SEMARAK ILMU
PUBLISHING
202103268166(003316878-P)

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Pantas.io: Game-Based Learning to Cultivate Programming Skills for Primary School Students

Sumayyah Dzulkifly[1,*], Wong Yoke Seng[2], Lee Hoi Yeh[3], Abu Bakar Ibrahim[1], Song Zhiqiang[2]

[1] Department of Software Engineering and Smart Technology, Faculty of Computing and Meta-Technology, Universiti Pendidikan Sultan Idris, 35900 Tanjong Malim, Perak, Malaysia
[2] Department of Computer Science and Digital Technology, Faculty of Computing and Meta-Technology, Universiti Pendidikan Sultan Idris, 35900 Tanjong Malim, Perak, Malaysia
[3] Department of Art and Design, Faculty of Art, Sustainability and Creative Industry, Universiti Pendidikan Sultan Idris, 35900 Tanjong Malim, Perak, Malaysia
[4] School of Art and Design, Xi'an Fanyi University, Xi'an, 710105 Shaanxi, China

ARTICLE INFO

Article history:
Received 4 November 2023
Received in revised form 19 May 2024
Accepted 29 June 2024
Available online 10 August 2024

Keywords:

Programming; game-based learning; primary school

ABSTRACT

Programming is considered as one of the serious subjects that requires skills such as planning and time management. With the rapid growth of the technology, programming is one of the most required talents worldwide. Yet, current trend shows that younger generation's lack of interests to pursue programming related subjects. This is due to fallacious idea that programming is a serious and complex subject. Nurturing "perceived serious subject" such as programming to younger students requires assimilations of a non-serious learning approach such as game-based learning. In parallel with the issue, the purpose of this study is to determine how well game-based learning could enhance student engagement and fundamental programming skills. A game-based learning application is created using Unity and C# language to evaluate advancements in a quasi-experimental setting using pre- and post-test questionnaires. This study utilizes pre- and post-test research design involving 30 year 4 students in selected school. The preliminary outcomes show an approximately 21.3 percent overall improvement in programming enthusiasm, coordination, planning, and time management abilities. The results support the main goal of the study, which is to adopt game-based learning to increase primary school pupils' motivation and develop critical programming abilities.

1. Introduction

The advancement in technology has shifted the learning environment at school. Previously, students acquired knowledge at schools through one-way teacher-centric communication, where the teacher teaches, and the students listen. These days, such a conventional method is no longer effective in achieving a good learning outcome. Various studies have shown that interactive learning, where students actively take part during the learning process is more effective in achieving the targeted learning outcome, especially for the subjects that focus on nurturing specific skills [1,2]. One

[*] *Corresponding author.*
E-mail address: sumayyah.dzul@meta.upsi.edu.my

https://doi.org/10.37934/araset.50.1.178190

of the most well-known interactive learning methods is game-based learning. Game-based learning is a method where the learning process is integrated into a game to achieve a specific learning outcome [1]. Game-based learning approach is usually used to boost students' motivation and increase their engagement in learning activities.

Game-based learning or gamification is considered one of the solutions to improve students' motivation in learning subjects related to programming or coding [1,3]. This is in line with the issues where students are no longer interested in learning subjects related to science or technology, which includes programming. This issue arises despite the increase in demand for talent in this field [4]. This is partly due to a lack of nurtured skills and motivation needed to encourage them to pursue education related to programming or coding. The issue becomes worse when such subjects are relayed in a conventional teaching method at school. It is important to instill interest and boost motivation at an early age. Programming requires a set of skills such as problem-solving, inquisitiveness, adaptability, and good planning. This should be cultivated early, with informal interactive lessons at school.

Acknowledging the issue, this study proposes an integration of a game-based learning approach to encourage the development of soft skills in programming for primary school children in Malaysia, named *Pantas.io*. The game is developed with one main goal, which is to boost programming skills among students during their primary education. The game is developed using Unity as the game engine, which is then converted into an Android Package Kit (APK) to enable installation on mobile devices. The game rules are written in C# language using Microsoft Visual Basic. This study employs a quantitative research approach, specifically a quasi-experiment where there are pre-and post-tests to observe the effect of game-based learning in terms of increased interest, motivation, and skills among primary school students.

This paper comprises five main sections, namely: 1. Introduction**,** where the overall background of the research is elaborated; 2. Literature Review provides an extensive review in terms of the issue of lack of interest to pursue programming among primary school children, prior researches on game-based learning or gamification, the differences between the two and the common motivation model used in the research; 3. Methodology elaborates on the techniques used to develop game-based lessons as well as the method used for data collection and analysis; 4. Results and Analysis provide intricate evaluations of the overall outcomes of this study; 5. Conclusion; summarizes the main points and concludes the study.

2. Literature Review

This section will cover relevant earlier studies that have addressed similar issues and proposed approaches that serve as the benchmark for developing a game-based lesson for this study.

2.1 Lack of Interest in Pursuing Programming Subject among Primary School Children

Despite the growing emphasis on computer science education in early years, a significant proportion of primary school students appear to struggle with or be disinterested in programming. The study in Sun *et al.*, [5] points out the possible reason behind disinterest in programming due to lack of experience which could be interpreted as any early exposure to skills related to programming. Meanwhile, studies in Riza *et al.*, [4] and Sun *et al.*, [5] suggest that this lack of interest may be attributed to the abstract and technical nature of programming concepts, which can appear daunting and irrelevant to young learners' daily experiences.

Recent research conducted by Morgan *et al.*, [7] and Soumia *et al.*, [8] delves deeper into this issue and highlights the role of instructional methods in mitigating primary school students' disinterest in programming. The authors in [7] and [8] argue that traditional teaching approaches often do not cater to the diverse learning styles and preferences of young children, leading to disengagement. Furthermore, the lack of exposure to practical, real-world applications of programming can exacerbate students' reluctance to engage with the subject. To address this challenge effectively, educators and curriculum designers need to explore innovative pedagogical strategies and curricular materials that make programming more accessible, enjoyable, and relevant to young learners.

