

Digital Game Based Learning of Stack Data Structure Using Question Prompts

<https://doi.org/10.3991/ijim.v13i07.10778>

Rosni Ramle ^(✉), D'oria Islamiah Rosli, Shelena Soosay Nathan, Mazniha Berahim
Universiti Tun Hussein Onn Malaysia, Johor, Malaysia
rosni@uthm.edu.my

Abstract—Data structure and algorithm is an important course in computer science and information technology programs, applied in almost all courses. Failure to master it will affect student's academic performance during study, getting job interviews, passing job interviews, and create an inefficient information technology worker. However, learning data structure is a worldwide problem because of its complex nature. Gameful visualization of data structures' algorithms has been gaining momentum as it resulted in increased motivation, engagement and learning outcome. But effectiveness of game-based learning could be hindered if improper learning strategies used. Instructional scaffold in game-based learning in the form of question prompts have been found to be the most effective way to scaffold self-learn in computer-based learning. Thus, a game-based learning of stack data structure using question prompts was designed, developed and tested based on an adopted model to help students understand the algorithms of stack's insert and delete operations for array implementation with gameplay that could create meaningful learning. A pre-game and post-game test was conducted to compare students' performance on the topic. Results indicated a generally positive outcome.

Keywords—Algorithm, data structure, game-based learning, stack

1 Introduction

Data structure are ways of organizing data in computer [1] which can be operated on by using various algorithm [2]. It is a core course in Information Technology (IT) programs in tertiary education as it is a fundamental knowledge applied in various areas [3]. Unfortunately, it is commonly difficult to comprehend [4], making teaching and learning challenging [5]. The main factor for low performance is the nature of the course itself, where students are required to learn the abstract concepts of various data structures, with each data structure having various operations, and each operation having its own algorithm. Understandably, students often unable to remember, let alone understand them all, hence mixing them up during informal and formal assessment, negatively affecting their overall mark and grade. Students lose interest in learning [6] and teaching requires much explaining and illustrating. Static [1], dynamic [7], and interactive visualization [8] failed to engage learners continuously, fre-

quently, and actively [9]. Since [4] did a comprehensive study on how computer games can be used in algorithm learning and visualization, this research area has flourished as it has shown promising results.

Game-based learning (GBL) combine game elements with defined learning outcomes in gameplay [10]. It is popular because of advancements in computer games technology [11], increased number of people playing games, and the benefits of game itself which makes learning interesting [12] and easier thus more effective [13]. Data structure and algorithm have many concepts, elements, and rules that are suitable for designing and developing educational games [4]. GBL connect abstract concepts in algorithms with objects from real world [6], making understanding easier. However, effectiveness of games on learning is still unclear [14], and may be compromised [15] if improper learning strategies used. Therefore, [16] suggested inclusion of instructional support in GBL environments to enable students' to apply game knowledge to real life problems.

One type of external scaffolds in GBL is question prompts, which have been found to improve learning outcome [15]. [17] subcategorize cognitive learning outcomes into three; declarative knowledge, procedural knowledge, and strategic knowledge. Data structure learning involves algorithmic problem solving where students need to know how to perform data structure operations according to their respective algorithm. Hence the cognitive learning outcome of data structure learning is procedural knowledge, which prompt can help apply [18].

A stack data structure has a last-in-first-out (LIFO) structure where values can be inserted (pushed) onto a stack or removed (popped) from the stack, and the value that was pushed on the stack most recently is the first to be popped from it. Common stack operations are pop, push, peek (get the top element without removing it), isEmpty (checks whether stack is empty or not), and isFull (checks whether stack is full or not). One way to implement a stack is using array-based. Stack is one of the most covered data structure in curriculum [19] and are often used when there are interrupts to handle, or when having recursive functions, or even when constructing a rudimentary AI for games [20].

This paper presents an alternative way of designing data structure GBL, which is through the use of question prompts, to create a gameplay that leads to meaningful learning of data structure algorithms. Stack data structure GBL with gameplay that simulates algorithm of array implementation of insertion and deletion operation using LIFO principle was designed, developed, and tested.

