

La Petite Fee Cosmo

Learning data structures through game-based learning

Owen Noel Newton
Fernando

School of Computer Science and
Engineering
Nanyang Technological
University, Singapore,
oferlando@ntu.edu.sg

Vinayak Teoh Kannappan
School of Computer Science and
Engineering
Nanyang Technological
University, Singapore,
vtkannappan@ntu.edu.sg

Xavier Tan
School of Computer Science and
Engineering
Nanyang Technological
University, Singapore, XAVI0007@
e.ntu.edu.sg

Jeffrey Yan Jack Hong
School of Art
Design and Media, Nanyang
Technological University,
Singapore,
JeffreyHong@ntu.edu.sg

Anupam Chattopadhyay
School of Computer Science and
Engineering
Nanyang Technological
University, Singapore,
Anupam@ntu.edu.sg

Hock Soon Seah
School of Computer Science and
Engineering
Nanyang Technological
University, Singapore,
ashsseah@ntu.edu.sg

ABSTRACT

This study aims to investigate the effectiveness of combining an interactive game and the concept of productive failure (PF) to nurture innovative teaching and learning. The study also aims to promote innovative approaches to improve students' learning experience in data structure concepts taught in computer science disciplines, especially in the linked list concepts. A 2D bridge-building puzzle game, "La Petite Fee Cosmo" was developed to assist students in understanding the underlying concepts of the linked list and foster creative usage of the various functionalities of a linked list in diverse situations. To evaluate the potential impact of the interactive game and implications of productive failure on student learning; a pre-test, post-test and delayed-test were developed and used in the evaluation process. Further, the technology acceptance model (TAM) was applied to examine the factors that influence the adoption of productive failure approach in learning data structures.

CCS CONCEPTS

Human-centered computing → Human computer interaction (HCI) → HCI design and evaluation methods → User studies

KEYWORDS

Gamification, game-based learning, data structure, linked list, productive failure, interactive learning

ACM Reference format:

Owen Noel Newton Fernando, Vinayak Teoh Kannappan, Xavier Tan, Jeffrey Yan Jack Hong, Anupam Chattopadhyay and Hock Soon Seah. 2019. La Petite Fee Cosmo: Learning data structures through game-based learning. In *Proceedings of VRCAI '19: The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI '19), November 14-16, 2019, Brisbane, QLD, Australia*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3359997.3365688>

1 Introduction

In computer science (CS) disciplines, students often find abstract concepts taught in computer science (CS) courses to be challenging to understand. One such concept is data structures. Data structures are one of the most fundamental concepts in CS, and it is one of the bases of writing useful and efficient programs. A data structure is a collection of data values with relationships among them. Functions or operations can be applied to the data structure for management or modification. Programmers often use them to efficiently access and modify data or values [Wirth 1978]. There are many types of data structures, but students find the concept of the linked list structure challenging to understand.

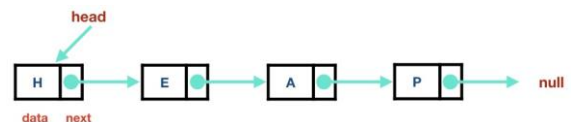


Figure 1: Singly linked list

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

VRCAI '19, November 14-16, 2019, Brisbane, QLD, Australia

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-7002-8/19/11...\$15.00

<https://doi.org/10.1145/3359997.3365688>

A linked list [Cormen et al. 2009] is a linear data structure of linked nodes. A visual representation of a linked list can be seen in Figure 1. A node has two items: a pointer referencing the next node and the data stored. Each node is considered a separate object in memory. A linked list must have a head pointer, which references the first node in the list and signifies the beginning point of the linked list. The last node always points to null or nothing, which signifies the end of the linked list. A linked list with a head pointer pointing to null is considered an empty list. The linked list is complemented with functions such as add, remove, and find nodes. In traditional lectures, some students find it hard to comprehend how these functions work internally and how to use them meaningfully, thus resulting in them to lose focus and be demotivated easily.

Data structures itself is a difficult subject, and the linked list is one of the first data structures that students learn. There were attempts to help students cope with learning data structures through visualization, but a study has proven that mere visualization is insufficient in learning as students may look at the visualization without understanding the context [Hundhausen et al. 2002]. We propose using game-based learning [Qian and Clark 2016; Sanchez 2019; Mayer 2019] or gamification [Mora et al. 2017] combined with the concepts of productive failure (PF) [Kapur 2006; Kapur 2012] as a learning technique to support active learning where students can explore, experiment, fail and learn from failure [Izu and Castro 2018]. Our approach aims to improve students' focus and motivation for learning, improve, and retain the learned knowledge through detailed instructions.

An understanding of the definition of games is required before utilizing the concepts of game-based learning and gamification. According to Huizinga [Huizinga 1955], games create a magic circle around the player where rules of "ordinary life" do not apply, instead only the rules of the game itself apply. Tekinbaş & Zimmerman [Tekinbaş and Zimmerman 2004] expands on this definition by adding games as "a system in which players engage in an artificial conflict, defined by rules, those results in a quantifiable outcome." Schell [Schell 2008] proposes an exciting addition to the definition of games by describing it as using a playful attitude to approach or solve a problem. Kapp [Kapp 2012] has expanded the definition by defining games as "a system in which players engage in an abstract challenge, defined by rules, interactivity and feedback that results in a quantifiable outcome often eliciting an emotional reaction." A game should also provide feedback in a typically prompt, direct and precise manner [Kapp 2012].