2.2 Gamified and Game-Based Learning for Primary Education

Several significant previous studies on gamification in programming lessons for children aged 8 to 11 years old are used as the foundation for this study. A study in Altaie *et al.*, [9] proposed a gamification framework in selected lessons based on student's learning styles to stimulate computational thinking (CT). It integrates the Felder-Silverman Learning Style Model (FSLSM) into its proposed gamification framework for its learning management system (LMS). Meanwhile, the study in Cheng *et al.*, [10] applies student-generated questions (SGQ) as the learning strategy in its proposed game-based learning (GBL) platform. Both Altaie *et al.*, [9] and Cheng *et al.*, [10] show significant improvements in terms of students' motivation and overall performance, especially in terms of their computational thinking (CT) ability.

Gamified lessons or learning approaches are not only applied as the learning platform. In some cases, physical devices (e.g., toys and robots) or specific game components (e.g., puzzles and blocks) are employed to increase student engagement and focus during the learning process. This is shown in the study by Cruz-García *et al.*, [11], which investigates the utilization of the visual programming approach to attract young students toward programming. It employs an open-source block game named *Blockly*, which is shown to produce notable gains in terms of the student's overall educational experience. In contrast, the study in Heljakka *et al.*, [12] uses robots to show students how programming fundamentally works. The practicality of the approach enables students to learn from the cause-and-effect dynamics of programming, which leads them to preserve focus during the learning process.

Interestingly, other researchers have employed slightly different approaches in integrating gamification for programming or coding-based lessons. For example, the authors in Vahldick *et al.*, [13] introduce a serious game to enable students to learn the fundamentals of coding. An approach as such allows continuous improvements of the game built specifically for learning purposes. Conversely, the study in Videnovik *et al.*, [14] adapts board games into instructional strategies such as flipped classrooms or project-based learning as the means to teach young students about programming skills. It shows that games could increase children's motivation when it comes to programming.

2.3 Differences between Game-Based Learning and Gamification

While the terms game-based learning and gamification have been widely utilized for research on education in the 21st century, there are prominent differences between these two terms. Game-based learning is the approach where conventional learning is adapted into a game to achieve specified learning outcomes of the subject. For example, role-play game (RPG) based learning is employed by authors in Chen *et al.*, [15] to stimulate critical thinking among high school students. A review by Byusa *et al*., [16] observes that various research employing game-based learning to teach chemistry could significantly improve students' interest and motivation.

Alternately, gamification does not focus on the game development or learning outcome. Instead, game elements such as punishments and/or rewards are employed in the learning process to make the learning interactive and fun for the students. It means that any teaching or learning approach that employs game elements is considered gamification. This concept is applied in Nand *et al*., [17] where the concept from "Who wants to be a millionaire" utilizes the rewards and punishment scheme to increase students' engagement in learning numeric operations. Meanwhile, a study in Schatten *et al*., [18] utilizes gamification to teach computer science to high school students in Croatia to increase their motivation to learn the subject.

2.4 ARCS Learner's Motivation Tools

Students' motivation is commonly evaluated using the Attention, Relevance, Confidence, and Satisfaction (ARCS) model [19-22] where each category is interpreted into subcategories such as follows:

- Attention: sensory stimulation, inquiry stimulation, and variation of approach.
- Relevance: goal-oriented and familiar.
- Confidence: learning requirement, success opportunities, ability to control the learning pace.
- Satisfaction: self-sustaining rewards, external reinforcements, and impartiality.

The model is employed in this study to observe the level of motivation of students acquiring programming skills through the developed game-based learning, namely *Pantas.io*.

3. Methodology

The methodology of the study is divided into two parts, namely the game development stage and the data collection stage. This will be elaborated further in the subsequent subsections. The development stage explains the steps taken in the game development while the data collection subsection details the sample, data collection approach, and the instrument used.

3.1 Rapid Application Development

The outcome of the lesson is specified at the beginning of the game development. It is to attract young students to nurture skills related to programming, namely problem-solving and time management. Based on previous research discussed in subsection 2.1, the game is developed with the following advisement:

- Maintain continuous engagement from students while enabling them to nurture the basis of programming skills.
- The game-based lesson is graphically pleasant for children aged 8 to 10 years old while being practical and easy enough to understand for them to independently play with minimal assistance from instructors or teachers.
- Minimize miscommunication or misinterpretation in the game. As the target for this study is students in Malaysia, the game is developed in the Malay language to reduce the language comprehension burden on the students.

- Tool utilization for ease of gameplay. *Pantas.io* was developed to enable students to play using their mobile devices online or offline. This enables the utilization of mobile devices as the learning tool for this study.

This study employs the most common game development research methodology, namely rapid application development (RAD) [23]. RAD integrates existing applications while iteratively improving the prototype based on user feedback. Figure 1 displays the flow of RAD methodology, which involves defining the requirement analysis, building the initial prototype, testing the prototype, refining the prototype based on user feedback, finalizing the prototype at the construction stage, and measuring user acceptance at the cutover stage.

