Mathematical Sciences and Informatics Journal
Vol. 1, No. 1, August 2020, pp 77-89
http://www.mijuitmjournal.com

Computer Game Application for JAVA Programming Language Learning

Nurul Hafizah Zainal Abidin
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, Perak, Malaysia
nurul354@uitm.edu.my

Roslah Arsad
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, Perak, Malaysia
rosla280@uitm.edu.my

Norliana Muslim
Faculty of Engineering and Life Sciences, Universiti Selangor, Malaysia
norliana_muslim@unisel.edu.my

Suraya Masrom
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, Perak, Malaysia
suray078@uitm.edu.my

Article Info

Article history:

Received Feb 05, 2020
Revised May 05, 2020
Accepted Jun 15, 2020

Keywords:

Programming
JAVA game
Learning
Teaching

ABSTRACT

Programming is one of the most advanced and valuable skills increasingly adopted by students these days. Learning a programming language is important as it helps to improve the ability to automate, collect, handle, correctly interpret data and information. However, some students have difficulties to internalize the concepts and master the programming skills. Consequently, students tend to perceive programming course as a difficult subject. The objective of this paper is to present the design and implementation of a computer game application that focused on JAVA programming and Bloom's taxonomy to enhance the students' understanding. To evaluate the JAVA game application, functional and usability testing have been conducted that involved students from the Diploma of Computer Sciences Program in Universiti Teknologi MARA, Perak, Malaysia. The results were very encouraging as majority of students were satisfied with the JAVA game application by means of programming skill improvement and usability acceptance.

Corresponding Author:
Nurul Hafizah Zainal Abidin
Faculty of Computer and Mathematical Sciences,Universiti Teknologi MARA, Perak Branch, Tapah Campus, Perak, Malaysia.
email: nurul354@uitm.edu.my

1. Introduction

Programming is an essential in the field of computer science, where students need analytical and logical skills to solve the problem of programming. Computer programming enables one to become a problem solver and technology creator. However, major students faced difficulties in learning and understanding programming language, especially JAVA programming [1]. Statistically, the percentage of pass rates for some universities reported in [2] is 25% for fourth-year students, 44% for third-year students, and 31% for second-year students, considered as very low pass rate. Furthermore, most of the students assumed that the programming course is very complicated, which leads to their disinterest in computer science, and thus produces a high dropout rate [2]. Researchers in [3] identified the increased motivation of the students while learning can be achieved

through the utilization of a game application. Therefore, the study reported in this paper introduced a computer game application for JAVA programming gameplay or JAVA game suitable for Diploma of Computer Science students. The JAVA game application includes basic syntax and decision-making logic programming such as selection and repetition with assessment that employed Bloom's taxonomy cognitive level [4]. The Blooms' taxonomy is very significant and effective to the performance of students in any fields of learning [5] but rarely adopted in the existing computer game applications mainly in JAVA gameplay learning application. The hypothesis of this study was students can improve their JAVA programming skill through the JAVA game learning approach.

This paper is organized as follow. Section 2 provides the literature review of topics related to learning programming followed with the research methodology in Section 3. The finding is presented in the Section 4 followed with the conclusion remarks in the last Section 5.

2. Literature Review

Programming is an essential part of software development as it is the basic way of human instructions with computer machine. In the Computer Sciences and Engineering courses, students are introduced to the concept of programming where they are taught with basic skills on identifying problems, use general logic and programming techniques to resolve the identified problem and implement the programming techniques with a programming language [6]. Today, there is a need to include the syllabus of the concept and programming skills in other courses even in the level of secondary schools' students. Moreover, past research showed that learning programming can improve academic performance by means of problem-solving and critical thinking [7]. Problem-solving skill is a fundamental skill for the young generation as the skill requires them to analyse and solve problems with proper methods and accurate decisions. Programming can also stimulate creative thinking as programming involves much logical thinking. Besides, improving their critical thinking, students can improve their confidence level through the process of designing their program. Another importance of programming is developing the soft skills such as focus, communication, and resilience [8],[9]. However, according to researchers in [10],[11] students with less motivation will cause inefficient programming. Therefore, students need to be passionate and have good attitudes to ensure them to be good at programming.