2 Literature Review

The use of e-learning motivates user to learn [21] thus could be adopted at higher education institutions [22] and can be integrated across many disciplines, such as language [23]. [19] created Stack Ship Activity where user navigates a space ship through space by touching desired values to Push a number in a stack (the main body of the ship) and tapping a POP button to Pop a value from the stack until the stack overflows or underflow. An undergraduate from Winston-Salem State University

created a stack game where a robot has to walk home facing different obstacles on land and sea that challenge its knowledge of the Stack data structure (by pushing and popping blocks) after its spaceship crashes [24]. [25] asked students to design and implement a program that allows them to play a simplified version of a card game with the computer. A shuffled deck of cards is dealt (push) between the computer and the player's draw piles. Each draw pile begins with (isFull) 26 cards and each winning pile starts out empty. For each round, the computer and player each place the top card from (pop) their draw pile onto (push) the center of the table. If the player's card has a higher value, the two cards are taken (pop) and placed at the top (push) of the player's winning pile, vice versa. In the case of a tie, the two cards are left in the middle of the table, to be picked up by the next round's winner. When one of the two players runs out of cards (isEmpty) in their draw pile, the game is over. The player with the highest total number of cards in their two piles (draw and winning) is declared the winner.

[26] separates question prompts into procedural, elaboration and reflection, where each serve different purposes for learners. Reflection prompts can be broken down further into knowledge prompts where learners reflect on the conceptual understandings of the underlying knowledge within game [27] and application prompts where game knowledge is applied outside of game [28]. Prompts have been found to scaffold students' high-order thinking in various domains and contexts, and help in knowledge integration [29], focus, articulating thoughts, reflection of learning processes [26] and improve self-regulated learning competence [30].

3 Methodology

There were four phases in conducting this research. In the first phase, a literature review was done on data structure learning, GBL, and question prompts. The second phase was designing the stack game based on identified constructs. This research is based on a game model by [17] where there are three phases; Input, Process, and Outcome. The objective is to design an instructional program that incorporates certain characteristics of games that trigger a cycle that includes user judgments or reactions, user behaviours, and further system feedback. The extent to which this cycle results in engagement in game play which leads to the achievement of learning outcomes depends on the success of pairing instructional content with appropriate game features. The model was combined with data structure game design by [4] which consists of general game design elements, game appearance design, and game mechanics design.

3.1 First-general game design elements

1. **Game idea:** Stack and unstack array of stones.
2. **Game goal:** Simulate insertion (push) and deletion (pop) operation of array implementation of stack data structure using LIFO principle.
3. **Game topic:** Array implementation of stack requires checking whether stack is full before each push (see Figure 1) and whether stack is empty before each pop (see

Figure 2). Push is not allowed if stack is full and pop is not allowed if stack is empty.

4. **Game start:** Head-Up-Display (HUD) on top of screen shows Maximum Player Lives (5), Minimum Player Score (0) and Maximum Time (100 seconds). Text narratives of story overview displayed under HUD. Game starts after story overview ended. Four empty slots presented.
5. **Game end:** Game ends when player finish within allocated time or when life or time equal to zero.
6. **Game milestone events:** Score increased by 10 for each correct action and Lives reduced by one for each wrong action.
7. **Game exit:** Game can be exited from the close (X) button on the game window. User will be asked for confirmation.

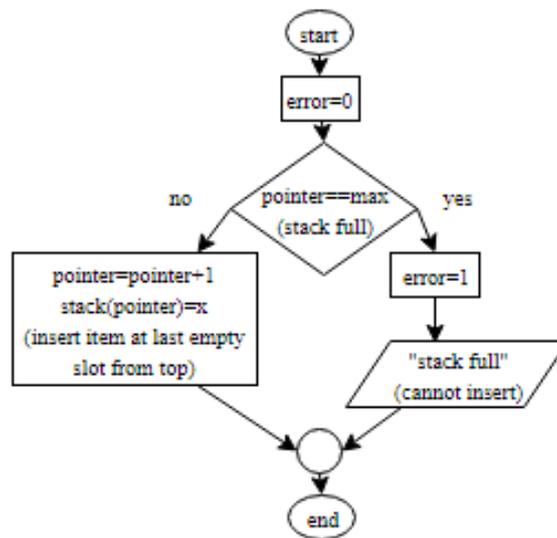


Fig. 1. Push algorithm for array implementation of stack

Definitions: **max:** maximum elements in stack. **pointer:** pointer to last element to be pushed or next element to be popped. **stack:** stack array. **x:** value to be pushed or popped