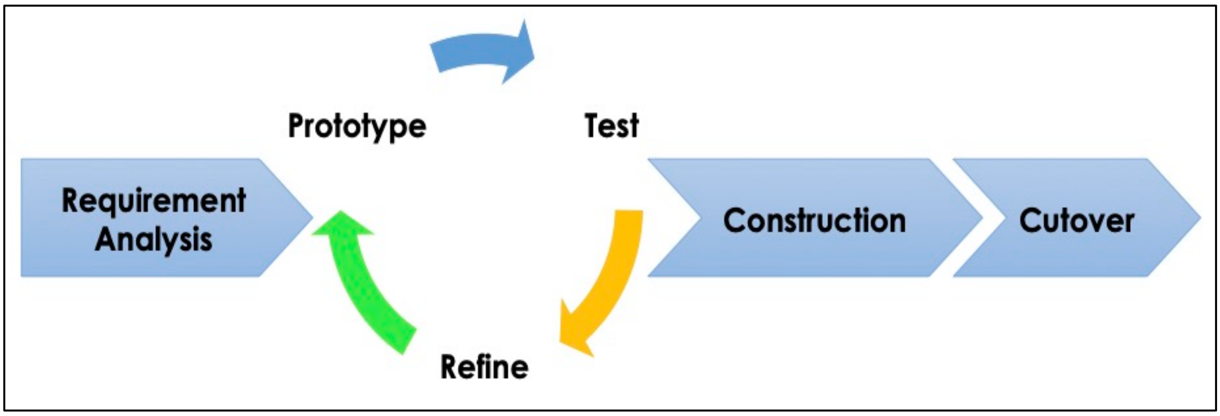


Fig. 1. The rapid application development (RAD) process for game development

3.1.1 Requirement analysis

Before the development of the game, a use case diagram is developed to specify the distinctive requirements according to the aim of this study. The target user for the game is students aged 8 to 10 years old with minimal to no programming skills. Hence, the flow of the game and the interfaces involved should be simple and easy to use. In addition, the common language used by these students is the Malay language.

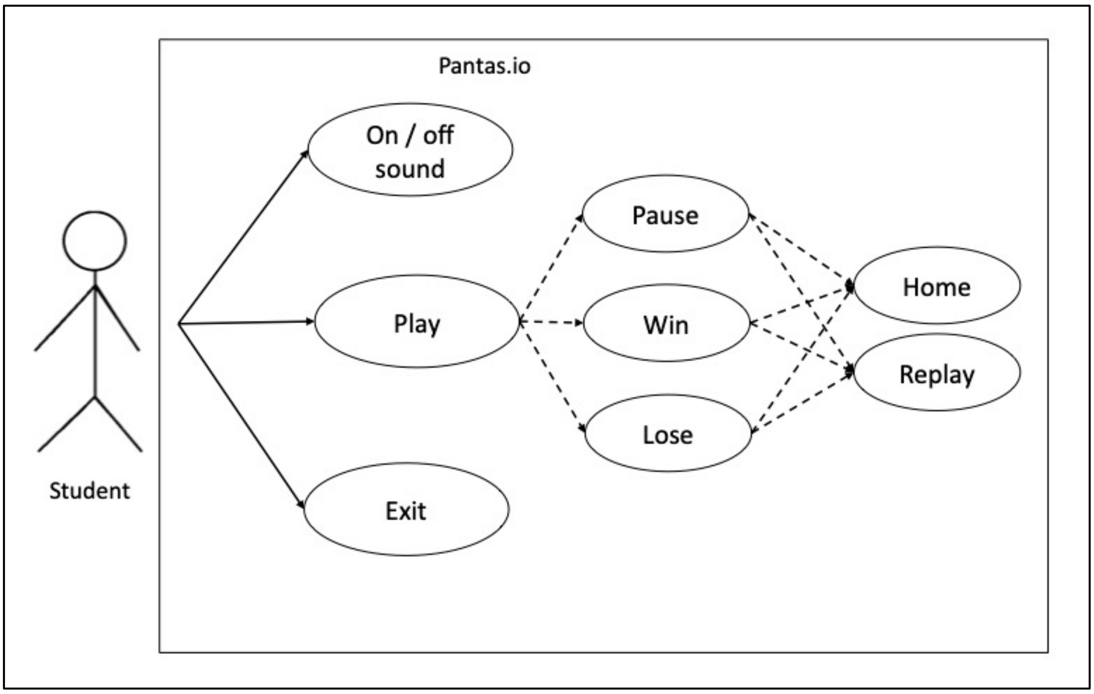


Fig. 2. The use case diagram for *Pantas.io* game

Figure 2 displays the use case diagram of the *Pantas.io* game, which involves straightforward functionalities such as on/off sound, play, exit, pause, win, lose, home, and replay. These functionalities are important in ensuring that the whole game can operate following the activity diagram as part of the sequence of activity of the game, which is translated in the user design stage of the study.

3.1.2 User design (prototype, test, and refine)

This stage involves the iterative process of building and refining the product, based on user feedback. The activity diagram in Figure 3 illustrates the flow of activity within the game. Once the user or the player, in this case, the student clicks the play button, the game arena is displayed and the timer starts. The player can use the given limited card to plan their move in order to reach the pre-set goal. This game continues until the cards run out or the player reaches the goal. During this stage, the game is continuously developed to ensure that it achieves the practicality that suits the targeted user, in this case, students aged 8 to 11 years old.