Gamification refers to applying game design elements in non-gaming contexts. According to Deterding [Deterding et al. 2011], gamification does not result in a full-fledged game, but rather a system that incorporates certain game elements. Gamification itself is also not limited by the technology used. Werbach & Hunter [Werbach and Hunter 2012] suggests that gamification is "the use of game elements and game design techniques in non-game contexts." Werbach & Hunter emphasizes the separation of game-design techniques from game elements, suggesting that good gamification requires more than just applying certain game

elements. Kapp [Kapp 2012] identifies gamification as "using game-based mechanics, aesthetics and game thinking to engage people, motivate action, and promote learning and solving problems." From Kapp's perspective, more concern should be placed on building a game-like experience rather than just the implementation of game elements.

Productive Failure (PF) is a learning technique designed to encourage students in finding solutions through failure for a set of complex or novel problems related to the subject through exploration in an unguided manner before an actual instructional course or lecture [Kapur 2006]. PF attempts to solve problems with early direct instruction (DI). In DI, ideas and concepts are taught to the students. In general, students do not have prior knowledge of the subject; they may not understand why the subject is represented and assembled in the way they are [Kapur 2006; Kapur 2012]. In games, allowing players to fail and learn from failures is an integral part of its design [Juul 2013], and it plays a part in the learning process in PF [Kapur 2006]. Therefore, it is reasonable to compare failure in games to failure in PF.

To help students understand the concept of Linked List in the Data Structures' course, we developed the data structure game called "La Petite Fee Cosmo". This game aims to provide the students to visualize and understand the important linked list operations, including finding a node from any location in the linked list, inserting a node at a specific location in the linked list, deleting a node from an arbitrary location in the linked list, and merging two linked list, etc.

A pre-test, post-test, and delayed-test were developed and used to evaluate the potential impact of the game on student learning. A group of students (22 students) is divided into 3 experiment groups: two different productive failure (PF) groups and one direct instruction (DI) group. A survey was conducted after each session to collect feedback from the students. We also use technological acceptance model (TAM) [Davis 1989] to study how well the game has fared amongst the participants. Initial assessment outcomes show promising results and the feedback from students is very positive according to the survey. We believe that this game will help instructors to engage students in understanding and to learn the concept of linked list presented in class by making the learning process a fun activity.

The rest of this paper is organized as follows: Section two (2) provides information about related work on data structures games. Section three (3) describes the design and implementation of the game in detail. Section four (4) discusses detailed information about experiment design and presents the assessment results. Section five (5) presents our conclusions and future work.

2 Related Work

There were many attempts to integrate gamification or game-based learning as part of teaching. One study on using competitive games to teach data structure by R. Lawrence [Lawrence 2004] has found that interesting assignments encourage student learning by actively engaging them. He also

emphasizes that a motivated student is more likely to be a more successful learner.

Another such example is a game called Space Traveler, developed by Zhang et al. [Zhang et al. 2015] to target the singly linked list data structure. The game features a spaceship and players are to prevent it from being destroyed by enemies. A spacecraft and orbs respectively represent the linked list and nodes. If an orb collides with an enemy, the spacecraft will be damaged. Players can enhance their ship through the purchase of weapons and adding them to the orbs. Orbs that are damaged need to be removed from the ship through filling in missing parts of the code. Feedback is also given if players fill in the wrong code.

Toda et al. [Toda et al. 2013] showed an interactive learning environment application for data structures. The application consists of all the underlying data structures such as stack, queue, list, trees, and array, and each contain a set of activities. Toda et al. employ gamification in which game elements like achievements, scores and badges are available. The game attempts to show some real-life analogies to the data structures.

Dicheva et al. developed a 3D puzzle game focusing on the stack data structure called “The Stack Game” [Dicheva et al. 2016; Dicheva and Hodge 2018]. The game was designed to help students understand the abstract concepts of stacks and the ability to write the main methods of stacks, using them to solve problems. Players control a robot whose spacecraft has crash-landed and must lead it back home while overcoming obstacles by solving stack related challenges. The game is separated into parts where each part represents a learning objective. Feedback is given to the students if the wrong code was written, and players can try again.

3 Design and Implementation

3.1 System Design

During the conceptualization, the type of game needs to be identified, and it must accommodate the concepts of a linked list. The mechanics and aesthetics need to be designed around it also. A puzzle game was decided upon which resulted in fantasy-themed 2D side-scrolling puzzle game titled “La Petite Fee Cosmo” (Cosmo Game). Cosmo game aims to not only help students reinforce their understanding of linked list but also to provide visualization in an interactive and interesting manner. The game contains challenges, which prompt students to use the basic functions of the linked list in different scenarios.