Teaching and learning programming subjects are not easy and always challenging. Hence, past researchers have proposed different practices to overcome this situation. For example, [12] proposes a games-based approach, as a way to interact and improve the student's skills. Students need to take an active part in the learning process. Researchers in [13] suggested that a visual programming environment with graphical objects and computational problem states may be useful in promoting computational problem-solving programming learning. In other way, researchers in [14] have conducted an experimental study to proof that Scratch method provides functional programming with the elements of 3D visualization on objects to improves the general programming skills of the programmers. Among these works, no single application that attempts to adopt Bloom's Taxonomy model.

Bloom's Taxonomy model was developed during the 1950s by Benjamin Bloom and allows us to define the learning accomplishments of the students. Bloom's taxonomy is a multi-tiered model of classifying thinking according to six cognitive levels, including remember, understand, apply, analyze, evaluate, and create [5]. In this study, the three-level taxonomy from remember to apply were used as the fundamental assessment approach. In remember level, students need to recall the facts and basic concepts learned, includes identifying a piece of information that corresponds to the knowledge and recalling the relevant knowledge from long-term memory. The next level is understanding where students need to explain their ideas and concepts. Understand refers to constructing meaning from instructional messages, includes interpreting and being able to present information in another form, exemplifying by giving an example of a concept or principle, classifying category, summarizing major point, comparing similarities and differences between two or more things, and explaining the cause-and-effect model of a system. Next, the apply level where students will apply the information in any situation to solve the program. Apply refers to carry out a procedure in a given situation [15]. It includes executing a procedure in a familiar task and implementing a procedure in an unfamiliar task.

Computer gaming is one of the most popular pastime activities for young adults especially students. Playing popular computer games has been connected to positive psychological ramifications [16]. In addition, playing computer games regularly is linked to a variety of cognitive skills including processing speed and problem solving [17].

Three types of popular game application in learning programming has been compared namely Code Combat (https://codecombat.com/), Code Hunt (https://www.microsoft.com/en-us/research/project/code-hunt/) and CodinGame (https://www.codingame.com/). Based on the game's exploration, none of them has been added with the concept of Bloom's taxonomy as listed in Table 1 below.

Table 1. JAVA Games comparison

Game	Code Combat	Code Hunt	CodinGame
Game elements	-Points and rankings. -Levels of achievements.	- Points and rankings. -Levels with challenges to unlock.	-Points, trophies, badges, rankings. -Levels of achievements.
Interaction style	Keyboard/Mouse	Keyboard/Mouse	Keyboard/Mouse
Environment	2D	2D	2D
Bloom's taxonomy	Not available	Not available	Not available

Table 1 also lists the basic and common features of JAVA games that used points of ranking for assessing the level of success for each player. Keyboard and mouse are the common interaction styles. Although 3D graphics can be more attractive but 2D environment can be possible enough for the gameplay.

3. Methodology

The game application was design based on the Game Development Life Cycle [18], which consists of five phases, namely initiation, pre-production, production, testing, and beta as shown in Figure 1.

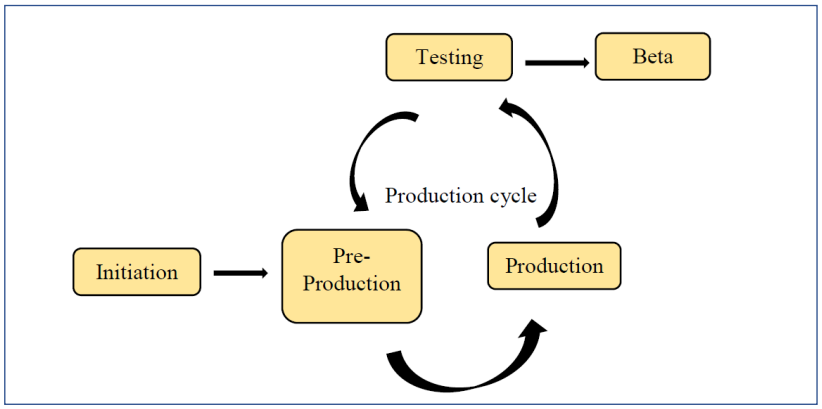


Figure 1. Game Development Life Cycle (GDLC)

The initiation phase is the most important step in GDLC. It is essential at this stage to conduct feasibility study and to identify the focus area and limitations. The next phase after initiation is pre-production that includes the game design and develop the game prototype as seen in Figure 2.

