Journal of ICT in Education (JICTIE)
ISSN 2289-7844 / 9 / 3 (Special Issue 1/2022: Student Engagement and EdTech) / 30-40

Learning Algorithm Concepts by Developing Them in Educational Gameplay: From the Perspective of College Students in Perak

Siti Sakinah Mohd Yusof[1], Azniah Ismail[2*], Nor Azah Abdul Aziz[2]

[1]*Computer Science Unit, Mathematics Department, Perak Matriculation College; bm-1621@moe-dl.edu.my*
[2]*Computing Department, Faculty of Art, Computing and Creative Industry, Universiti Pendidikan Sultan Idris;*
{azniah, azah}@fskik.upsi.edu.my

**correspondence author*

To cite this article (APA): Mohd-Yusof, S.S., Ismail, A., & Abdul-Aziz, N.A. (2022). Learning algorithm concepts by developing them in educational gameplay: From the perspective of college students in Perak. *Journal of ICT in Education*, *9*(3),30-40. https://doi.org/ 10.37134/jictie.vol9. sp.1.3.2022

To link to this article: https://doi.org/ 10.37134/jictie.vol9. sp.1.3.2022

Abstract

The purpose of this study was to determine whether the game developer tool has good capabilities to assist students in learning algorithm concepts and determine whether the game developer tool promotes a good student attitude. This research is based on a descriptive study design involving only one single sample. Quantitative data was collected through a questionnaire. The questionnaire is divided into 3 sections which are demographic information, tool capability, and student attitude. There are a total of 78 students from two modules (Physics and Biology), who took part in this research. The highest score for the tool capability section is "Scratch 3.0 help me achieve my objective to develop the educational game" while the highest score for the student attitude section is "I enjoy learning using Scratch 3.0". It is proven that learning algorithm concepts by developing them in educational gameplay help students to learn in an enjoyable environment.

Keywords: online learning; algorithm concept; game developer tool; Scratch 3.0; college students

INTRODUCTION

Programming is a subject that requires problem-solving strategies and involves a great number of programming logic activities that pose challenges for learners (Wang & Hwang, 2017). Problem-solving skills are paramount to university students while learning programming languages (Lawan, Abdi, Abuhassan & Khalid, 2019). Beginner programmers are facing many difficulties while learning to program (Mladenović, Boljat, & Žanko, 2018). Problem-solving techniques are one of the primary topics introduced to computing and information technology students in tertiary education (Basuhail,

Received: 1 February 2022; Accepted: 15 September 2022; Published: 1 December 2022

2019). Many limitations and problems such as learning to program, application of programming concepts, understanding of programs, and students' motivation surfaced (Marcolino, Schaefer, & Barbosa, 2017).

The use of digital platforms to solve problems is relatable to computational thinking as a set of cognitive and metacognitive strategies in which the learner is engaged in an active design and creation process and mobilized computational concepts and methods (Romero, Lepage, & Lille, 2017). Programming in block-based environments is a key element of introductory computer science (Grover & Basu, 2017). Programming games present these challenges as puzzles where computational thinking and coding skills have a fundamental role in successfully overcoming them (Arnedo-Moreno and D. García-Solórzano, 2020). Games have important motivational power (Figueiredo & García-Peñalvo, 2020).

Computational thinking (CT) has been used to teach problem-solving skills and programming education in the recent past. A CT approach can be explored for designing a smart learning environment to support students in learning computer programming (Agbo, Oyelere, Suhonen & Adewumi, 2019). Novice programmers seem to learn basic programming skills amazingly fast by using visual programming environments like Scratch or Snap (Talbot, Geldreich, Sommer & Hubwieser, 2020). Scratch 3.0 new programming blocks offer greater experimentation for novices and professionals alike (Bau, Gray, Kelleher, Sheldon & Turbak, 2017). It is more interesting to develop students' logical thinking skills and problem-solving skills through programming approaches or CT (García-Peñalvo & Mendes, 2018). The students make use of the CT while using the algorithms to solve a problem and while solving the problem with calculation. One of the important components of CT is Algorithmic Thinking (Korkmaz, Çakir, & Özden, 2017).

PROBLEM STATEMENTS

Algorithm concepts are very hard to be grasped without enough practicing, enhancing the classical introduction to programming courses through real-life problem-based game development with a Scratch programming environment potentially improves the learners' programming skills and motivation (Topalli, & Cagiltay, 2018). Most instructors agree that there is a problem when teaching programming and many students are unable to understand programming logic so there must be a method to encourage them (Elshiekh & Butgerit, 2017). Digital games possess important elements for learning programming, such as incremental challenges, personalized immediate feedback, and visualization of the consequences of errors (Gomes, Falcão & Tedesco, 2018). However, we have to be more selective in choosing tools that can help not only with the learning but also with student engagement, which allows students to work on their own while motivating them to complete the tasks. We are keen to know tools that can serve such purposes.