The game was designed and developed using the Unity game engine. Figure 2 shows the system overview of Cosmo Game. Players use the keyboard and mouse primarily to navigate with the game application. In Unity, all game objects are stored in a scene file. The manager objects are necessary to create functionality in the scene. In Cosmo Game, the primary manager objects are the game rules controller, level manager and game manager. The managers are built using a modified singleton pattern that complements Unity’s design architecture. The game rules controller contains a set of rules for each scene and players are unable to operate outside the bounds of the rules. The level

manager manages the game environment such as defining the playable bounds of the game world, handling developer-scripted events for the level, and functionality to manipulate the game world. The game manager is used to control the event sequences in the scene.

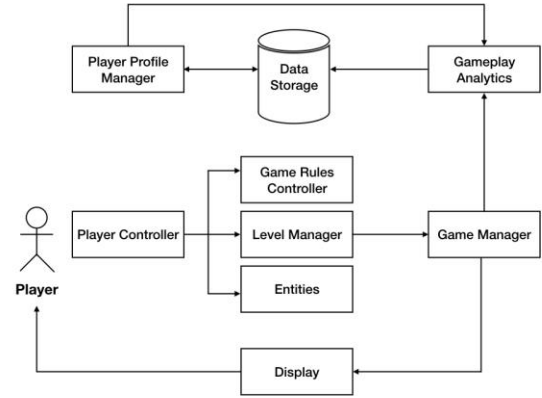


Figure 2: System overview

The goal of the game is to assist the character (named Cosmo) using the available actions to build a bridge, which will enable her to reach the opposite end of the level. Upon successful completion of building the bridge, crystals, as well as a portal, will show up. Cosmo needs to collect the crystals and enter the portal to complete the level. A bridge represents the linked list is made up of floating rock platforms, which represent the nodes. The node’s data is represented by a glyph and a numeric value wrapped in a speech bubble. The glyph is used to represent unique memory addresses of the nodes, and the numeric value represents the data stored.

3.2 Game Design and Rules

Figure 3 shows the Action Panel (section 1), which players use to control entities in the game. Each button executes an action that controls Cosmo or the game world. Players can view the description for the action by hovering the mouse cursor over the button. Some actions are shared across different levels, while some actions are unique to each level. The common actions include moving Cosmo, storing or removing a value to the inventory and setting or removing a value from inventory to a platform. Figure 3 (section 2) displays the elapsed game time since the start of the game. Figure 3 (section 3) shows the value collection, the inventory and the head. The value collection shows a list of values that players can use for the stage — the inventory stores a single value from the value collection, or a value retrieved from a platform. The head shows the glyph of the first platform in the bridge. Players can pause, resume, restart or quit the game at any time.

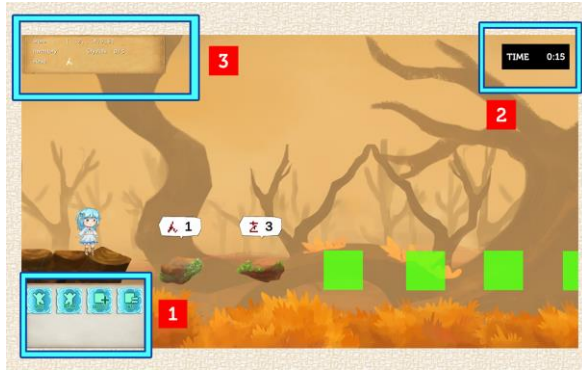


Figure 3: The game screen and highlighted game elements

The game consists of 3 levels, broken into a total of 5 stages. Each level contains a challenge. Levels 1 and 2 are further broken into two stages; the reason being that level 1 and 2 share a common challenge theme. Level 3 has only one stage. Following one of the core principles of PF, the stages are designed to be complicated enough but not impossible to solve [Kapur 2012]. The images in Figures 4, 5, 6 show the layout of each stage.



Figure 4: Level 1-1 (left) and level 1-2 (right).

In level 1 (as shown in Figure 4), players need to bring either the smallest or the largest value to the front of the given list. A staircase like a layout is used to imply the ordering of the values. This level introduces students to the concepts of adding and rearranging nodes by modifying the links between nodes. Students are also introduced to the concept of the head pointer and how it works.



Figure 5: Level 2-1 (left) and level 2-2 (right)

In level 2 (as shown in Figure 5), node values must be sorted either in ascending or descending order. Level 2 is an extension of the first level as it tests students on the abilities learned in the previous level.

In the final level, players must place the values from the lower bridge into alternate positions of the upper bridge. Cosmo is placed at the start point of the upper bridge, which indicates the main bridge that needs to be completed. This level will not

only test abilities learned in the previous level but also new concepts such as finding, adding, and removing nodes in different positions, alternating between multiple linked lists and advanced concepts of linking nodes.