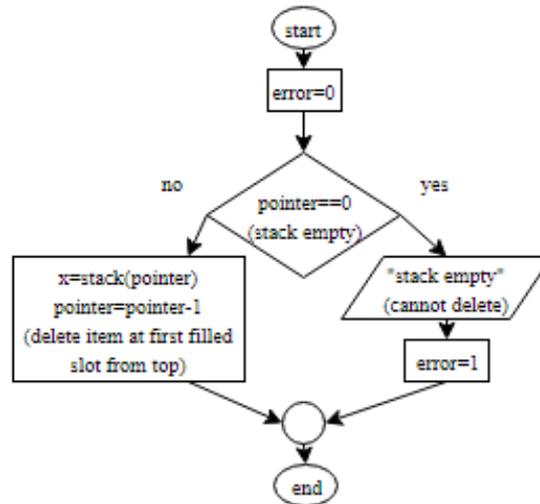


Fig. 2. Pop algorithm for array implementation of stack

Definitions: **pointer:** pointer to last element to be pushed or next element to be popped. **stack:** stack array. **x:** value to be pushed or popped.

3.2 Second-algorithm game appearance design

Graphics design

1. **Game Graphic Items:** 2D sprites, each with name, texture, size, and position.
2. **Node:** Item used to visualize one node of the algorithm data structure, which is stone.
3. **Collections:** A set of similar nodes organized according to specific rules. A collection is used to visualize an algorithm data structure, which is stack of stones.
4. **Playing tools:** Items used to play game, which are stone and bomb.
5. **Buttons:** Non animated graphic objects used in screens design. Yes, No, and Arrow buttons used during game.

Game assets

6. **Texture:** Stone, Bomb, Buttons.
7. **Sounds:** Background audio during game and sound effects during game opening, mouse click, correct action, incorrect action, game won, and game lost.

3.3 Third-Algorithm Game Mechanics Design

1. **Input Design** handles game input from mouse. Each player input event in game has a feedback in terms of Score, Lives, or sounds.
2. **Game Properties** are parameters initialized at start of game; Maximum Player Lives, Maximum Level Time, and Minimum Player Score. Other game attributes

are calculated and displayed to show game current state; Player Remaining Lives, Current Level Timer, and Current Player Score.

3. **Gameplay** is responsible for implementing playing rules of game according to visualized algorithm behavior. General gameplay of the game is depicted in Figure 3. While levelTimer and playerLives are more than zero, and while tasks are not finished, each incorrect action will decrease playerLives, while each correct action will increase playerScore. If player finishes game within given levelTimer and playerLives, player wins, vice versa.

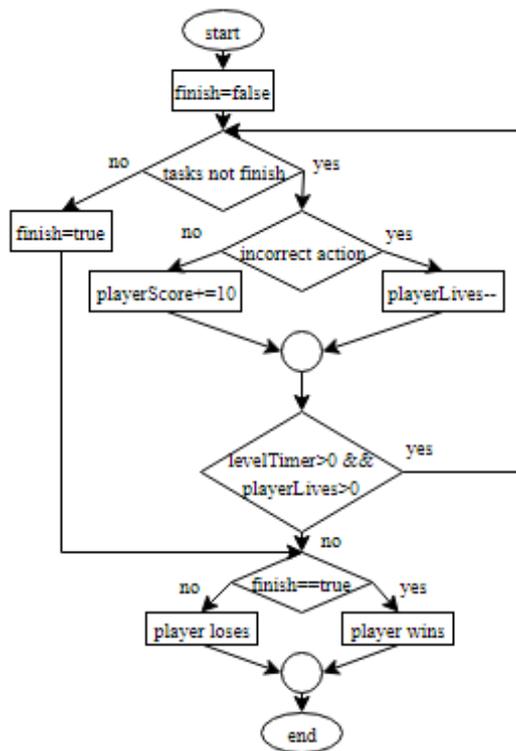


Fig. 3. General algorithm of gameplay

Specific playing rules that simulate the algorithm steps (see Figure 4) are as follows:

Screens design

1. **Title** screen displays the game title which is Train to Kluang, inspired by South Korea’s blockbuster movie titled Train to Busan, with similar storyline; main character running away from zombies.
2. **Main menu** screen displays game main menu options and handles player choices. It is separated into three sections; Notes, Demo, and Play.
3. **Start** screen displays story overview, which can be skipped.

4. **Play** screen displays game for player to play, including HUD that shows game properties; Player Remaining Lives, Current Level Timer, and Current Player Score.
5. **Exit** screen allows player to stop and end game at any time.
6. **Won Game** screen displays won message when player wins (player finish within allocated time).
7. **Lost Game** screen displays lost message when player loses game. There are two; lost because of no lives left and lost because time is up.
8. **Game Demo** screen displays self-running demo of game.

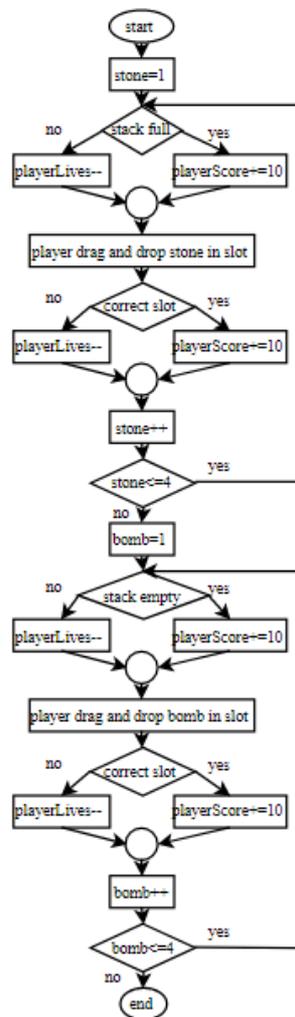


Fig. 4. Algorithm of array gameplay

Player need to stack (push) and unstack (pop) four stones. Before each stacking, player will be asked whether the stack is full (see Figure 5). Player then drag and drop stone in slot (see Figure 6).



Fig. 5. Checking whether stack is full before each push



Fig. 6. Push stones (data) in slots (stack)

Before each unstacking, player will be asked whether the stack is empty (see Figure 7). Player then drag and drop bomb in slot (see Figure 8).



Fig. 7. Checking whether stack is empty before each pop



Fig. 8. Pop stones (data) from slots (stack)

The series of questions are actually a replication of the thinking process that a student should undergo to solve problems regarding array implementation of stack data structure. Because game can be used as a training as it can be repetitively played, the idea here is for the students to ingrain the problem solving process in their mind until the point that they are able recreate (self-ask) the series of questions (process) when solving problems of similar nature.

The third phase of the research is the development of a prototype based on the design. Adobe Photoshop was used for graphic creation and editing, while Adobe Flash was utilized to integrate the multimedia elements text, graphic, audio, animation and video.

In the fourth phase of the research, the game was tested to measure its effect on learning outcome. A sample consisting of 29 first-year Diploma in Information Technology students who had taken the course in the previous semester was chosen. Players were required to answer a pre-game and post-game test that have the same stack topic questions before and after playing. Players were given time to study before playing. They were then separated into three groups to play the game.

If the game is successful, the gameplay triggers repeated cycles of user judgments (e.g., enjoyment) which leads to behaviour (play until won or play until lost or replay), and feedback (HUD, correct and incorrect audio, won and lost screens). Debriefing is the review and analysis of events that occurred in the game whose process allows players to transform game events into learning experiences. Successful debriefing leads to the achievement of cognitive learning outcome of the game which is the procedural knowledge (knowledge about how to perform a task) whereby the players demonstrate the ability to apply knowledge of stack to solve a specific problem.