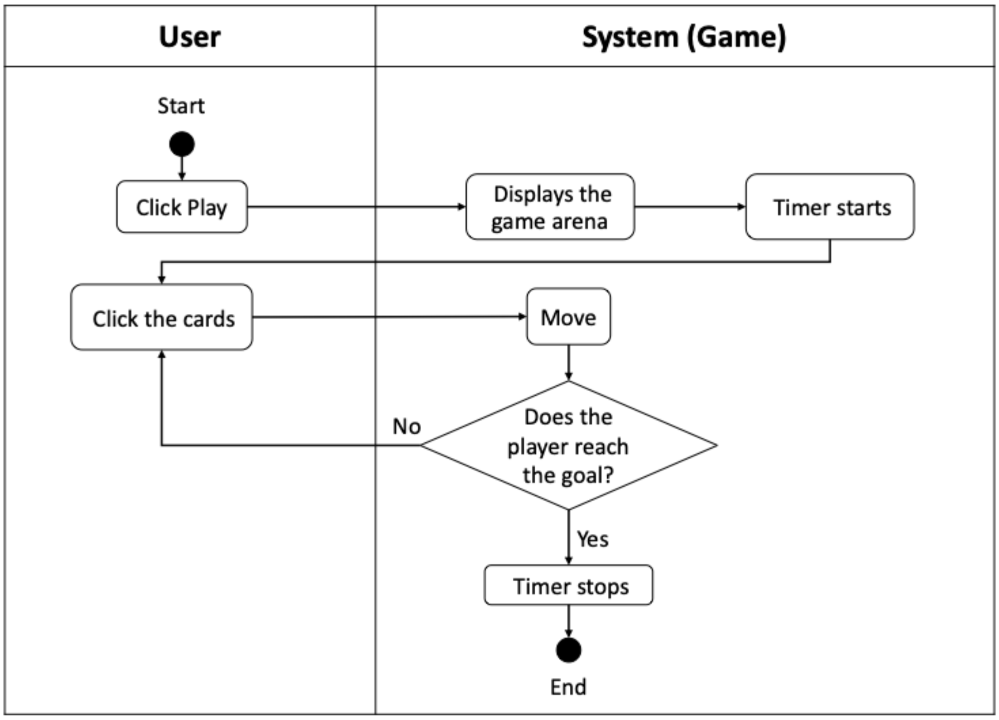


Fig. 3. The flow of activity for *Pantas.io* game

3.1.3 Construction and cutover

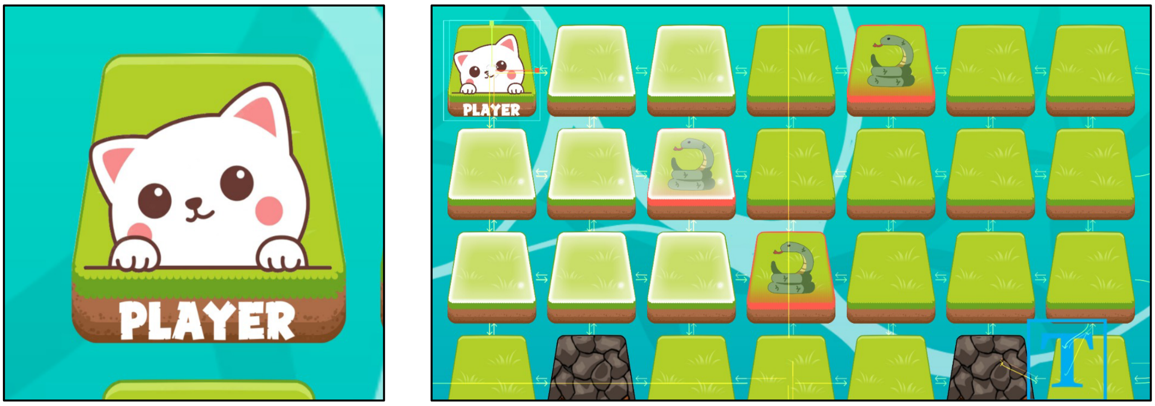
At this stage, the developed game is finalized based on the accumulated feedback from users. Some of the modifications made are on the graphics, the positioning of the goal, and the number of card click limits allowed per user. Final user acceptance is carried out to gauge the level of user satisfaction with the game.

3.2 Pantas.io*: Online Game Development*

The first part of the game is the development of the logo. The game emulates the concept of the logos for *Plants vs. Zombie* and *Snake and Ladder*. The game is given the name *Pantas.io* as the game evaluates how fast the player can reach the goal using the best route planning. Figure 4 illustrates the finalized game logo. As mentioned previously, the game is developed using Unity as the game engine, and the rule for the game is written using C# language.

Fig. 4. *Pantas.io* designed logo

Another part of the game is the number of important elements in the game including the graphics for the player, the movement box, the penalty box, and the wall or the hurdle. The graphics selected are based on the target user of the game, as illustrated in Figure 5 (a) and (b). This is to ensure that the player can understand the game well with minimal instructor intrusion. In order to achieve this, the icons for the player, the penalty, and the wall or the hurdle are selected in such a way that the player can easily understand what the icons mean.



(a) (b)

Fig. 5. (a) The player icon in the *Pantas.io* game and (b) The penalty and wall icons in the *Pantas.io* game