Based on Jafry (2017), the first problem is that there is no pedagogical integration in e-learning. The second problem is that there should be an appropriate approach to be used in the e-learning environment so that the R&D process has a significant impact on student achievement and not just moving from the traditional form to the digital form (Jafry, 2017). Visual coding languages consist of preset blocks that can be dragged together based on their shapes and desired outcomes. This block-based structure saves students a lot of frustration because they are not discouraged by syntax errors (Sáez-López et al., 2016). Matriculation Programme Semester Examination (PSPM) is held at the end of each semester of the study session. Examinations are conducted within 2 hours of each course. Shown below in Table 1 are findings from six reports of PSPM papers for Computer Science Subject from Bahagian Matrikulasi Kementerian Pendidikan Malaysia (BMKPM) that showed students lost many marks for not mastering the basics of algorithm building.

Table 1: Candidate's report for computer science PSPM paper

No.	Year	Description
1.	2013	Students have difficulties answering problem-solving questions. Students have difficulties in the application of iteration control structure.Weakness in understanding arithmetic questions.
2.	2014	Students failed to differentiate between a syntax error and a logic error.
3.	2015	Students are less proficient in programming. Students are weak in tracing output and did not understand the purpose of the process in the program given.
4.	2017	Students are less proficient in the iteration control structure. Students did not apply the correct arithmetic operator.
5.	2018	Students did not use the correct form or programming logic in pseudocode. Students are less proficient in programming resulting in syntax errors.
6.	2021	Students have trouble writing the answers to the output for the calculation process correctly. Students also fail to identify the correct control structure and are unable to provide the correct inputs, processes, and outputs.

A group of lecturers in a college in Perak has decided to use the learning by doing approach for their students in their Algorithm course. These lecturers purposely chose educational mini-game projects as the project theme and Scratch 3.0 for the game developer tool. Students are required to learn and apply algorithm concepts as much as they can as long as the concepts are suitable for their gameplay. They have to design and develop a few different games individually until complete within 14 weeks.

Received: 1 February 2022; Accepted: 15 September 2022; Published: 1 December 2022

We have observed a few of the project samples taken from the lecturers and we literally found that the students have applied suitable algorithm concepts to different gameplay. For example, one student had designed and developed a game called Math Ninja that asks its player to slash a fruit by entering the correct value that can solve the math problem given. The math question varies randomly from all basic math operations (addition, subtraction, multiplication, and division). The game will stop after the third failed attempt and the player's score will be shown on the screen. Algorithm concepts that have been used in this game are decompose, pattern, abstraction, algorithm, logical reasoning, and evaluation.

Figure 1: Math ninja

Another student had designed and developed a game named Angry Math. The game also asks its player to enter the correct value for each math problem given. However, the game has four levels of difficulties starting from level 1: addition, level 2: subtraction, level 3: multiplication, and level 4: division. The game will stop after the player lost three lives and the player's score will be shown. It is observed that the algorithm concepts that have been used in this game are decompose, pattern, abstraction, algorithm, logical reasoning, and evaluation.

Figure 2: Angry math

A different student had designed and developed games that ask the player to solve math problems. The player can choose from three levels of difficulties: easy, medium, and hard. Additionally, the player can also choose from four basic math operations i.e., addition, subtraction, multiplication, and division. Interestingly, the player will be competing with time indicated by a snail race. The game will stop after the player lost in the race and the score will be shown at the end of the game. From these examples, the students seem to gain a good understanding of some of the important algorithm concepts and were able to apply the concepts in their gameplay. This shows the ability of the game developer tool to help assist the students' understanding of the concepts. In this study, we describe further the performance of the game developer tool from the students' perspectives.

Figure 3: Math snail race

OBJECTIVE

The purpose of this study was:

1. To determine whether the game developer tool has good capabilities to assist student learning.
2. To determine whether the game developer tool promotes a good student attitude.

METHODOLOGY

This research is based on a descriptive study design involving only one single sample. Quantitative data was collected through a questionnaire.

Participants

A total of 78 students in a matriculation college in Perak were involved in this study. The total number of participants was 78 students from 9 classes (6 Module 2 classes and 3 Module 3 classes). 27 participants were male (27%) and 57 participants were female (73%). 51 participants were from Module 2 (65%) and the other 27 participants were from Module 3 (35%). The students had to take the Algorithm course for the whole semester. They used Scratch 3.0 as their game development tool to build several educational games individually within 14 consecutive weeks. A total of 234 educational game projects were collected by their lecturer at the end of Week 14. The ratio of the number of students to the number of games collected was 1:3.