4 Results and Discussions

Results from testing are summarized in Table 1. There are eight multiple-choice questions regarding stack data structure in pre and post-game tests. The first question asked players to choose the structure of stack. Second and third questions require players to select the name of the insert and delete operations in stack correspondingly. Fourth and fifth questions need players to pick stack operations in array implementation that occur before push and pop operations respectively. Sixth and seventh questions want players to determine the location of a new node during push and which node is removed during pop figuratively. Eighth question inquire whether there is a limit on the number of nodes that can be pushed in array implementation of stack.

Table 1. Comparison between pre-game and post-game test

Question number	Question	Correct		Change
		Pre	Post	
1	Principle	13	27	+
2	Push	26	28	+
3	Pop	27	26	-
4	Check full	14	17	+
5	Check empty	12	19	+
6	Push (array)	26	27	+
7	Pop (array)	17	22	+
8	Number of nodes (array)	25	26	+

Based on the results, all questions but one showed positive changes where number of correct answers after playing game (post) is higher than before (pre). Only one

question had negative change where number of correct answers decreased after playing game. But this result is not a major problem as the difference is only one.

To evaluate game usability, players answered 14 five-point Likert scale questions ranging from strongly disagree to strongly agree which consists of three sections; user interface, user friendliness, and interactivity. Results are summarized in Table 2.

Table 2. Game usability

Question number	Question	1	2	3	4	5
1	Effective use of screen area	0	0	11	16	2
2	Consistent layout	0	1	11	16	1
3	Attractive interface	0	2	6	19	2
4	Active buttons clearly highlighted	0	3	6	15	5
5	Easy to use	0	0	4	24	1
6	Easy navigation	0	1	10	16	2
7	Provision of Exit and Menu	0	0	9	17	3
8	Clear instructions	0	1	17	9	2
9	Meaningful icons	0	2	10	16	1
10	Consistent and readable text	0	1	7	15	7
11	Creative animations	0	0	6	15	8
12	Video enhances instructional effect	0	0	10	15	4
13	Graphics enhance instructional effect	0	0	13	14	2
14	Clear audio	0	0	4	20	5
Average (%)		0.00	2.46	30.54	55.91	11.08

- 1 - Strongly Disagree (%)
- 2 - Disagree (%)
- 3 - Neutral (%)
- 4 - Agree (%)
- 5 - Strongly Agree (%)

Players’ opinions were mostly positive for all questions. But there are some issues with layout (question 2), interface (question 3), button (question 4), navigation (question 6), instruction (question 8), icon (question 9), and text (question 10). In term of layout, players may felt it is inconsistent as there are many screens in the game application, each with a different layout. We admit that attractiveness of interface may be low, as the graphics were not designed by professional graphic designers, but by final-year diploma students with limited skills. Therefore we understand when players suggested a “more creative” interface with “more animation”, such as “motion for zombies”. We are not sure why some players have issue with active buttons, as all buttons in the game increase in size or changes colour when mouseover, clearly highlighting them. We think that the game has a simple navigation, but may be some players were confused with the icons used. Some players thought that the instructions were general, making it ambiguous, and suggested for them to be “more detailed”. In our opinion, fonts used in the game are consistent and readable, as the size is big and “colourful”. Thus we are not sure why a player had an issue with it. Players also recommended an increase in Time and Lives. Provision for “Level” is proposed to add “difficulty” for “more challenge”.

5 Conclusion

This research attempted to simplify learning of array implementation of stack data structure operations through game-based learning with question prompts. Educators could use the game to diversify teaching method and students can use it to self-assess understanding. Testing calls for improvements in terms of game design, appearance, and mechanics.

6 Limitations and Future Studies

The game developers have limited skills, thus compromising the attractiveness of the game. Furthermore, the game was tested in groups, thus not all students have direct interaction with the game. Some became passive participants, or merely an observer, or worst, an onlooker. These may affect their understanding of stack and opinions of the game. Future recommendations include application of GBL to other data structure topics using various game genres with more challenging and exciting gameplay to make learning more meaningful.

7 Acknowledgement

This paper was sponsored by the Tier 1 grant H117 by UTHM's Research Management Centre.