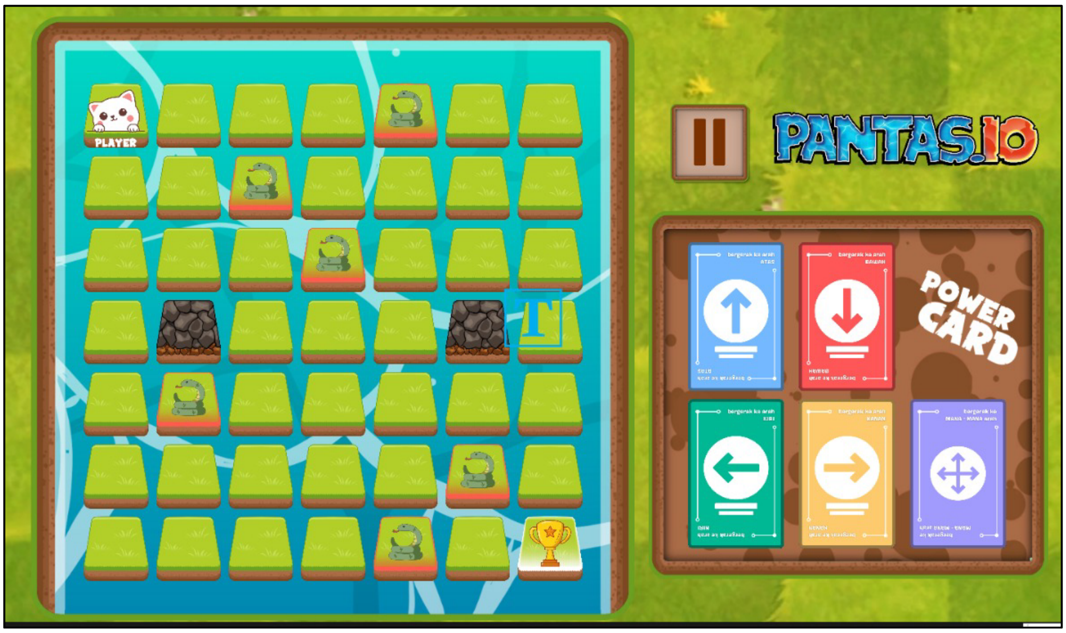
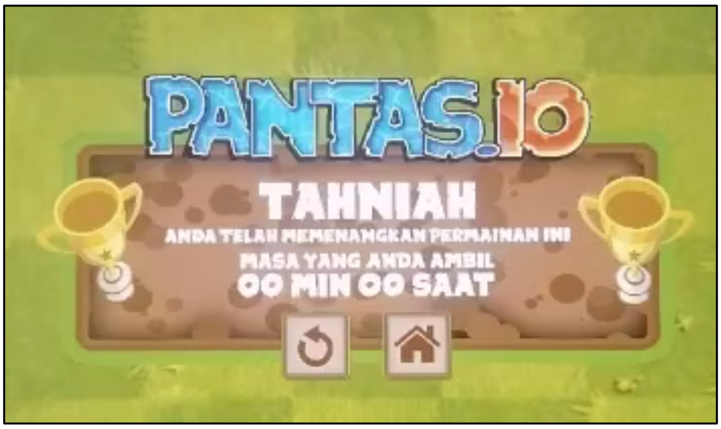


Fig. 6. The *Pantas.io* game arena view

The overall view of the game is as illustrated in Figure 6 where the player is given a set of power cards to plan the best move to reach the goal. The pause button is included to ensure that the player can pause the game at any time. In addition, we have also included a page that informs players of the time taken to complete the game, as illustrated in Figure 7.



TAHNIAH
ANDA TELAH MEMENANGKAN PERMAINAN INI
MASA YANG ANDA AMBIL
00 MIN 00 SAAT

Fig. 7. The display for the time taken by the player to finish the game

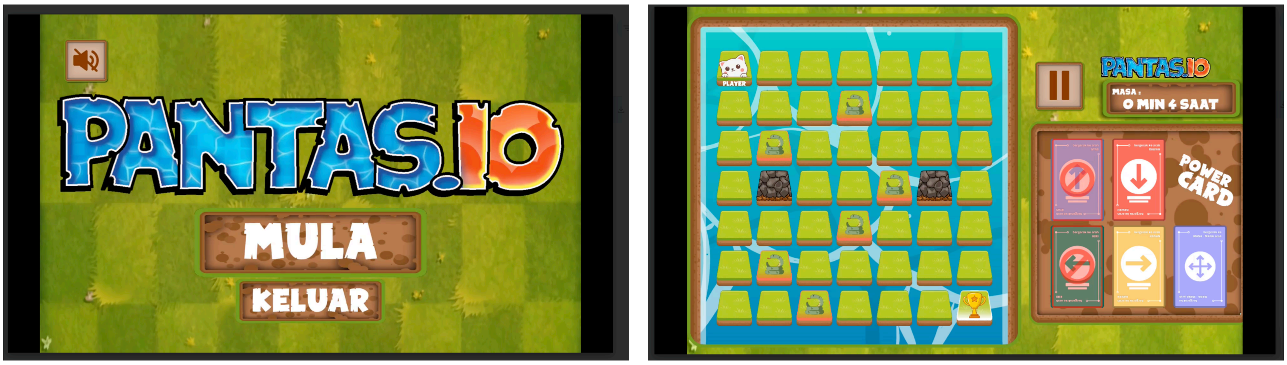
3.3 Data Acquisition and Analysis

The study observes the effects of the proposed game-based lesson by employing quasi-experimental quantitative research, where we utilize pre- and post-tests as one of the research instruments. The questions in the test are inclined toward the student's motivations and skills gained from the game-based lesson. In Malaysia, students in Year 4 of primary school, aged 10 years old are exposed to a programming sub-topic that uses a well-known visual programming tool named *Scratch*, developed by MIT's full name here (MIT). We use the sub-topic module as a comparison with our game-based lesson for the controlled group while the experimental group utilizes the game-based lesson.

Due to the time and cost limitations, we apply purposive sampling, where a number of students from primary schools in Malaysia are selected to be assigned to the controlled and experimental groups. This is carried out under the approval of the school administrators and parents of the students involved. Analysis of the pre-and post-test is carried out using descriptive analysis whereby we identify the impact of the game-based lesson in comparison to the existing sub-topic module.