Phases	Activities	Techniques/software	Deliverable
Pre-production	1. Developing the story or script 2. Design flowchart 3. Draw the storyboard of the project.	1. Microsoft Word 2. Microsoft PowerPoint	1. Prototype of the game 2. Documentation

Figure 2. Pre-production phase

In the pre-production phase, the task begins by developing the story scripts with the selected theme that suits the students' level. The story helps to ensure a smooth flow between levels in the game application. Real imagination and creativity are needed in building the storyline to make a proper logical game. The design of the game was presented in a form of flowchart and the storyboard as depicted in Figure 3 and Table 2. Furthermore, Table 3 presents the game storyboard for medium and hard levels.

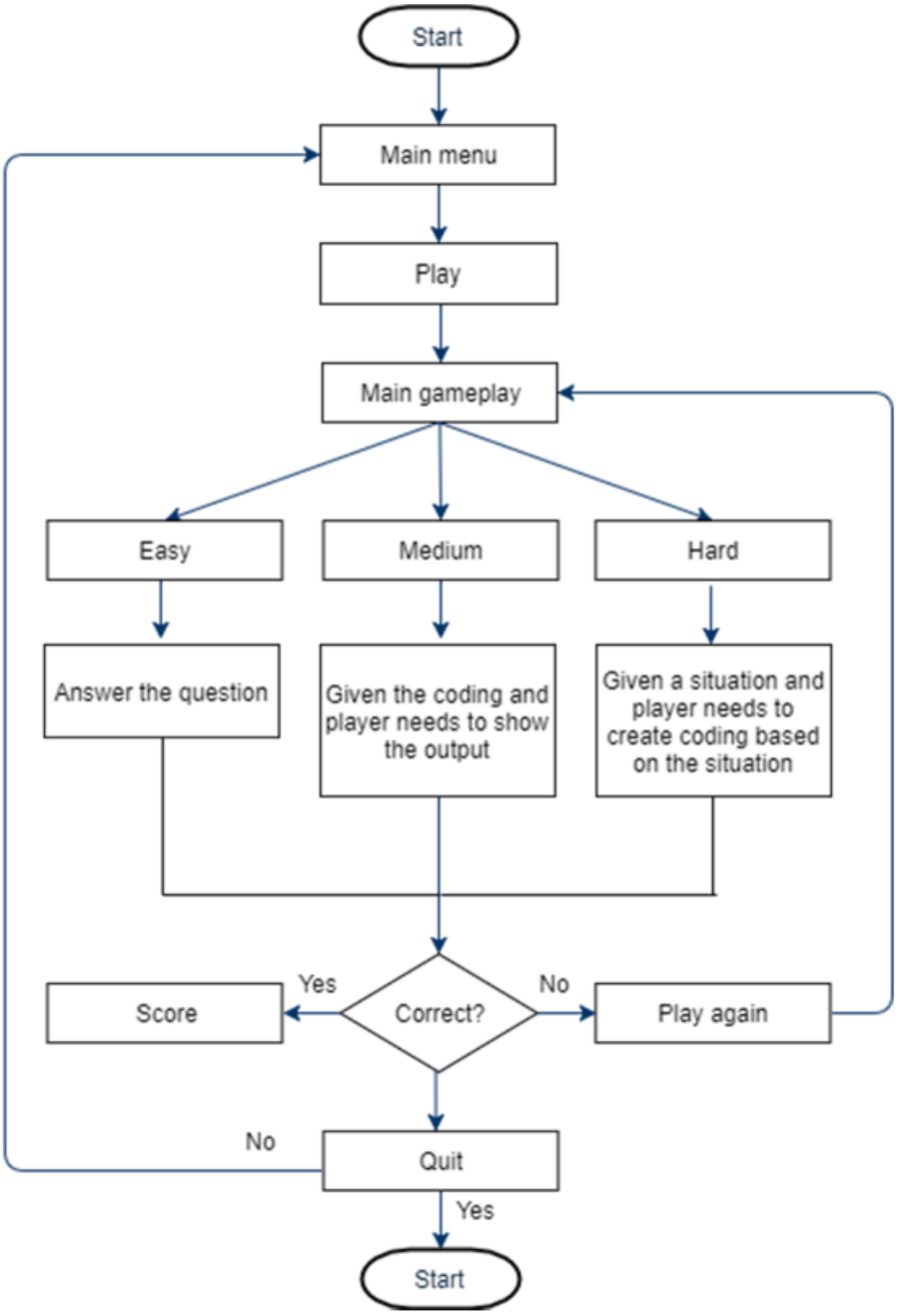


Figure 3. Flowchart of the JAVA game

Table 2. Storyboard of game

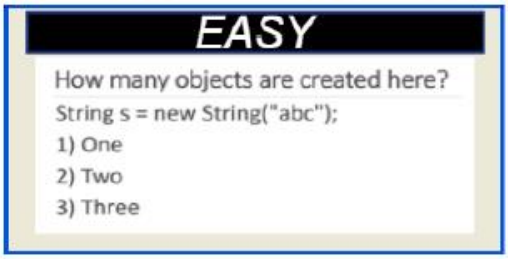
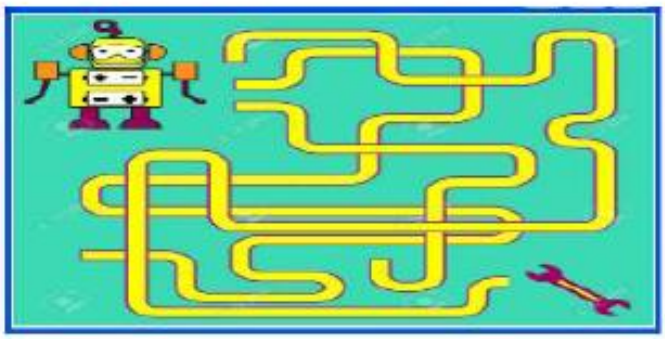
Initial interface design	Description
	Front page of the game application.
	PLAY and QUIT button.
	Levels divided into three stages which are easy, medium, and hard
  	For each level, the player needs to play the game until the end of the point. Once completed to the end, easy question will be asked. Before the player answers the questions, brief notes related to JAVA programming will be given. Example question at the EASY level. This question measures the player with the level one of Bloom's taxonomy namely remember.

Table 3. Game storyboard for Medium and Hard levels

Initial interface design	Description
 	In the MEDIUM level, the player needs to find out the program output through the maze. It is to ensure players understand the basic concepts in JAVA programming to follow the level 2 of Bloom's taxonomy.
	This is an example of the activity at the HARD level. Players are given an incomplete program and they need to complete the program by filling up the empty space in the program. This is to employ level 3 (apply) of Bloom's taxonomy.

The flowchart can be used as a guidance to design the storyboard. Based on the flowchart, the game begins with the start button and then goes to the main menu, play and main gameplay. The main gameplay gives the players an option to choose the levels in the game, which are easy, medium, and hard. The questions prepared for each level are based on Bloom's taxonomy. The easiest level is related to remember element in the Bloom's taxonomy. Their memory stages about the basics of JAVA programming are tested at this level. The medium level is to test the level of understanding, the second level of Bloom's taxonomy. In this level, players are given a program and they need to show the output of the program based on the maze. The hard level considers to apply the knowledge level, where players will be given a situation where they need to identify the problem and solve the problem by applying the programming knowledge to suit with the level three of Bloom's taxonomy. Players need to develop and run their own programs to solve the problem. If each level is correct, players get the score and win. If not, they will be given the opportunity to repeat and choose the level again. If players do not want to continue the game, they can quit and the gameplay will end.

