perjons [15], MobileEdu evolved through various versions in iterative phases of problem explication, outlining an artifact, defining requirements, design and development of the artifact, demonstration of the artifact and evaluation of the artifact. After rigorous explication of problem and requirement definition, a prototype app was developed.

Detailed activities, application, evaluation, and results of the evaluation are presented in [14]. Further assessment by three expert evaluators and opinions received from students during interviews in the first evaluation led to second iteration and redesign. The two functional versions of MobileEdu over the period of 34 months constitute concrete outcome of the research work. Fig. 2 represents DSR development process of MobileEdu as consecutive enhancements in usability and functionality based on responses from evaluations that led to further reflection and redesign. In the future, there are plans to undertake heuristic and concrete settings’ evaluations before final version of MobileEdu is released.

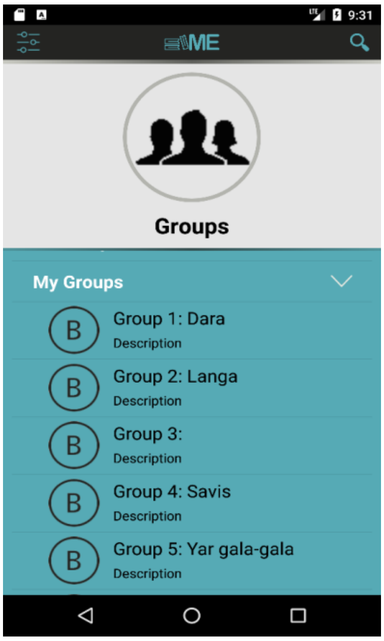


Figure 1: Group Tab from MobileEdu app.

2.2 Ayo strategy board game implementation in MobileEdu

The latest version of MobileEdu consists of a new tab - “Game” (Fig. 3). The Game tab implements an *Ayo* game (in Yoruba language of Southwest Nigeria) from *Mancala* family. Ayo is an ancient game with over 3000 years of history and originated from Africa [13]. It has different names and methods of play across Africa. *Ayo* is a two-player game and played on a board with 12 bins arranged in array-like rows facing each other. A larger bin (store or home bin) containing captured seeds is located at the end of each player’s row. The objective of the play is to capture all the bins on the board by capturing as many seeds as possible in each round. The game begins with a predefined number of seeds (usually 4 or 6, depending on variant) in each player’s bins, and empty store bin (Fig. 4).

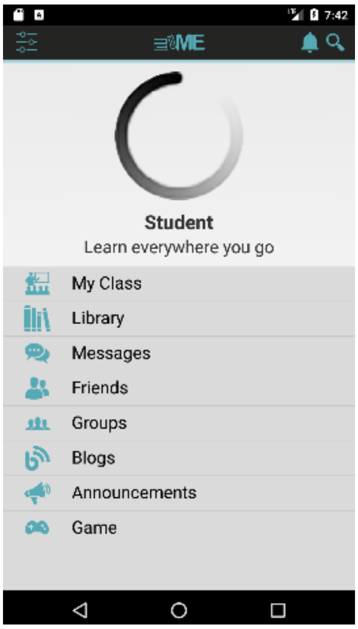


Figure 3: Home screen of the latest version of MobileEdu showing Game tab.

2.2.1 *Rules, strategies, and game play.* A player selects one of his non-empty bins, evacuates the seeds from that bin, and dispenses one seed in each bin at a time in counter-clockwise direction until all seeds are dispensed. If the last seed is dispensed into an empty bin, the player’s turn ends. If a player dropped the last seed in a non-empty bin, he collects the seeds and continues dispensing. A player captures and places seeds into his store bin, if the bin visited contain 4 seeds and reside in his row. However, if a player makes 4 seeds in one of his opponent’s row, the player loses the seeds. Therefore, when making a move, players calculate and strategically avoid making 4 seeds in the opponent’s row. Nevertheless, if the player makes 4 seeds in the opponent’s row and that is the last seed in the player’s hand, the player get the 4 seeds instead of losing them. The round ends when one player has no more seeds in his row and game ends when a player capture all the opponent’s bin row. There are several variations of these rules in *Mancala* family of games, and several strategies to win the game. The most commonly used strategies are *Kroo building* or *offensive*, *overloading*, *defensive*, *pressure*, *attack*, and *counter-attack*. Detailed explanation and application of these strategies are given in [18].

The mobile touchscreen version of Ayo consists of four play modes (Fig. 5). A player has the option of playing against the computer (one player), playing against another player on the same device (two player), playing against another player over Bluetooth (play Bluetooth) and playing against another player online (play online). The play Bluetooth option is contextually relevant to developing countries such as Nigeria that is struggling with poor Wi-Fi and networking infrastructure [19]. Two players who want to play via a Bluetooth or an online connection must mutually agree on the host and guest before setting up the connection (Fig. 6 shows the Bluetooth game setup). In our implementation, the game interface consists of player names, number of seeds captured by each player, number of bins captured by each player and seed sowing counter (decreases as seed dropping continues on each bin). The game board showing all game features during two player game session is presented in Fig. 7.