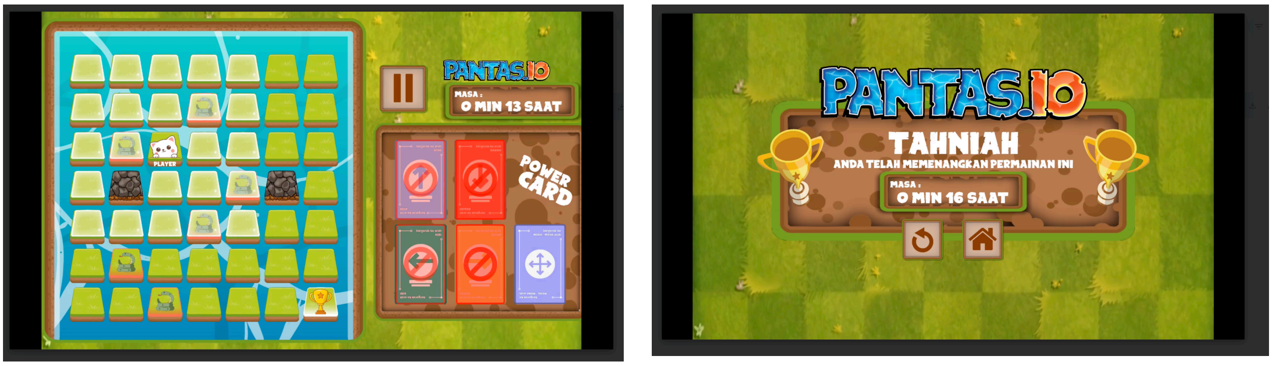
4. Results and Analysis

As elaborated in section 3.2, specific attention has been given to developing the game. The development of the game yields the outputs as illustrated in Figure 8 (a), (b), (c) and (d). The player clicks the *Pantas.io* icon to launch the game and will be greeted with the home page as displayed in Figure 8 (a). The player is directed to the game window when the button "Play" is clicked, as shown in Figure 8 (b), where the timer will start. There will be possible movements suggested to the player if the power card is clicked, as shown in Figure 8 (c). The game is complete once the player reaches the goal or stops playing, as illustrated in Figure 8 (d).



(a) (b)

Fig. 8. (a) Launching display for the *Pantas.io* game and (b) The *Pantas.io* game window or game arena



ANDA TELAH MEMENANGKAN PERMAINAN INI

(c) (d)

Fig. 8. (c) The potential move that the player could choose when the multiple card is clicked and (d) The sample of timer display when the game ends

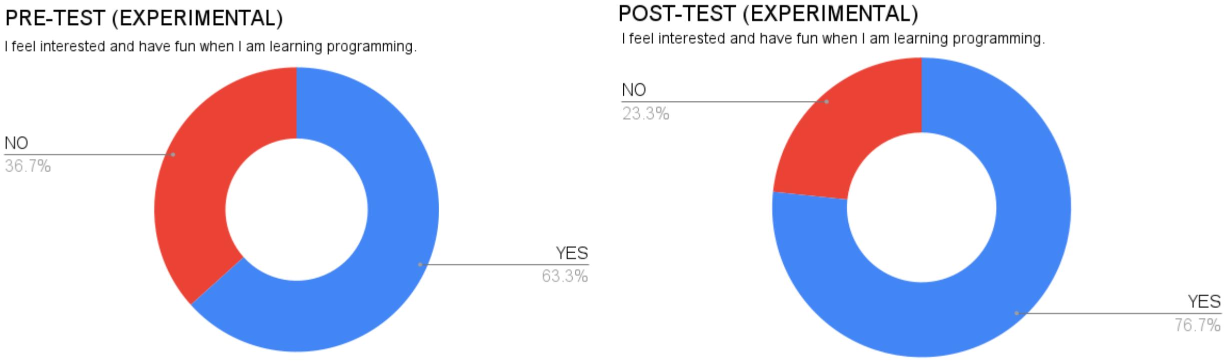
Meanwhile, for the result analysis, we gave several pre- and post-test questions to the students. As this is preliminary data collection, thirty students from standard 4 in one of the schools in Klang, Selangor were selected. Consent from the school principal, the teacher, and the parents involved was obtained before the experiment was carried out. The aim of our proposed game-based lesson is to motivate and nurture early skills in learning programming. Hence, we outline several key questions, listed in Table 1.

Table 1
List of key questions for pre- and post-test

Questions	Details
1	I feel interested and have fun when I am learning programming.
2	Game-based learning helps me learn coordination and estimate steps needed.
3	I do not know how to manage or estimate the time needed to get to the destination.

For these statements, we obtain feedback from the students, which yields the results as displayed in Figures 9 (a) and (b); 10 (a) and (b); 11 (a) and (b). For this study, we would like to highlight the post-test of the experimental and controlled groups' answers to Question 1 as detailed in Table 1.

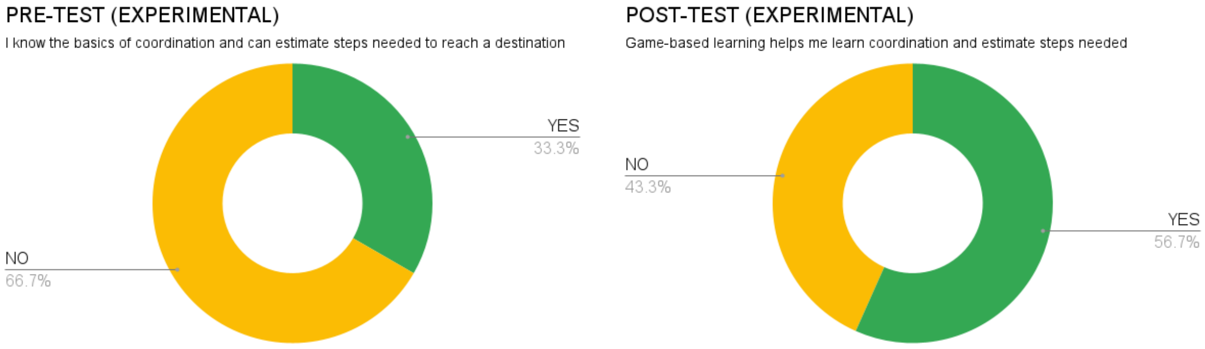
From the chart in Figure 9 (a) and (b), it can be seen that the experimental group that utilizes the game-based lesson has more interest and fun in learning the programming skills. The experimental group has shown a 20.6 percent increase in terms of feeling fun and having an interest in learning programming. While deeper result analysis is needed to get the full picture, the preliminary outcome has shown some potential in our proposed game-based lesson.