3.1 Production

In Figure 4, the activities of production were implemented based on the pre-production outputs.

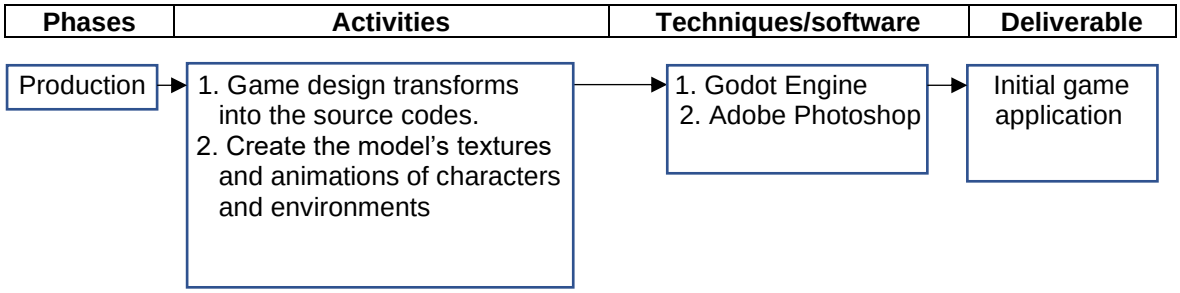


Figure 4. Production phase

The production phase is the fundamental process which the game design will be converted into the software implementation. The model's textures and animations characters were created. Godot Engine software has been used to build the game and Adobe Photoshop software was used to create the characters. SQLite software is the selected database platform that used to store the scores information. Table 4 and Table 5 list the minimum hardware and software required for the development.

Table 4. Hardware requirement for development

Hardware	Specifications
Device	Laptop
Model	Acer
RAM	4GB
Storage	500GB (Internal Hard Disk)
Processor	Intel ® Core™ i3-5005U CPU, 2.0GHz

Table 5. Software requirement for development

Software	Specifications
Computer Operating System	Window
Computer Application Development	Godot engine
Application Interface	Pencil Project
Flowchart	Online application Draw.io
Database	SQLite

To deploy the JAVA games, users need minimum hardware requirements as listed in the Table 4 and the application is only applicable in the Window Operating System.

After the production phase is completed, the game was tested in two phases namely functionality and usability (beta testing). Once errors were identified in the functionality test, the activities in the pre-production phase were reimplemented. After completing the functionality test, usability test in the second phase was conducted that involves users. In this research, the users were 31 students who registered JAVA programming course from the Diploma of Computer Sciences program in the Universiti Teknologi MARA, Perak Branch Tapah Campus. Two set of questionnaires were given separately before and after playing the game application as to see their level of understanding in JAVA programming. Furthermore, the usability testing was conducted to get the users' perception on the JAVA game ease of use.

4. Results and Discussions

This part presents the interfaces of the JAVA game and the testing results.

4.1 The JAVA Game Interfaces

The main interface of the JAVA game is presented by Figure 5 with initial score 0.

Figure 5. Main interface

When START button is selected, screen of easy game level will be displayed first (refer to Figure 6) before the users can be asked with the JAVA question.

Figure 6. Gameplay interfaces before easy level question

In the JAVA games, player needs to jump up or down to get each box until the end of the point. To jump up, user can use keyboard buttons to interact with the game such as up arrow to jump up and down arrow to jump down. Once a box has been reached, the user can move on to the next

box by using the keyboard buttons. When the user arrived at the end of the point, an easy question on JAVA program will be displayed as depicted in Figure 7.

Figure 7. Gameplay interfaces for easy level question

The answer can be selected from the three buttons given at the bottom of the question. Then, the scores will be calculated based on the correct answer given by the user. If wrong answer was given, user will be brought back to the JAVA game interface in Figure 7 and another easy question will be given. Once correct answer is given by the user, the game interface with medium question as in Figure 8 will be displayed.