Research Design

The study took place after the completion of the educational game projects. The students were given an online questionnaire for them to answer. The questionnaire was given by the researchers through Google Form. Their class lecturers forwarded the Google Form link through the students' WhatsApp group. Once data were collected, the data were analysed statistically.

Instruments

The online questionnaire was built using Google Form. It consists of three sections: Section A - demographic information (items no. 1 to 4), Section B – tool capability (items no. 5 to 8), and Section C - student attitude (items no. 9 to 13). Section A includes race (item no.1), gender (item no.2), program module (item no.3), and tutorial class (item no.4). The item distribution and item code for Section B and Section C are presented in Table 2. This questionnaire utilized a 4-point Likert scale to get specific responses (on agreement) without the 'neutral' option. The scale ranges from 1 (totally disagree), 2 (totally agree), 3 (agree), and 4 *(*totally agree). The reliability statistic for this questionnaire is r=0.8 so it is proven that it is well constructed and can be used to gauge the respondents' opinions.

Table 2: Questionnaire items and description

Section	Item no	Item code	Description
B TOOL CAPABILITY	5	B1	Scratch 3.0 help me achieve my objective to develop the educational game.
	6	B2	Blocks with different colour schemes in Scratch 3.0 assist me to determine different algorithm concepts.
	7	B3	Scratch 3.0 is suitable for my game development project.
	8	B4	Scratch 3.0 is easy to use.

Table 2 (continued)

C STUDENT ATTITUDE	9	C1	Scratch 3.0 makes me interested in learning algorithms (programming).
	10	C2	Scratch 3.0 motivates me to concentrate on working on my project.
	11	C3	Scratch 3.0 helps to improve my understanding of different algorithm concepts.
	12	C4	Scratch 3.0 promotes interaction between me and my lecturer.
	13	C5	I enjoy learning using Scratch 3.0.

Data Analysis Design

Data were analysed statistically. Section A is demographic information hence only a cross-tabulation table containing frequency and percentage was created. Section B focuses on Scratch 3.0 capability whereas Section C focuses on student attitude towards Scratch 3.0 as their game developer tool. These two sections only collect 4-point Likert-scale agreements. Thus, we only find the number of times the agreement occurs (Mode) and the average agreement response (Mean). The mode will tell the most common response to each statement while the mean will give the overall average response.

RESULTS

Demographic Information

Table 3 shows that the total number of participants was 78 students. 27 participants were male (27%) and 57 participants were female (73%). 51 participants were from Module 2 (65%) and the other 27 participants were from Module 3 (35%).

Table 3: Crosstabulation table between gender and program module.

Program Module	Male		Female		TOTAL	
	Freq.	%	Freq.	%	Freq.	%
Module 2 (Physics)	18	23 %	33	42 %	51	65%
Module 3 (Biology)	3	4%	24	31 %	27	35%
TOTAL	21	27 %	57	73 %	78	100.0

The result of Table 4 is divided into 2 sections. The first section is Section B: Tool Capability. It started with item no. 5 (item code B1), "*Scratch 3.0 help me achieve my objective to develop the* *educational game*". The mode score is 3 while the median score is 3.45. Then, item no. 6 (item code B2), "blocks with different colour schemes in Scratch 3.0 assist me to determine different algorithm concepts". The mode score is 3 while the median score is 3.44. Next, item no. 7 (item code B3), Scratch 3.0 is suitable for my game development project. The mode score is 3 while the median score is 3.36. Finally, item no. 8 (item code B4), Scratch 3.0 is easy to use. The mode score is 3 while the median score is 3.38.

The second section is Section C: Student Attitude. It started with item no. 9 (item code C1), Scratch 3.0 make me interested in learning algorithms (programming). The mode score is 4 while the median score is 3.47. Then, item no. 10 (item code C2), Scratch 3.0 motivate me to concentrate on working on my project. The mode score is 3 while the median is 3.4. Next, item no. 11 (item code C3), Scratch 3.0 helps to improve my understanding of different algorithm concepts. The mode score is 3 while the median score is 3.44. After that, item no. 12 (item code C4), Scratch 3.0 promotes interaction between me and my lecturer. The mode score is 3 while the median score is 3.44. Lastly, item no. 13 (item code C5), I enjoy learning using Scratch 3.0. The mode score is 4 while the median score is 3.5.