(a) (b)

Fig. 9. (a) Pre-test experimental graph for question 1 and (b) Post-test experimental graph for question 1

Next, Question 2 asks about the student's skill improvement in terms of understanding the coordination and planning the steps. From the chart shown in Figure 10 (a) and (b), it can be seen that there is a 23.4 percent increase in their coordination and planning skills after the students are exposed to the game-based lesson.



(a) (b)

Fig. 10. (a) Pre-test experimental graph for question 2 and (b) Post-test experimental graph for question 2

Finally, we look into Question 3, which asks about the time management skills of the students. It can be seen from Figure 11 (a) and (b) that there is a 20 percent increase in time management skills for students who learn programming through game-based lessons. This means that the game lesson is able to instill the skills needed for the student to pursue their interest in learning programming in the future.

From the analysis carried out, we can see that there is an increase of approximately 21.3 percent from pre- to post-test. The results have shown a significant improvement in nurturing the motivation and skills related to programming. The results support the primary aim of this study, namely to utilize game-based lesson to improve motivation and the needed programming skills for students in primary school. While the result is still not sufficient, this will be the benchmark for an extended study that we will carry out in the future.

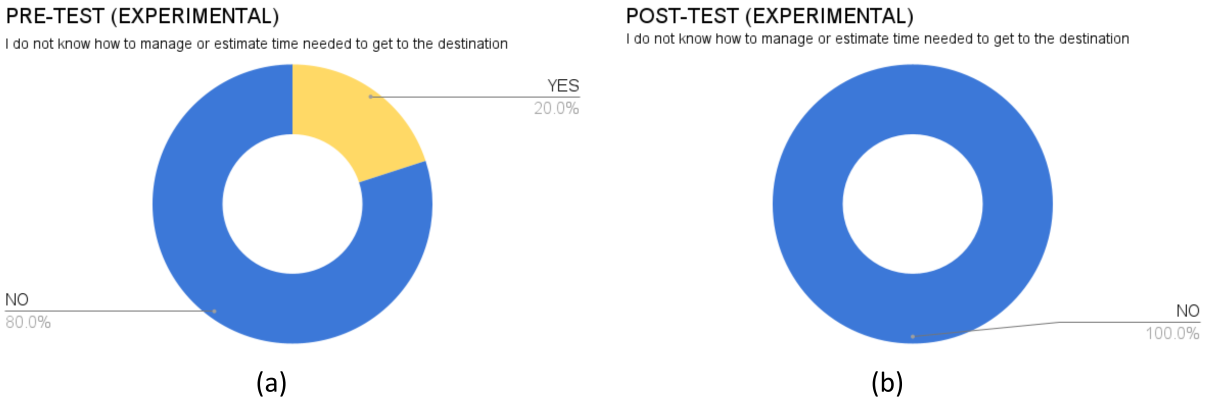


Fig. 11. (a) Pre-test experimental graph for question 3 and (b) Post-test experimental graph for question 3

5. Conclusion

In conclusion, this study looks into the significance of using game-based lessons to improve students' motivation and nurture basic skills needed to venture into programming. *Pantas.io*, developed using Unity as the platform and C# language as the tool to define the rules within the game is launched. Observations are made using the quasi-experimental approach, where a set of questions are given to the students pre- and post-test to observe the improvements. All in all, early observations have shown substantial improvements in terms of students' interest and enjoyment in learning programming, coordination and planning abilities, as well as time management skills. In an overarching view, the results unequivocally demonstrate a meaningful advancement in nurturing motivation and essential programming-related skills. These outcomes strongly support the core objective of the study, namely utilizing game-based lessons to enhance motivation and critical programming skills among primary school students. In summary, the initial findings underscore the positive impact of game-based lessons on both motivation and skill development in the context of programming education. This study successfully achieves its primary goal and lays a foundation for further research and implementation of game-based learning methods in primary school programming curricula. More observations and game improvements are needed to verify the positive impacts of the game-based lesson.

Acknowledgment

This research was funded by Geran Galakan Penyelidikan Universiti (GGPU, Grant No.: 2021 - 0078 - 107 - 01) under Universiti Sultan Pendidikan Sultan Idris (UPSI).

References

[1] Tangkui, Rayner, and Tan Choon Keong. "Enhancing pupils' higher order thinking skills through the lens of activity theory: Is digital game-based learning effective?." *International Journal of Advanced Research in Education and Society* 2, no. 4 (2020): 1-20.

[2] Mustapha, Nur Arfah, Hadijah Iberahim, and Syed Jamal Abdul Nasir Syed Mohamad. "Beyond the Norm: Leveraging Carrots and Sticks to Fuel Post-graduate Students' Enthusiasm in Class." *Semarak International Journal of Innovation in Learning and Education* 1, no. 1 (2024): 43-53.

[3] Ismail, Safinah, Aemy Elyani Mat Zain, Haslina Ibrahim, Nazneen Ismail, Nur Aisyah Abu Hassan, and Fatin Farzana Dass Meral. "Kepentingan Aplikasi Digital dalam Pembelajaran Anak Muda Era Industri 4.0: The Importance of Digital Applications in Young Children's Learning Industry Era 4.0." *Semarak International Journal of STEM Education* 1, no. 1 (2024): 28-38.

[4] Riza, Lala Septem, Erna Piantari, Enjun Junaeti, and Ichsan Seanaldi Permana. "Implementation of the Gamification Concept in the Development of a Learning Management System to Improve Students' Cognitive In

Basic Programming Subjects Towards a Smart Learning Environment." *ADI Journal on Recent Innovation* 5, no. 1 (2023): 43-53. doi: 10.34306/ajri.v5i1.902.

[5] Sun, Lihui, Linlin Hu, and Danhua Zhou. "Single or combined? A study on programming to promote junior high school students' computational thinking skills." *Journal of Educational Computing Research* 60, no. 2 (2022): 283-321. doi: 10.1177/07356331211035182.