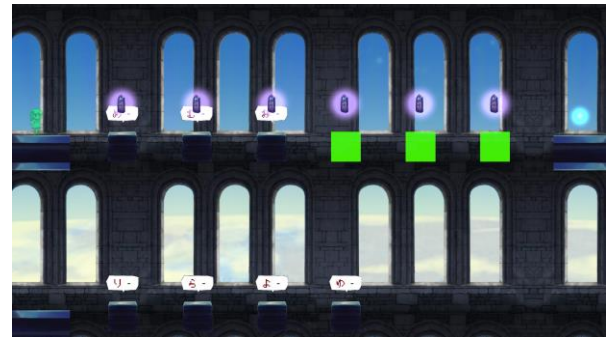


Figure 6: Level 3 – Final level

3.3 Design Choices and Game Aesthetics

The design choices in the game were inspired by ideas from two prominent game designers, Miyamoto Shigeru from Nintendo and Yasuhara Hirokazu from Sega. In Figures 3, 4, 5 and 6, the green squares represent slots where platforms will be placed. Miyamoto explains his design process on how to teach a player to play the game through its mechanics, level design and aesthetics rather than spoon-feeding players with instructions [deWinter and Kocurek 2015]. The green slots prompt players that the bridge requires more platforms. Using Miyamoto's design methodology [deWinter and Kocurek 2015], each stage in the game is designed with unique layouts, and the layouts give players an idea of the goal instead of having to refer to a manual or a tutorial page.



Figure 7: From left to right: Slot, crystal and, exit portal

The game provides immediate feedback to the players for every action that they make. Hence, allowing them to learn from their mistakes. When players move Cosmo, sounds and animations are played to indicate movement. When a stage's challenges are complete, purple crystals and the exit portal appears, as shown in Figure 7, which indicates positive progress. Yasuhara shares his methodology on using moving objects to create focus [Yasuhara 2017]. Whenever players modify the bridge, the camera moves towards the location where the action is happening, forcing students to focus their attention on the action and observe the behavior of the linked list through animations. Yasuhara mentions that humans are attracted to unique things [Yasuhara 2017]. Both the crystals and the portals glow to amplify their visibility, thus prompting players to collect

the crystals and enter the portal to complete the level. Upon entering the portal, the game displays the results screen, which shows the performance of the player. Though the game primarily uses gameplay mechanics and level design to teach the player, an instruction manual was made to aid players who have difficulty navigating the game. The manual does not provide any hints on how to solve the level; it only provides pointers on how to operate the game and navigate the screens.

```

1  if(pre = findNode(*ptrHead, index -1) != NULL){
2      if(pre->next == NULL)
3          return -1;
4      cur = pre->next->next;
5      free(pre->next);
6      return 0;
7  }

```

Figure 8: Code snippet of a removing a node in linked list

The rules of the game are designed in such a way that it would force the player to act programmatically. Hence, it would be easier for the player to relate back to the code after being introduced to the game. For instance, in level three, students are required to remove a certain node. To do so, first, they would have to move the character to the particular node. This corresponds to the first line of the code in Figure 8. The player would then have to delete the node. When deleting the node, the animation would show that the previous node connects to the next node before the link to the current node gets removed. This corresponds to the order of the code where line four was done before line five. Lines 2 and 3 are not reflected in game, as it is a validation check.

The game comes with narration and lore to give purpose and a form of progression mechanic for the game. Character background is also written to give players a purpose to care and sympathize with the character. The lore provides players with an extra layer of immersion and as a form of motivation. The lore and character background were printed in the manual for players as reference.

The game's aesthetics reflects the lore to enrich the environment. Since the game relies on a fantasy-themed storyline with fae elements, Gaelic, Germanic, and French elements were chosen. In the mountainous region stages (Figure 4), the hills were inspired by the Isle of Skye in Scotland. The floating castle was inspired by Le Mont-Saint-Michel in Normandy, France. The forest stages (Figure 5) were based upon the Schwarzwald Forest during autumn in Germany. Finally, the castle stage was primarily inspired by the feel of the interior of Le Mont-Saint-Michel's abbey. The art style used was inspired primarily by the Nausicaa watercolor impressions by Miyazaki Hayao [Miyazaki 1996]. The goal was to attempt to replicate the feel from children's fairy tale book to the game. The aesthetics were chosen in hope to make the game more comforting to play, which in turn should at least lower some level of frustration from the players as they try to solve the levels.

4 Experimental Design and Evaluation

The experiment was done at Nanyang Technological University, with 22 participants. Most of the participants are undergraduate students of the university. The entire experiment lasted for around 3 hours for each group, excluding the one-week break. The long experiment hours were believed to be one of the main reasons for the low turnout rate.

To evaluate the effectiveness of the PF game in improving the student's learning outcome, the participants are split into three different groups. Group 1 plays the game first; thereafter-detailed instructions in the form of lectures have been given to them. The lectures are similar to one found from the module. After that, they take a paper-based post-test. However, Group 2 took the paper-based quiz as a pre-test before getting any detailed instructions on the linked list. Thereafter, take another paper-based post-test. Group 3, referred to as direct instruction (DI) group, gets the detailed instructions first before taking the post-test. They follow the same conventional teaching method. All instructions given are the same. All questions for pre-test, post-test and delayed-test given are the same. A 30-minute rest is given in-between each activity. All three groups then take a delayed-test one week later. A survey was conducted as well to gain feedback about the game. Figure 9 depicts an overview of the experiment flow.