Table 4: Questionnaire items and description

Section	Item no	Item code	Description	Mode	Mean
B TOOL CAPABILITY	5	B1	Scratch 3.0 help me achieve my objective to develop the educational game.	3	3.45
	6	B2	Blocks with different colour schemes in Scratch 3.0 assist me to determine different algorithm concepts.	3	3.44
	7	B3	Scratch 3.0 is suitable for my game development project.	3	3.36
	8	B4	Scratch 3.0 is easy to use.	3	3.38
C STUDENT ATTITUDE	9	C1	Scratch 3.0 makes me interested in learning algorithms (programming).	4	3.47
	10	C2	Scratch 3.0 motivates me to concentrate working on my project.	3	3.4
	11	C3	Scratch 3.0 helps to improve my understanding of different algorithm concepts.	3	3.44

Table 4 (continued)

Section	Item no	Item code	Description	Mode	Mean
	12	C4	Scratch 3.0 promotes interaction between me and my lecturer.	3	3.44
	13	C5	I enjoy learning using Scratch 3.0.	4	3.5

DISCUSSION

The first purpose of this study was to determine whether the game developer tool has good capabilities to assist student learning. Based on the result of Table 4, the highest mean and median score (3 and 3.45 each) is from item code B1, Scratch 3.0 help me achieve my objective to develop the educational game. Then, the second-highest score (3 and 3.44 each) is from item code B2, blocks with different colour schemes in Scratch 3.0 assist me to determine different algorithm concepts. Next, the third-highest score (3 and 3.38 each) is from item code B4, Scratch 3.0 is easy to use. Finally, the fourth-highest score (3 and 3.36 each) is from item code B3, Scratch 3.0 is suitable for my game development project. So, it is proven that the game developer tool has good capabilities for students learning algorithm (programming) concepts.

The second purpose of this study is to determine whether the game developer tool promotes a good student attitude. Based on the result in Table 4, the highest mean and median score (4 and 3.5 each) is item C5, I enjoy learning using Scratch 3.0. Then, the second-highest mode and median score (4 and 3.47 each) is item C1, Scratch 3.0 makes me interested in learning algorithms (programming). Next, the third-highest score (3 and 3.44 each) is item code C3), Scratch 3.0 helps to improve my understanding of different algorithm concepts. After that, the fourth-highest mean and mode (3 and 3.44 each) is item code C4, Scratch 3.0 promotes interaction between me and my lecturer. Lastly, the fifth highest mode and median score (3 and 3.4 each) is item code C2, Scratch 3.0 motivates me to concentrate working on my project. The mode score is 3 while the median is 3.4. So, it is proven that the game developer tool promotes a good student attitude while learning algorithm (programming). Some of the student’s comments are shown in Table 5 below.

Table 5: Students comments

No.	Comment
1.	*“The use of Scratch in tutorials can also attract students to learn in a happy state.”*
2.	*Scratch is very fun because it can help students to understand the concept of algorithms easily.*
3.	*The use of Scratch can help attract students to learn it.*
4.	*Interesting and fun.*
5.	*Scratch really helped me.*
6.	*Can recognize the use of algorithms in computers.*
7.	*It can attract my attention to learn it.*
8.	*Create learning modules in manual form as well.*
9.	*Effective learning outcomes.*

IMPLICATION OF FINDINGS

This research finding is important for Computer Science and programming curricula. It is also beneficial to all matriculation colleges and the Ministry of Education. This research should be done on a bigger scale such as involving more matriculation colleges instead of only one. It is proven that the game developer tool has good capabilities for student learning and the game developer tool also promotes a good student attitude while learning algorithms (programming). It is similar to the results in Ng & Cui (2020)'s study, the block-based programming environment supported the students' modeling and algorithmic thinking, and the students utilized computational abstractions in the form of variables, and employed different approaches, to formulate mathematical models in a programming context. Algorithm concepts that have been used in this game are decomposed pattern, abstraction, algorithm, logical reasoning, and evaluation.

CONCLUSION

In conclusion, the edutainment element in Scratch provides an enjoyable learning experience for mature students which is like the result in Chis, Moldovan, Murphy, Pathak & Muntean, (2018) study. Game-based learning, problem-based learning, visual programming, and projects are technologies that can potentially help learners to perform better in the introduction to programming courses, in turn affecting their performances (Elshiekh & Butgerit, 2017). Students responded by stating that the use of Scratch is interesting and fun which can attract students to learn in a happy state, it also helps students to understand the concept of algorithms easily. The students also suggest that learning modules in manual form should be created as well.