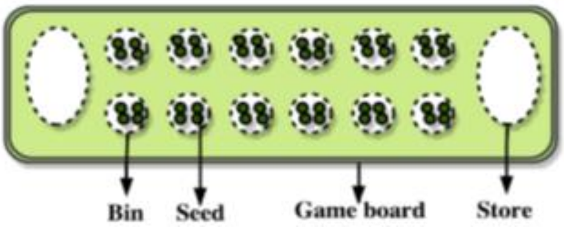


Figure 4: Ayo board in preliminary position having 4 seeds in each of the 6 array-like bins and 0 seed in the players store bin.

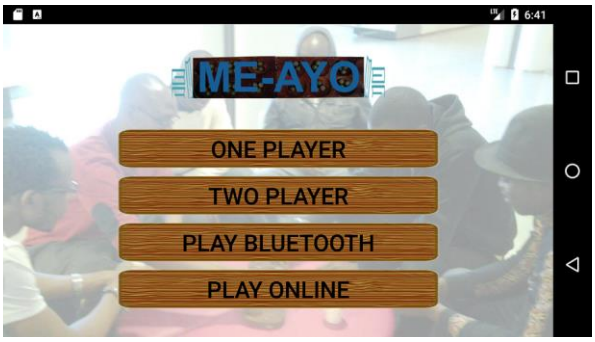
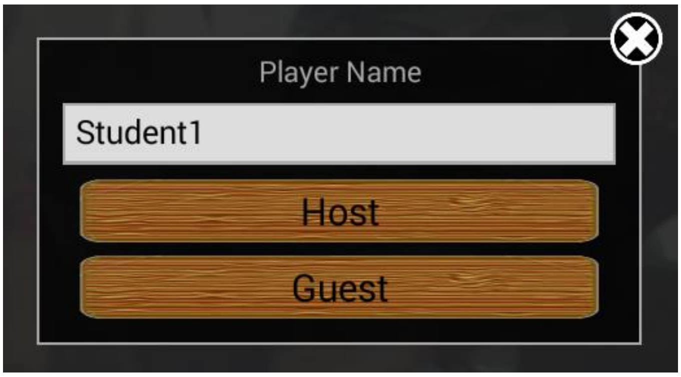


Figure 5: Play mode selection screen.



Player Name

Student1

Figure 6: Bluetooth game setup screen.

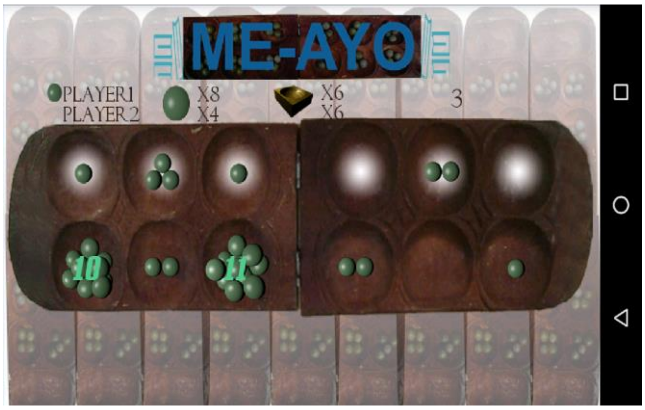
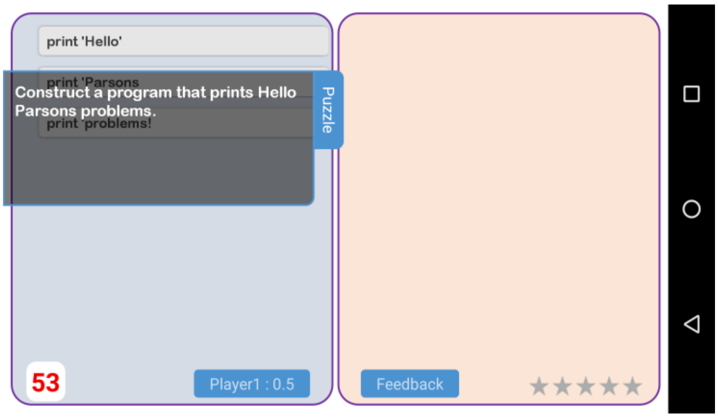


Figure 7: Game board showing two-player game session.



```
print 'Hello'
print 'Parsons
print 'problems!
```

Construct a program that prints Hello Parsons problems.

Figure 8: Puzzle label is shown containing task instruction.