8 References

- [1] S. Matzko and T. A. Davis, "A graphics-based approach to data structures," *ACM SIGCSE Bulletin*, vol. 40, no. 3, pp. 109-113, 2008. <https://doi.org/10.1145/1597849.1384302>
- [2] A. Sherrod, *Data structures and algorithms for game developers*, Boston: Charles River Media, 2007.
- [3] D. Wassila and B. Tahar, "Using Serious Games to Simplify Algorithm Learning," in *Conference on Education and e-Learning Innovations*, Sousse, 2012. <https://doi.org/10.1109/iceeli.2012.6360569>
- [4] S. S. Shabanah, J. X. Chen, H. Wechsler, D. Carr and E. Wegman, "Designing computer games to teach algorithms," in *2010 Seventh International Conference on Information Technology*, Las Vegas, 2010. <https://doi.org/10.1109/itng.2010.78>
- [5] I. Bezáková, J. E. Heliotis and S. P. Strout, "Board Game Strategies in Introductory Computer Science," in *44th ACM Technical Symposium on Computer Science Education*, Denver, 2013. <https://doi.org/10.1145/2445196.2445210>
- [6] L. Végh, "Using Interactive Game-Based Animations for Teaching and Learning Sorting Algorithms," in *12th International Scientific Conference eLearning and Software for Education*, Bucharest, 2016.
- [7] T. Lauer, "When does Algorithm Visualization Improve Algorithm Learning?—Reviewing and Refining an Evaluation Framework," in *Proceedings of Informatics Education Europe III*, Venice, 2008.

- [8] M. Boilen, M. T. Goodrich, R. Tamassia and B. A. Stibel, "Testers and Visualizers for Teaching Data Structures," *ACM SIGCSE Bulletin*, vol. 31, no. 1, pp. 261-265, 1999. <https://doi.org/10.1145/384266.299779>
- [9] T. L. Naps, "JHAVÉ: Supporting Algorithm Visualization," *IEEE Computer Graphics and Applications*, vol. 25, no. 5, pp. 49-55, 2005. <https://doi.org/10.1109/mcg.2005.110>
- [10] D. W. Shaffer, K. R. Squire, R. Halverson and J. P. Gee, "Video Games and the Future of Learning," *Phi Delta Kappan*, vol. 87, no. 2, pp. 105-111, 2005. <https://doi.org/10.1177/003172170508700205>
- [11] R. N. Van Eck, "Digital Game-Based Learning: Still Restless, After All These Years," *EDUCAUSE Review*, vol. 50, pp. 13-28, 2015.
- [12] Y. Kafai, "The Educational Potential of Electronic Games: From Games-To-Teach to Games-To-Learn," in *Playing by the Rules*, Chicago, 2001.
- [13] Y.-T. C. Yang, "Building Virtual Cities, Inspiring Intelligent Citizens: Digital Games for Developing Students' Problem Solving and Learning Motivation," *Computers & Education*, vol. 59, no. 2, pp. 365-377, 2012. <https://doi.org/10.1016/j.compedu.2012.01.012>
- [14] V. Law and C. Chen, "Promoting Science Learning in Game-based Learning with Question Prompts and Feedback", *Computers & Education*, vol. 103, pp. 134-143, 2016. Available: 10.1016/j.compedu.2016.10.005. <https://doi.org/10.1016/j.compedu.2016.10.005>
- [15] K. Yang, H. Chu and L. Chiang, "Effects of a Progressive Prompting-based Educational Game on Second Graders' Mathematics Learning Performance and Behavioral Patterns", *Journal of Educational Technology & Society*, vol. 21, no. 2, pp. 322-334, 2018.
- [16] P. Wouters, C. van Nimwegen, H. van Oostendorp and E. van der Spek, "A Meta-Analysis of The Cognitive and Motivational Effects of Serious Games.", *Journal of Educational Psychology*, vol. 105, no. 2, pp. 249-265, 2013. <https://doi.org/10.1037/a0031311>
- [17] R. Garris, R. Ahlers and J. E. Driskell, "Games, Motivation, and Learning: A Research and Practice Model," *Simulation & gaming*, vol. 33, no. 4, pp. 441-467, 2002. <https://doi.org/10.1177/1046878102238607>
- [18] C. Reigeluth and F. Stein, "The elaboration theory of instruction", in *Instructional Design Theories and Models: An Overview of Their Current Status*, C. Reigeluth, Ed. Hillsdale: Lawrence Erlbaum, 1983, pp. 335-383. <https://doi.org/10.4324/9780203824283>
- [19] E. B. Costa, A. M. Toda, M. A. A. Mesquita and J. D. Brancher, "DSLEP (Data Structure Learning Platform to Aid in Higher Education IT Courses)," *Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, vol. 8, no. 4, pp. 1143-1148, 2014.
- [20] YoYo Games Ltd., "DS Stacks," YoYo Games Ltd., 2018. https://docs.yoyogames.com/index.html?page=source%2Fdadiospice%2F002_reference%2Fdata%20structures%2Findex.html. [Accessed 30 April 2019].
- [21] A. AM, A. Zulaiha and S. Jelani, "Motivating Low-Achieving Learners to Use Mobile Assistive Courseware through Signaling Principle", *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 13, no. 04, p. 153, 2019. <https://doi.org/10.3991/ijim.v13i04.10527>.
- [22] T. Chuchu and T. Nodoro, "An Examination of the Determinants of the Adoption of Mobile Applications as Learning Tools for Higher Education Students", *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 13, no. 03, p. 53, 2019. <https://doi.org/10.3991/ijim.v13i03.10195>.
- [23] A. Syahidi, A. Supianto and H. Tolle, "Design and Implementation of Bekantan Educational Game (BEG) as a Banjar Language Learning Media", *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 13, no. 03, p. 108, 2019. <https://doi.org/10.3991/ijim.v12i7.9257>.