Figure 8. Gameplay interfaces for medium level question

In the medium level, the player was asked to find the output of the program as to test the student understanding on the logic program. Another question will be asked until correct answer was given, and more difficult question will be displayed as presented in Figure 9 that reached user to the hard level.

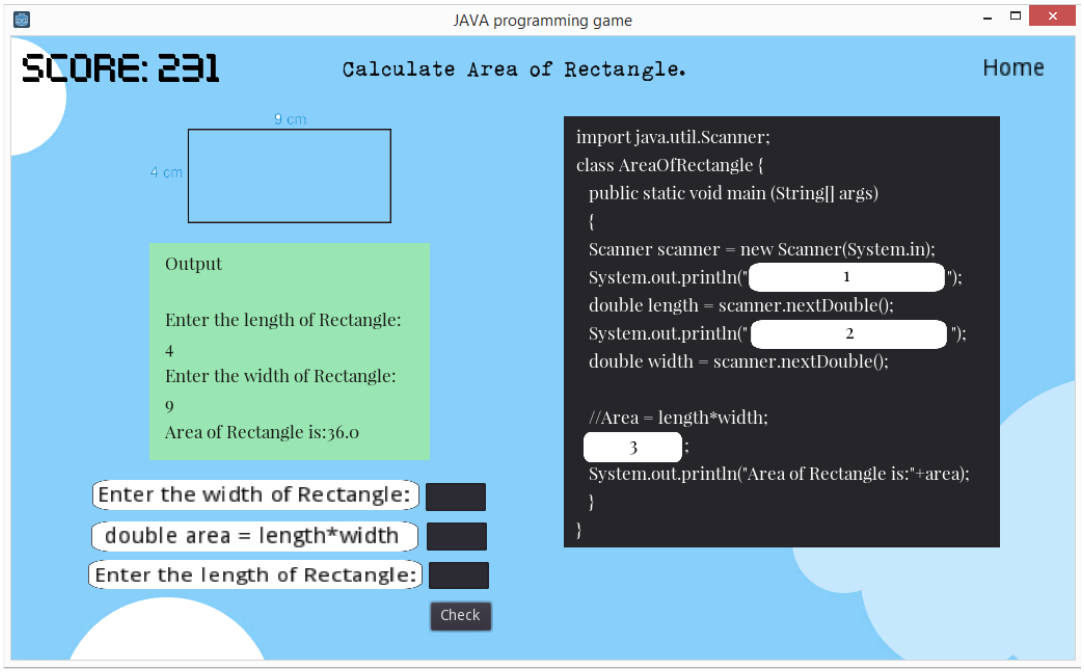


Figure 9. Gameplay interfaces for hard level

The example of question in Figure 10 demonstrates the ability of the JAVA game player to resolve rectangle area calculation. Output of the program is given, and the player can write the correct statements in the black box. The Check button is to display the total scores calculated for the player as presented in Figure 11.