REFERENCE

Agbo, F. J., Oyelere, S. S., Suhonen, Jarkko, Adewumi, S. (2019). A Systematic Review of Computational Thinking Approach for Programming Education in Higher Education Institutions. *Koli Calling '19: Proceedings of the 19th Koli Calling International Conference on Computing Education Researc*h. November 2019 Article No.: 12 Pages 1–10. https://doi.org/10.1145/3364510.3364521.

Arnedo-Moreno, J., & García-Solórzano, D. (2020, November). Programming is fun! a survey of the Steam digital distribution platform. In *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)* (pp. 1-4). IEEE. https://doi.org/10.1109/CSEET49119.2020.9206214.

Basuhail, A. (2019). E-Learning objects designing approach for programming-based problem solving. *International Journal of Technology in Education, 2*(1), 32-41.

Bau, D., Gray, J., Kelleher, C. L., Sheldon, J., & Turbak, F. (2017). Learnable programming: blocks and beyond. Communications of the ACM. May 2017. https://doi.org/10.1145/3015455.

Chis, A., Moldovan, A., Murphy, L., Pathak, P., & Muntean, C. (2018). Investigating flipped classroom and problem-based learning in a programming module for computing conversion course. *Journal of Educational Technology & Society, 21*(4), 232-247.

Elshiekh, R., & Butgerit, L. (2017) Using gamification to teach students programming concepts. *Open Access Library Journal*, *4*(8), 1-7. https://doi.org/10.4236/oalib.1103803.

Figueiredo, J., & García-Peñalvo, F. J. (2020, April). Increasing student motivation in computer programming with gamification. In *2020 IEEE Global Engineering Education Conference (EDUCON)* (pp. 997-1000). IEEE. https://doi.org/10.1109/EDUCON45650.2020.9125283.

García-Peñalvo, F. J., Mendes, A. J. (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.

Gomes, T. C. S., Falcão, T. P., & Tedesco, P. C. D. A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. *International Journal of Child-Computer Interaction*, *16*, 77-84. https://doi.org/10.1016/j.ijcci.2017.12.005.

Grover, S., & Basu, S. (2017, March). Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 267-272). https://doi.org/10.1145/3017680.3017723.

Jafry Mohd Daud. (2017). PhD Thesis. Kesan Kaedah Pengajaran dan Pembelajaran berasaskan Integrasi Model Salmon dan Teori Konstruktivisme 5 Fasa Needham terhadap Pencapaian dan Sikap Pelajar dalam E-Pembelajaran Subjek Sains Komputer di Kolej Matrikulasi Perak. Universiti Pendidikan Tanjong Malim, UPSI.

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, *72*, 558–569. https://doi.org/10.1016/j.chb.2017.01.005.

Lawan, A. A., Abdi, A. S., Abuhassan, A. A., & Khalid, M. S. (2019, April). What is difficult in learning programming language based on problem-solving skills?. In *2019 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 18-22). IEEE. https://doi.org/10.1109/ICOASE.2019.8723740.

Marcolino, A. S., Santos, A., Schaefer, M., & Barbosa, E. F. (2017). Towards a catalog of gestures for m-learning applications for the teaching of programming. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE. https://doi.org/10.1109/FIE.2017.8190495.

Mladenović, M., Boljat, I. & Žanko, Ž. Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Education and Information Technologies*, *23*(4), 1483-1500. https://doi.org/10.1007/s10639-017-9673-3.

Ng, O.L., Cui, Z. (2020). Examining primary students' mathematical problem-solving in a programming context: towards computationally enhanced mathematics education. *ZDM–Mathematics Education*, *53*(4), 847-860.. https://doi.org/10.1007/s11858-020-01200-7.

Romero, M., Lepage, A. & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*(1), 1-15. https://doi.org/10.1186/s41239-017-0080-z.

Sáez-López, J.M., Román-González, M., & VázquezCano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, *97*, 129-141. https://doi.org/10.1016/j.compedu.2016.03.003

Talbot, M., Geldreich, K., Sommer, J., & Hubwieser, P. (2020). Re-use of programming patterns or problem solving? representation of scratch programs by TGraphs to support static code analysis. In *Proceedings of the 15th Workshop on Primary and Secondary Computing Education* (pp. 1–10) https://doi.org/10.1145/3421590.3421604.

Topalli, D., & Cagiltay, N.E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education*, *120*, 64-74. https://doi.org/10.1016/j.compedu.2018.01.011.

Wang, X.M., & Hwang, G.J. (2017). A problem posing-based practising strategy for facilitating students' computer programming skills in the team-based learning mode. *Education Technology Research and Development*, *65*(6), 1655–1671 (2017). https://doi.org/10.1007/s11423-017-9551-0.