[6] Zokhi, Aini Nurrasyidah Md. "Aplikasi Inovasi Q-Track Kit Dalam Proses Pengajaran dan Pembelajaran Bagi Modul Teoritikal: Innovative Application of Q-Track Kit in the Teaching and Learning Process for Theoretical Modules." *International Journal of Advanced Research in Future Ready Learning and Education* 27, no. 1 (2022): 20-29.

[7] Morgan, Alice, Dave Smaldone, Steven Selin, Jinyang Deng, and Marion Holmes. "Adding relevancy to STEM interest through adventure education: A mixed methods study." *Interdisciplinary Journal of Environmental and Science Education* 18, no. 4 (2022): e2294. doi: 10.21601/ijese/12214.

[8] Soumia, Y. A. M. O. U. L., O. U. C. H. A. O. U. K. A. Lynda, R. A. D. I. D. Mohamed, and M. O. U. S. S. E. T. A. D. Mohamed. "Implementing a serious game as a learner motivation tool." *Procedia Computer Science* 210 (2022): 351-357. doi: 10.1016/j.procs.2022.10.163.

[9] Altaie, Mayyadah A., and Dayang Norhayati Abang Jawawi. "Adaptive gamification framework to promote computational thinking in 8-13 year olds." *Journal of e-Learning and Knowledge Society* 17, no. 3 (2021): 89-100. doi: 10.20368/1971-8829/1135552.

[10] Cheng, Yu-Ping, Chin-Feng Lai, Yun-Ting Chen, Wei-Sheng Wang, Yueh-Min Huang, and Ting-Ting Wu. "Enhancing student's computational thinking skills with student-generated questions strategy in a game-based learning platform." *Computers & Education* 200 (2023): 104794. doi: 10.1016/j.compedu.2023.104794.

[11] Cruz-García, Iago, Juan Antonio Martín-García, Diana Pérez-Marín, and Celeste Pizarro. "Programming Teaching Experience for Boys and Girls following a Gamified Approach." In *SIIE*. 2020.

[12] Heljakka, Katriina, Pirita Ihamäki, Pauliina Tuomi, and Petri Saarikoski. "Gamified coding: Toy robots and playful learning in early education." In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 800-805. IEEE, 2019. doi: 10.1109/CSCI49370.2019.00152.

[13] Vahldick, Adilson, Paulo Roberto Farah, Maria José Marcelino, and António José Mendes. "A blocks-based serious game to support introductory computer programming in undergraduate education." *Computers in Human Behavior Reports* 2 (2020): 100037. doi: 10.1016/j.chbr.2020.100037.

[14] Videnovik, Maja, Elena Vlahu-Gjorgievska, and Vladimir Trajkovik. "To code or not to code: Introducing coding in primary schools." *Computer Applications in Engineering Education* 29, no. 5 (2021): 1132-1145. doi: 10.1002/cae.22369.

[15] Chen, Hsiu-Ling, and Cheng-Ting Wu. "A digital role-playing game for learning: Effects on critical thinking and motivation." *Interactive Learning Environments* 31, no. 5 (2023): 3018-3030. doi: 10.1080/10494820.2021.1916765.

[16] Byusa, Edwin, Edwige Kampire, and Adrian Rwekaza Mwesigye. "Game-based learning approach on students' motivation and understanding of chemistry concepts: A systematic review of literature." *Heliyon* 8, no. 5 (2022).

[17] Nand, Kalpana, Nilufar Baghaei, John Casey, Bashar Barmada, Farhad Mehdipour, and Hai-Ning Liang. "Engaging children with educational content via Gamification." *Smart Learning Environments* 6 (2019): 1-15. doi: 10.1186/s40561-019-0085-2.

[18] Schatten, Markus, and Marinela Schatten. "Gamification of game programming education: A case study in a croatian high school." In *Central European Conference on Information and Intelligent Systems*, pp. 13-18. Faculty of Organization and Informatics Varazdin, 2019.

[19] Tsai, Chun-Yen, Wen-Ling Shih, Fu-Pei Hsieh, Yun-An Chen, Chien-Liang Lin, and Hui-Ju Wu. "Using the ARCS model to improve undergraduates' perceived information security protection motivation and behavior." *Computers & Education* 181 (2022): 104449. doi: 10.1016/j.compedu.2022.104449.

[20] Mirzaei, Azizullah, Hanieh Shafiee Rad, and Ebrahim Rahimi. "Integrating ARCS motivational model and flipped teaching in L2 classrooms: A case of EFL expository writing." *Computer Assisted Language Learning* (2022): 1-30. doi: 10.1080/09588221.2022.2068614.

[21] Hsia, Chih-Hsien, Chin-Feng Lai, and Yu-Sheng Su. "Impact of using ARCS model and problem-based leaning on human interaction with robot and motivation." *Library Hi Tech* 40, no. 4 (2022): 963-975. doi: 10.1108/LHT-07-2020-0182.

[22] Cai, Xiaoyan, Zhiying Li, Jiahui Zhang, Mingmei Peng, Shuai Yang, Xiaoying Tian, Qiaohong Yang, and Fengxia Yan. "Effects of ARCS model-based motivational teaching strategies in community nursing: a mixed-methods intervention study." *Nurse Education Today* 119 (2022): 105583. doi: https://doi.org/10.1016/j.nedt.2022.105583.

[23] Gananjaya, Ivan, Jesse Owen Theodore Chandra, Johann Felix Alexander Christanto, Mochammad Haldi Widianto, and Jesslyn Audrey. "“A Lone Burglar” Stealth Game Development Using Rapid Application Development." In *2022 4th International Conference on Cybernetics and Intelligent System (ICORIS)*, pp. 1-5. IEEE, 2022. doi: 10.1109/ICORIS56080.2022.10031499.