Comparison between groups 1 and 2 is to evaluate the effectiveness of the game in creating an environment for students to learn better with PF. The DI group is formed as a control group. The flow for the DI group represents the traditional method of teaching. Firstly, the students receive instructions through lectures or seminars; thereafter, they go for tests or exams. Hence, the DI group should be a baseline representation of how well the current students are faring in data structures, especially in the linked list.

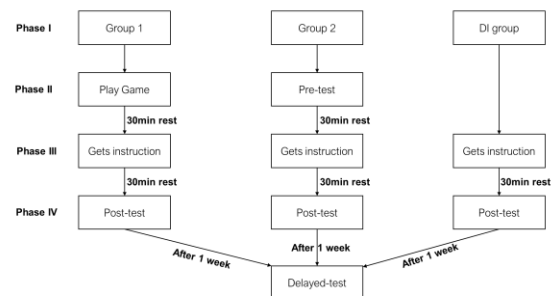


Figure 9: Overview of the experiment flow

The game and the pre-test are avenues for the participants to experience failure in understanding and using linked list concepts. The instructions given to groups 1 and 2 are intentionally delayed as the concept of PF suggests that students learn better if they get the instruction after experiencing failure [Kapur and Bielaczyc 2012].

The post-tests, delayed-test, and group 2's pre-test are conducted on the same system used by the students for their data structure module activities. It is an automated programming assessment system (APAS), meaning that the way the participants are evaluated will be fair and consistent. In APAS, they would have to code a specific function using basic functions provided, such as add, remove or find a node. The system will then evaluate the correctness of the function coded by running a series of test cases. The tests are essentially a comprehensive list of inputs. The C language is used for the coding. In the pre-test, there are three questions. Examples of the questions for pre-test, post-test and delayed-test are as in Figures 10, 11, and 12.

```
Content :
(moveToFrontLL) Write a C function moveToFrontLL() that moves the last element of a given
linked list to the front.

The function prototype is given below:
void moveToFrontLL(LinkedList *ll);

For example:
If the given linked list is (1, 5, 2, 8, 7), then the resulting linked list will be (7, 1,
5, 2, 8).
If the given linked list is (1, 2, 3, 4, 5), then the resulting linked list will be (5, 1,
2, 3, 4).
```

Figure 10: Example questions from pre-test

```
Content :
(pairWiseSwapLL) Write a C function pairWiseSwapLL() that swaps elements of a given
linked list. Start from the head node and traverse the list. While traversing through the
linked list, swap data of each node with its next node's data.

The function prototype is given below:
void pairWiseSwapLL(LinkedList *ll);

For example:
If the linked list is (1, 2, 3, 4, 5, 6, 7), the resulting linked list will be (2, 1, 4,
3, 6, 5, 7).
If the linked list is (1, 2, 3, 4, 5, 6) then the resulting linked list will be (2, 1, 4,
3, 6, 5).
```

Figure 11: Example questions from post-test

```
Content :
(moveMaxToFront) Write a C function moveMaxToFront() that traverses a linked list of
integers at most once, then moves the node with the largest stored value to the front of
the list.

The function prototype is given as follows:
int moveMaxToFront(ListNode **ptrHead);

For example, if the linked list is (30, 20, 40, 70, 50), the resulting linked list will be
(70, 30, 20, 40, 50).
```

Figure 12: Example questions from delayed-test

For group 1, the game that they played would represent the pre-test equivalent for group 2. Both the game and quiz have the exact same challenges. The game was built to analyse and infer player's knowledge, skill level and improvements. A player profile manager and gameplay analytics framework were developed to support saving data on the PC for multiple users. The analytics system logs all player performance statistics for each stage. A new log file is created for every replay of the same level – these interfaces directly with the operating system's file system for data storage. Singleton pattern was also used to develop these systems. Each time a new player logs into the game, a file with a unique id is generated. The gameplay analytics system relies on the player's unique id to match its analytics file to a player. The post-test was done to measure their improvement after the lecture. The delayed test, on the other hand, was conducted to evaluate how well students can retain the knowledge over a certain period for the respective group and to verify the results.

5 Experiment Results

5.1 Game Statistics

The game is able to capture data on player actions. The data we collected has included, the total playtime, number of restarts, number of actions used, and number of crystals collected. All participants were able to complete the game for all three levels. It was only a matter of how long they took to complete the game. For levels 1 and 2, none of them restarted the game. However, for level 3, 80% of the participants restarted once. This indicates that players did experience some form of failure in the game and was able to learn from it. Figure 13 shows the average actions performed playtime and restart count for each level. Do note that average restart count displayed in the graph seems very small because the number of restarts is very small compared to the other statistics, which are in the 50s and 100s range.