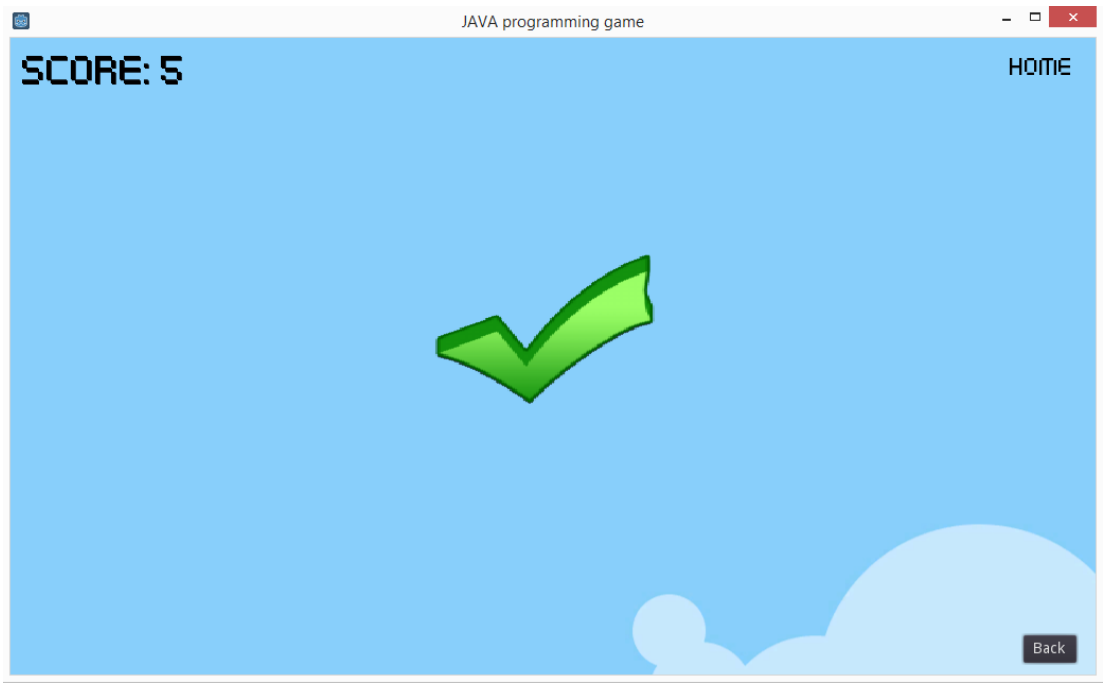


Figure 10. Gameplay interfaces to show the score of the player

The JAVA game was used by 31 students from the Diploma of Computer Sciences in Universiti Teknologi MARA, Perak Branch. To evaluate the effectiveness of the game in assisting students' understanding to learn JAVA programming, data collected from a set of questionnaires survey was analyzed based on effectiveness of student performance and usability.

4.2 Effectiveness of student performance

Tables 6 and Table 7 show the students' perception regarding to the JAVA programming skill that they perceived before and after using the JAVA games respectively.

Table 6. Students' rate their skill in JAVA programming before using the JAVA game

Rate	Number of respondents	Percentage
Very low	0	10
Low	8	25.81
Medium	20	64.52
High	3	9.67
Very high	0	0

Before using the game application, 20 out of 31 respondents have chosen medium to rate their skill in JAVA programming. There were 3 respondents perceived their JAVA skill as high and another 8 respondents with low skill. However, there were some improvement on the results after they played the JAVA game as in Table 7 below.

Table 7. Students' rate skill in JAVA programming after using the JAVA game

Rate	Number of respondents	Percentage
Very low	0	0
Low	0	0
Medium	9	29.03
High	14	45.16
Very high	8	25.81

Most of them perceived some improvement from medium to high regarding their JAVA skill after using the JAVA game. No students that rated their JAVA skill to low and very low. More interesting, 8 of the 31 students perceived more confident with very high rating.

4.3 Usability of the JAVA game

Usability testing is to measure the users' acceptance towards the ease of use in the JAVA game. Table 8 is the number and percentage of respondents that evaluate the usability of the JAVA game.

Table 8. Usability of the JAVA game

Rate	Number of respondents	Percentage
Strongly Agree	13	41.94
Agree	16	51.61
Neutral	2	6.45
Disagree	0	0
Strongly Disagree	0	0

There were 13 out of the 31 respondents were strongly agree on the JAVA game features and functionality are easy to use. Another 16 respondents were agreed and none of them were disagreed nor strongly disagreed.

5. Conclusions

The JAVA game application has several weaknesses for further improvement in the future research. Firstly, the application currently is only compatible in a computer that use Window Operating Systems. Secondly, this game application covers on the basic JAVA concept. Therefore, the nearest improvement should be directed to focus on these two limitations.

However, the JAVA game application proposed in this study has been proved to run smoothly and can bring good effect on the users who learn JAVA program. The game can be used as additional learning material for students outside the formal classes given by the lecturers. It is anticipated that the JAVA games will be useful and interesting to the school students. Therefore, further study of students' acceptance should be conducted on different kind of students who need to learn JAVA program.