- [24] A. Hodge, "Learning Data Structures with Fun: The Stack Game," American Association for the Advancement of Science, 2 February 2017. <http://new.emerging-researchers.org/projects/12934/>. [Accessed 29 June 2018].
- [25] S. Kakavouli, "Lab 7 CS230 Data Structures," Wellesley College, 13 March 2007. <http://cs.wellesley.edu/~cs230/spring07/labs/lab7.html>. [Accessed 30 April 2019].
- [26] X. Ge and S. Land, "Scaffolding students' problem-solving processes in an ill-structured task using question prompts and peer interactions", Educational Technology Research and Development, vol. 51, no. 1, pp. 21-38, 2003. <https://doi.org/10.1007/bf02504515>.
- [27] F. Tsai, C. Kinzer, K. Hung, C. Chen and I. Hsu, "The importance and use of targeted content knowledge with scaffolding aid in educational simulation games", Interactive Learning Environments, vol. 21, no. 2, pp. 116-128, 2013. <https://doi.org/10.1080/10494820.2012.705852>.
- [28] S. Barzilai and I. Blau, "Scaffolding game-based learning: Impact on learning achievements, perceived learning, and game experiences", Computers & Education, vol. 70, pp. 65-79, 2014. <https://doi.org/10.1016/j.compedu.2013.08.003>.
- [29] E. Davis and M. Linn, "Scaffolding Students' Knowledge Integration: Prompts for Reflection in KIE", International Journal of Science Education, vol. 22, no. 8, pp. 819-837, 2000. <https://doi.org/10.1080/095006900412293>.
- [30] G. van den Boom, F. Paas, J. van Merriënboer and T. van Gog, "Reflection Prompts and Tutor Feedback in a Web-based Learning Environment: Effects on Students' Self-Regulated Learning Competence", Computers in Human Behavior, vol. 20, no. 4, pp. 551-567, 2004. <https://doi.org/10.1016/j.chb.2003.10.001>.

9 Authors

Rosni Ramle, D'oria Islamiah Rosli, Shelena Soosay Nathan and Mazniha Be-rahim work for Universiti Tun Hussein Onn Malaysia, Johor in Malaysia

Article submitted 2019-03-17. Resubmitted 2019-04-21. Final acceptance 2019-06-04. Final version published as submitted by the authors.