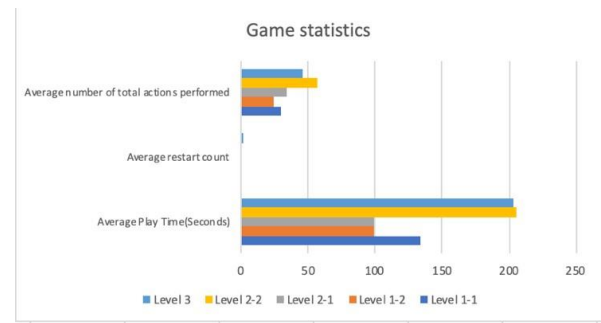


Figure 13: Statistics from the game

5.2 Pre-Test

The results for group 2 were quite fair, with only 57% of the participants passing the test (achieving above 50% out of 100%). The statistics for the pre-test results are in Table 1.

Table 1: Group 2's pre-test statistics.

Group 2		
Mean		48.2%
Standard Deviation (SD)		40.87
Passing rate (≥ 50)		57.14

5.3 Post-Test

Group 1's post-test result was comparatively better than group 2's. Group 1 has a passing rate of 60%, while group 2 only has 42.9%. Group 3's result was dismal, with only 28.6% passing. The statistics for the post-test group is as in Table 2. From the statistics, it has shown that the game does provide a suitable environment for the participants to gain context first and

therefore learn better from the instructional lectures. They are showing better results compared to DI group.

5.4 Delayed-Test

As the delayed-test was done a week later, there could be many factors involved in this part of the test. The students could have done more self-studying or tried more tests. The statistics for the delayed-test are as in Table 3. The passing rate for group 1 is the same, but the mean has increased. While for group 2 as well as DI group, there was a significant increase in their average score as well as passing rate. However, group 2's average score for post-test results has decreased. The decreased could have caused by several factors, such as the difficulty level of the questions differ much more than we initially thought or the condition of the participants. Additionally, because of the high weight of each question in the quiz coupled with low sampling rate, a small change in marks would result in a higher than proportionate change in the percentages.

Table 2: Statistics for post-test results.

	Group 1	Group 2	DI Group
Mean	60.80%	41.57%	40.00%
Standard Deviation (SD)	35.84	34.02	42.86
Passing rate (≥ 50)	60.00%	42.90%	28.60%

Table 3: Statistics for delayed-test results.

	Group 1	Group 2	DI Group
Mean	65.00%	75.00%	54.20%
Standard Deviation (SD)	37.91	41.83	43.06
Passing rate (≥ 50)	60.00%	71.40%	42.90%

5.5 PF (Productive Failure) Game Discussion

From the results of the post-test and delayed-test, it is clear that Group 1 and Group 2 performed better than the DI group. In post-test, Group 1 and Group 2 have higher mean scores of 60.80 and 41.57 respectively compared to DI group. DI group only score 40.00%. Group 1 and Group 2 have passing rates of 60% and 42.90% respectively, while DI group is significantly lower at 28.60%. While it can be difficult to draw statistical significance from the results due to the low number of participants and large standard deviation, we are sanguine about the effects of the game and PF. The nature of the question has also contributed to the high levels of deviation; each test has at most three questions, and hence, each question holds very high weight. A failure in one question would, therefore, significantly affect the overall score.

Although the quantitative statistics do show satisfactory results to substantiate the idea, the small number of sample size may not have given us the full picture. At the very least, it does point us in the right direction and a stepping-stone for future experiments. In similar studies, they have utilized their actual classes for the experiments, such as the one from R. Lawrence's "Teaching data structures using the competitive game", he used his data structure class, consisting of 55 students to carry out his study and experiment [Lawrence 2004]. In another study was done by Dicheva et al. [Dicheva et al. 2019] used actual classes to carry out their experiment. Their total number of participants is 27. Hence, for our experiment, we should have around 30 to 40 participants, ideally.

To understand the participants view towards the game, anonymous open-ended comments were allowed to be written by them. One of the participants said, "*The animations and the design of the game made it enjoyable to play.*" Another commented, "*The game does not look very technical and hence, it is more enticing to play as it does not remind the difficulty of coding.*" One participant also commented that the game was intuitive to play and navigate through. To further understand and substantiate the effects of the PF game, we conducted a technology acceptance model (TAM) based survey [Davis 1989] to gather feedback from the participants.

6 Survey

The survey questions were made to gather feedback about the technological acceptance level. Likert scale was used for the response in the survey [Elaine and Seaman 2007]. For the purpose of data analysis, each response was given a score (Strongly disagree = 1, disagree = 2, neutral = 3, agree = 4, strongly agree = 5). For negative typed questions, the score is inversed, meaning strongly disagree = 5, disagree = 4, so on and so forth. The survey was conducted right after the post-test to ensure that they have the freshest memory possible to answer more accurately. From the survey, 80% of the participants either agreed or strongly agreed on the statement that "I felt that I was more engaged in class after playing the game". 100% of them either agreed or strongly agreed to the statement "I felt that the game provided better learning experience than the traditional course". 60% of the participants either disagree or strongly disagreed that "The mistakes (failure) decrease my interest in learning". The survey does indeed show that the participants feel that the PF game helped them engage and learn better about the subject.