Acknowledgment

The acknowledgment is firstly dedicated to the research assistant of this project, Wan Syahirah Wan Pakhzi who developed the JAVA games application. The authors would also like to thank Universiti Teknologi MARA, Perak Branch Tapah Campus for the full support to implement this project.

References

[1] T. Jenkins and J. Davy, "Diversity and motivation in introductory programming," *Innov. Teach. Learn. Inf. Comput. Sci.*, vol. 1, no. 1, pp. 1–9, 2002.

[2] K. A.-M. Sarpong, J. K. Arthur, and P. Y. O. Amoako, "Causes of failure of students in computer programming courses: The teacher-learner Perspective," *Int. J. Comput. Appl.*, vol. 77, no. 12, 2013.

[3] R. Ibrahim, R. C. M. Yusoff, H. Mohamed-Omar, and A. Jaafar, "Students perceptions of using educational games to learn introductory programming," *Comput. Inf. Sci.*, vol. 4, no. 1, p. 205, 2011.

[4] M. Forehand, "Bloom's taxonomy," *Emerg. Perspect. Learn. teaching, Technol.*, vol. 41, no. 4, pp. 47–56, 2010.

[5] S. Masapanta-Carrión and J. Á. Velázquez-Iturbide, "A systematic review of the use of Bloom's taxonomy in Computer Science education," in *Proceedings of the 49th acm technical symposium on computer science education*, pp. 441–446, 2018.

[6] M. N. Ismail, N. A. Ngah and I. N. Umar. Instructional strategy in the teaching of computer programming: a need assessment analyses". *TOJET: The Turkish Online Journal of Educational Technology.*, vol. 9, no. 2, 2010.

[7] C. C. Liu, Y. B. Cheng, and C. W. Huang. "The effect of simulation games on the learning of computational problem solving". *Computers & Education.*, vol. 57, no. 3, pp. 1907-1918, 2011.

[8] M. Barak, and S. Rafaeli. "On-line question-posing and peer-assessment as means for web-based knowledge sharing in learning". *International Journal of Human-Computer Studies.*, vol. 61, no. 1, pp. 84-103, 2004.

[9] C. Kazimoglu, M. Kiernan, L. Bacon, and L. Mackinnon. "A serious game for developing computational thinking and learning introductory computer programming". *Procedia-Social and Behavioral Sciences.*, vol. 47, pp. 1991-1999, 2012.

[10] A. Robins, J. Rountree, and N. Rountree. "Learning and teaching programming: A review and discussion". *Computer Science Education.*, vol. 13, no. 2, pp. 137-172, 2003.

[11] M. Hauswirth and A. Adamoli. "Teaching Java programming with the Informa clicker system". *Science of Computer Programming*, vol. 78, no. 5, pp. 499-520, 2013.

[12] R. Rajaravivarma. A games-based approach for teaching the introductory programming course". *Inroads* SIGCSE Bulletin., vol. 37, no. 4, pp. 98-102, 2005.

[13] P. Y. Chao. "Exploring students' computational practice, design and performance of problem-solving through a visual programming environment." *Computers & Education.,* vol. 95, pp. 202-215, 2016.

[14] D. Topalli, and N. E. Cagiltay. "Improving programming skills in engineering education through problem-based game projects with Scratch". *Computers & Education.*, vol. *120*, pp. 64-74, 2018.

[15] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock. A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives. New York., Longman, pp. 67, 2001.

[16] I. Granic, A. Lobel, and R. C. Engels. "The benefits of playing video games". *American psychologist.*, vol. 69, no. 1, pp. 66, 2014.

[17] M. J. Stroud, and S. K. Whitbourne. "Casual video games as training tools for attentional processes in everyday life". *Cyberpsychology, Behavior, and Social Networking.*, vol. 18, no. 11, pp. 654-660, 2015.

[18] R. Ramadan, and Y. Widyani, "Game development life cycle guidelines". In *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS).,* pp. 95-100, 2013.