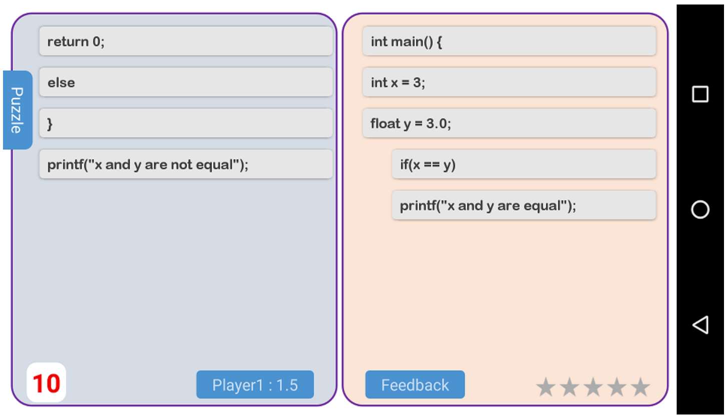
3 INTEGRATION OF PARSON'S PROGRAMMING PUZZLES INTO THE AYO BOARD GAME

The primary idea is to teach basic programming concepts to novices with Ayo game. Ayo game programming puzzles are a type of scaffold program construction tasks where the learner is given set of code fragments, blocks of a single or multiple lines of code, multiple-choice question-type in the form of Parson's puzzles. The task of the student is to construct a working program from these skeleton codes thereby gaining understanding and comprehension of programming logic. The goal is for the student to read and understand existing program code, deduce logic, and determine how to solve the problem. Parson's puzzle interface on our application is modelled similarly to previous solution [10].

The interface is made up of two sections. On the left side of the screen is problem area, and the right side consists of solution area. At each round of Ayo game play, a player is prompted with puzzle task and a timer. On the puzzle interface is a toggle-able label named "*Puzzle*," containing task instructions (Fig. 8). *Puzzle* is situated on left side of the screen and originally visible once a new task is displayed. With a tap on *Puzzle*, the task instruction is rendered either visible or invisible (Fig. 8).

The problem area (left side) consists of a small display countdown timer. When the time is elapsed without solving the problem, the player loses his chance and gets no points. The next round of Ayo game play commences automatically after each puzzle prompt. The solution area (right side) consists of *feedback* button and 5 stars. Linear-type of feedback is displayed for each turn of puzzle.

A player that solves a puzzle correctly is awarded 0.5 points. Points are displayed along with the player name besides the timer. One star is awarded to a player that accumulated 4 points in a puzzle round. In the future implementation, these points could be associated to seeds captured on Ayo game play. Parson's puzzles presented to novices in this application focused mainly on a language-independent drag and drop of program codes, multiple-selection options, code indentation and ordering of program codes. Fig. 9 shows drag and drop and indentation puzzle.



```
return 0;
else
}
printf("x and y are not equal");
```

```
int main() {
int x = 3;
float y = 3.0;
    if(x == y)
    printf("x and y are equal");
```

Figure 9: Drag and drop puzzle and indentation puzzle.

4 CONCLUSION AND FUTURE WORK

This study offered the expansion of MobileEdu mobile learning system to include a traditional African strategy board game Ayo with Parson's puzzles as a possible solution to the problem of decrease in interaction, motivation, and engagement during the process of learning to program. Future planned evaluation will determine whether the tool meets users' expectations and supports learning of programming on mobile devices. If the integration of Parson's puzzles into strategy board game were shown to be successful through the evaluation, MobileEdu would be used as an all-inclusive mobile learning environment for computing education. Similarly, since the overall aim of the study is to invent a framework on how to integrate board games into computing education, future work will address integration of other board games into the study.

Motivated and engaged instruction is a sought after part of any teaching and learning environment; lectures are intended to explicate abstract ideas, which are accompanied with practical activities. However, instructional environments could be made more engaging and attractive if hands-on skills, such as apprenticeship, problem-solving, and indigenous knowledge, could be taught through both face-face instruction and game-based tools for instance, on Ayo game.

Some suggestions on how the integration of board games to computing education should be done are as follows: identify the students' needs, particularly taking into account of the local practices and contexts; creating a brainstorming sessions to unravel the pedagogical attributes which are inherent in the indigenous artefacts and mapping those to computing knowledge; crafting a tool to mediate between learning contents, conceptual understanding and learning-by-doing; and finally, inventing a culturally responsive learning environment to support computer science education through board games which is largely related to ethnocomputing [20]. The computing education framework that will be developed will be tested further with other indigenous related board games such as *Afrikan tähti* (Finnish) meaning "the star of Africa", which has its origin from Finland and designed based on a story about Africa.

The main limitation of the study is an acknowledgment that playing African game might not be relevant and interesting for all learners. Furthermore, we cannot claim that the game would provide fun aspect to all learners and we cannot assume with certainty the influence of puzzles and game-based learning with other variables such as achievement, emotional engagement, success level, enthusiasm, and social interactions. Moreover, integrating Parson's puzzles into a strategy board game exposes two fundamental aspects of software development that required due attention: tool's usability and efficiency.