6.1 Technological Acceptance Model

The model provides insights to factors that will affect the user's acceptance and behaviour to the technology (referring to the game), and it was introduced by Davis [Davis 1989]. The study model takes into consideration two main factors: perceived ease of use (PEOU) and perceived usefulness (PU) of the technology. Perceived ease of use is the degree of effort an individual believes is required when using a system. Perceived usefulness of the technology is the degree of enhancement a system will bring

to the individual's performance. These two factors form the beliefs of a user when they first use the system, which in turn influence their "Attitude Towards Using" (AT) and "Behavioral Intentions" (BI). AT is the measurement of the desirability of the user to use the system while BI measures the strength of one's intention to perform a specified behaviour. Figure 14 shows the model that TAM has adopted.

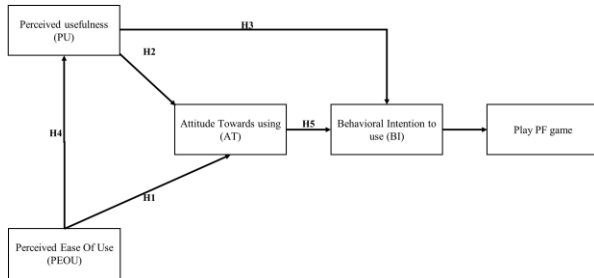


Figure 14: TAM study model

Based on the model, we can formulate up to five hypotheses. The hypotheses are as follows:

H1: PEOU will positively influence players' AT playing the game.

H2: PU will positively influence players' AT playing the game.

H3: PU will positively influence players' BI towards playing the game.

H4: PEOU will positively influence PU of the game

H5: AT the game will positively influence players' BI to play the game

6.2 Result Analysis

To test for the reliability and internal consistency of the survey for TAM analysis using Cronbach's coefficient alpha was used. A coefficient value of 0.7 to 0.90 was said to be acceptable, and anything above 0.90 could mean that there were redundant questions [Tavakol and Dennick 2011]. The results are shown in Table 4.

Table 4: Cronbach's coefficient alpha.

Factor	Number of Questions	Cronbach's Alpha
Perceived Ease of Use (PEOU)	7	0.766
Perceived Usefulness (PU)	6	0.955
Attitude Towards using (AT)	5	0.899
Behavioural Intention to use (BI)	2	0.833

The results are indeed satisfactory except for PU where the coefficient value is above 0.90; hence, there could be too many questions in the survey of the same purpose. The Pearson

correlation is used to investigate the relationship between the four factors in the TAM. The coefficient ranges for -1 to 1, where a more positive value corresponds to a more positive association, vice versa. The results are collated, as in Table 5.

Table 5: Cronbach's coefficient alpha.

	PEOU	PU	AT	BI
PEOU	1	0.988	0.635	0.988
PU	0.988	1	0.652	0.991
AT	0.635	0.652	1	0.583
BI	0.988	0.991	0.583	1

The linear regression analysis was done to test the hypotheses. The results are depicted in Table 6. The perceived ease of use has the highest positive impact on the players' perceived usefulness of the game, with a standardized coefficient of 1.5. Additionally, it is significant with a relatively low p-value of 0.0015 (less than 0.05). The only other significant hypothesis is H3. Where the perceived usefulness will positively influence the players' behavioural intention to play the game again. It also has a relatively high level of impact with a standardized coefficient of 0.969 and a p-value of 0.001.

Table 6: Linear regression analysis.

Hypothesis	Standardized Coefficient	P-value
H1: PEOU will positively influence players' AT playing the game.	0.564	0.250
H2: PU will positively influence players' AT playing the game.	0.382	0.233
H3: PU will positively influence players' BI towards playing the game.	0.969	0.001
H4: PEOU will positively influence PU of the game	1.50	0.0015
H5: AT the game will positively influence players' BI to play the game.	0.972	0.302

6.3 Survey Discussion

Our experiment did not manage to find any significant correlation between PEOU and AT, PU and AT, AT and BI. This is likely, again due to the small sample size of our survey participants. To further understand the participant's sentiments about the experiment, similarly like before, we gathered some verbose feedback. One participant explained that after playing the game, it is easier to understand the instructions the lecturer explained. Another commented that even though he would not

play the game on his own, he feels that the game should be incorporated into the course as self-learning material.

7 Conclusion and Future Work

In conclusion, the experiment has proved that the PF game has indeed played a part in improving students' learning experience. Although the experiment did not have many participants, it was able to point to the direction that the game was able to improve the learning outcome for CS concepts like a linked list. Of course, there will always be a bias where the participants who played the game were naturally more academically inclined into the subject. However, the groupings were already randomized to reduce such bias. In future experiments, we would have grouped them according to their grade point average (GPA). Furthermore, the game will be expanded to include more data structures like queues, stacks, and binary trees. An expanded gameplay idea will also be developed to accommodate other data structures. Additionally, we would try to simplify the entire experiment process in the hope of attracting more participants for possibly a future experiment to further substantiate the results.

The design of the game was approached in a different way compared to the others mentioned in the related work section. The game was developed based on practical questions on the linked list, which in turn was one of the challenges faced. However, this ensures that different application scenarios of the linked list are covered in the game. Failure has also become an integral part of the game with the application of PF. The failure aspect of the game is designed not to frustrate and demotivate; instead, it gives students a chance to review and try new solutions and help their problem-solving skills. We hope that the game will become instrumental in helping students become better achievers by providing them with an opportunity to learn more interestingly and engagingly.

ACKNOWLEDGMENTS

The research was supported by the Singapore Ministry of Education (MoE) Tertiary Education Research Fund (TRF) and the Centre for Augmented and Virtual Reality.

REFERENCES

- Niklaus Wirth, 1978. *Algorithms & Data Structures*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Thomas H. Cormen, Charles. E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms* (third edition). The MIT Press.
- Christopher D. Hundhausen, Sarah A. Douglas, and John T. Stasko. 2002. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing*, 13(3), 259–290.
- Meihua Qian and Karen Rebecca Clark. 2016. Game-based Learning and 21st century skills: A review of recent research. *Computers in Human Behavior*, 63, 50–58.
- Eric Sanchez. 2019. Game-Based Learning. In: Tatnall A. (eds) *Encyclopedia of Education and Information Technologies*. Springer.
- Richard E. Mayer. 2019. Computer games in education. *Annual review of psychology*, 70, 531–549.
- Alberto Mora, Daniel Riera, Carina S. González, and Joan A. Moreno. 2017. Gamification: a systematic review of design frameworks. *Journal of Computer High Education*, 29:3, pp 516–548.
- Manu Kapur. 2006. Productive failure. In *Proceedings of the 7th International Conference on Learning Sciences (ICLS '06)*. Int. Society of the Learning Sciences 307–313.
- Manu Kapur. 2012. Productive failure in learning the concept of variance. *Instr Sci*, 40:4, pp 651–672.
- Cruz Izu and Olga S. Castro. 2018. Designing Active Mediated Learning Tasks: Can Small Failures Enhance Student Learning? In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA.
- Johan Huizinga. 1955. *Homo Ludens: A Study of the Play-Element in Culture*. Boston, MA, USA: Beacon Press.
- Katie S. Tekinbaş and Eric Zimmerman. 2004. *Rules of Play: Game Design Fundamentals*. Cambridge, MA, USA: The MIT Press.
- Karl M. Kapp. 2012. *The gamification of learning and instruction: game-based methods and strategies for training and education*. San Francisco, CA, USA: John Wiley & Sons.
- Jesse Schell. 2008. *Game Mechanics Support Puzzles, The Art of Game Design*, pp. 207–219.
- Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, New York, NY, USA, pp. 9–15.
- Kevin Werbach and Dan Hunter. 2012. *For the win: how game thinking can revolutionize your business*. Philadelphia, PA, USA: Wharton Digital Press.
- Jesper Juul. 2013. *The art of failure: An essay on the pain of playing video games*. Cambridge, MA, USA: The MIT Press.
- Ramon Lawrence. 2004. Teaching Data Structures Using Competitive Games. *IEEE Transactions on Education*, 47 (4), 459 - 466. 10.1109/TE.2004.825053.
- Jinghua Zhang, Mustafa Atay, Elvira R. Caldwell, and Elva J. Jones. 2015. Reinforcing student understanding of linked list operations in a game. *IEEE Frontiers in Education Conference (FIE)*, El Paso, TX, pp. 1–7.
- Armando M. Toda, Marcell A. A. Mesquita, Carlos R. S. Moraes, Artur F. N. Freires, and Jacques D. Brancher. 2013. Interactive learning environment for data structures with gamification concepts. In *WWW/Internet. IADIS*.
- Darina Dicheva, Austin Hodge, Christo Dichev, and Keith Irwin. 2016. On the design of an educational game for a data structures course. In *IEEE Int. Conf. on TALE, Bangkok, Thailand*, 7–9 Dec.
- Darina Dicheva and Austin Hodge. 2018. Active Learning through Game Play in a Data Structures Course. In *SIGCSE'18*, Baltimore, MD, USA, 21–24 Feb.
- Jennifer deWinter and Carly A. Kocurek. 2015. Shigeru Miyamoto: Super Mario Bros., Donkey Kong, The Legend of Zelda. *Bloomsbury Academic*, New York, USA.
- Hirokazu Yasuhara. 2017. How to Make a Game Fun. Accessed: Apr. 28, 2019. [Online Video]. Available: <https://www.youtube.com/watch?v=J-3avMBqJ9s>
- Hayao Miyazaki. 1996. *The Art of Nausicaa of the Valley of the Wind: Watercolor Impressions*. Tokyo: Tokuma Shoten (in Japanese).
- Manu Kapur and Katherine Bielaczyc. 2012. Designing for Productive Failure. *Journal of the Learning Sciences*, vol. 21, no. 1, pp. 45–83.
- Darina Dicheva, Keith Irwin, and Christo Dichev. 2019. Engaging Students in a Gamified Data Structures Course. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pp. 386–392.
- Fred Davis. 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3), 319–340.
- Allen I. Elaine, and Christopher A. Seaman. 2007. Likert scales and data analyses. *Quality progress*, 40(7), 64–65. ZY.
- Mohsen Tavakol and Reg Dennick. 2011. Making sense of Cronbach's alpha. *International journal of medical education*, 2, 53–55.