Therefore, there is need for further investigation to evaluate the learning tool with respect to usability and efficiency. At least two experiments will be conducted to validate the implementation. The first experiment would investigate the interface usability and test if the tool supports students to improve programming skills through increased motivation, engagement, and interaction. The second experiment is meant to ascertain the impact of our new tool on split-attention and cognitive load during the process of solving programming task. The learning tool will be trialed among computer science students in different contexts.

REFERENCES

[1] A. Balanskat and K. Engelhardt, "Computing our future: Computer programming and coding priorities, school curricula and initiatives across Europe," European Schoolnet, October 2015.

[2] A. Robins, J. Rountree and N. Rountree, "Learning and teaching programming: A review and discussion," Computer Science Education, vol. 13, pp. 137-172, 2003.

[3] A. Mathrani, S. Christian and A. Ponder-Sutton, "PlayIT: Game based learning approach for teaching programming concepts," Educational Technology & Society, vol. 19, pp. 5-17, 2016.

[4] D. Parsons, and P. Haden, "Parson's programming puzzles: A fun and effective learning tool for first programming courses," Conferences in Research and Practice in Information Technology Series, vol. 52, pp. 157-163, 2006.

[5] P. Ihantola and V. Karavirta, "Two-Dimensional Parson's puzzles: The concept, tools, and first observations," Journal of Information Technology Education, vol. 10, pp. 119-132, 2011.

[6] T. Sirkiä, "Combining Parson's problems with program visualization in CS1 context," In Proceedings of the 16th Koli Calling International Conference on Computing Education Research, ACM, New York, NY, USA, pp. 155–159, 2016.

[7] M. Lopez, J. Whalley, P. Robbins and R. Lister, "Relationships between reading, tracing and writing skills in introductory programming," In Proceedings of the Fourth international Workshop on Computing Education Research, ACM, New York, NY, USA, pp. 101–112, 2008.

[8] T. Rajala, E. Kaila, R. Lindén, E. Kurvinen, E. Lokkila, M. Laakso and T. Salakoski, "Automatically Assessed Electronic Exams in Programming Courses," In Proceedings of the Australasian Computer Science Week Multiconference, ACM, New York, NY, USA, pp. 11:1–11:8, 2016.

[9] P. Denny, A. Luxton-Reilly and B. Simon, "Evaluating a new exam question: Parsons problems," In Proceedings of the Fourth international Workshop on Computing Education Research, ACM, New York, NY, USA, pp. 113–124, 2008.

[10] V. Karavirta, J. Helminen and P. Ihantola, "A mobile learning application for Parsons problems with automatic feedback," In Proceedings of the 12th Koli Calling International Conference on Computing Education Research, ACM, New York, NY, USA, pp. 11–18, 2012.

[11] S.H. Ab Hamid and Y.F. Leong, "Learn programming by using mobile edutainment game approach," in Proceedings - DIGITEL 2007: First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning, pp. 170-172, 2007.

[12] S. Vinay, M. Vaseekharan and D. Mohamedally, "RoboRun: A gamification approach to control flow learning for young students with TouchDevelop," arXiv preprint arXiv:1310.0810, 2013.

[13] T. Huang, "Strategy game programming projects," In Proceedings of the sixth annual CCSC northeastern conference on the journal of computing in small colleges, vol. 16, pp. 205–213, 2001.

[14] 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, Springer, online first version, 2017.

[15] P. Johannesson and E. Perjons, A Design Science Primer, CreateSpace Independent Publishing Platform, 2012.

[16] T. Amiel and T.C. Reeves, "Design-based research and educational technology: Rethinking technology and the research agenda," Journal of Educational Technology & Society, vol. 11, pp. 29-40, 2008.

[17] De Villiers, M R (Ruth) and P.A. Harpur, "Design-based research - the educational technology variant of design research: Illustrated by the design of an m-learning environment," pp. 252–261, 2013.

[18] C.N. Galilea, "Decision making system for the game oware," Home University: Facultad de Informática de Madrid (Spain) & Institut für Algorithmen und Kognitive Systeme (IAKS), Fakultät für Informatik der Universität Karlsruhe (Germany), pp. 1-68, 2004.

[19] S.S. Oyelere, J. Suhonen and E. Sutinen, "M-learning: A new paradigm of learning ICT in Nigeria," International Journal of Interactive Mobile Technologies (iJIM), vol. 10, pp. 35-44, 2016.

[20] M. Tedre, E. Sutinen, E. Kähkönen, and P. Kommers, "Ethnocomputing: ICT in cultural and social context," Communication of ACM 49, 1, pp. 126-130, 2006.