Game-Based versus Gamified Learning Platform in Helping University Students Learn Programming

Oka Kurniawan
Singapore University of Technology and Design
Singapore

Norman Tiong Seng Lee
Singapore University of Technology and Design
Singapore

Nachamma Sockalingam
Singapore University of Technology and Design
Singapore

Kin Leong Pey
Singapore University of Technology and Design
Singapore

Novice learners of programming may benefit from some prior knowledge in programming before taking their first introductory programming course. In this study, we describe a workshop aiming to do so, that is offered to undergraduate students before their first programming course. Two online platforms are used in this study, a game-based platform (CodeCombat) and a gamified online tutorial (CodeAvengers). We compare the effects on learning of the two platforms on their academic performance, and investigate students' preferences and subsequent usage of these platforms. Results show that the workshop participants prefer the gamified platform over the game-based platform for learning, and use during their programming course for revision and more practice. We found no significant difference in learning outcomes amongst those who participated in the workshop and those who did not. We discuss the findings and implications of this study in the paper.

Keywords: game-based learning, gamification, programming, online platform

Introduction

The ability to program is becoming more important in many engineering disciplines (E. A. Lee & Messerschmitt, 1998) and also in architecture (Leitão, Cabecinhas, & Martins, 2010). Thus, it would be advantageous to any undergraduate in a technological discipline to have sufficient programming skills. Hence, all four degree programs offered by the Singapore University of Technology and Design (SUTD) have several courses that require the use of programming skills. All students in SUTD take common courses in their first three terms, one of which is an introductory programming course in Python programming called "The Digital World". This course has to cater to students with a wide variety of academic abilities, varying levels of prior programming experience and motivation. The majority of the students have no prior experience in programming and thus are novice learners.

The difficulties that novice learners of programming face when learning programming have been well-documented (Robins, Rountree, & Rountree, 2003). Novice learners tend to make syntax errors, possess alternative conceptions in programming concepts and find it difficult to plan, write and debug programs (Qian & Lehman, 2017). Other studies have reported that novice programmers lacked skills in reading and tracing code (Lister et al., 2004) and problem-solving ability (McCartney, Boustedt, Eckerdal, Sanders, & Zander, 2013; McCracken et al., 2001). This is due to the observation that programming tasks tend to be cognitively demanding, with several cognitive demands on learners, starting with the learning of language features, and ending with developing problem-solving skills (Linn & Dalbey, 1985).

Hence, it seems intuitive that students with some form of prior programming experience should perform better on assessments programming course compared to those without. Currently, studies are mixed on whether prior programming experience has a positive effect on their performance on their first programming course in the university. The study by Hagan and Markham (2000) suggests that students who knew one or more programming languages prior to attending university showed better academic performance in programming assignments. Watson et al (2014) found a similar effect but suggested that the number of languages that a student knows has no effect. However, Bryne and Lyons (2001), Bergin and Reily (2005) and Ayalew et al (2018) reported no effect on students' performance. Such different results could possibly be explained by differences in educational context and sample size.

Among several interventions that have been shown to improve learning outcomes in CS1 courses, one intervention is to introduce a separate introductory course (termed "CS0") before the beginning of the formal introductory programming course (termed “CS1” in the literature) (Vihavainen, Airaksinen, & Watson, 2014). Some of these courses have used media computation with Python (Sloan & Troy, 2008), Scratch (Rizvi, Humphries, Major, Jones, & Lauzun, 2011) or focused on applications of programming to robotics, games and music (Haungs, Clark, Clements, & Janzen, 2012).

In some educational contexts, like ours, it may not be possible to introduce a CS0 course in the curriculum. Bittencourt et. al. (2015) describes a one-week Scratch workshop that was conducted prior to the beginning of a C programming course in a Brazillian university. Although they viewed the workshop favourably, the effect on the participants’ motivation and performance in the subsequent C course was not reported

Apart from Scratch, online learning platforms have the potential to provide prior knowledge in programming. Kim and Ko (2017) analysed over thirty such online platforms. Many of these platforms were assessed to have sufficient content coverage, and able to provide immediate feedback that was shallow. Both CodeAvengers and CodeCombat (the online platforms that are employed in this study) were assessed to have the required coverage of introductory programming topics, possess features that showed how code is used but lacking features on why each content taught should be used. Participants in the study by Lee and Ko (2015) comparing two online platforms, Gidget and Codeacademy, showed learning gains when measured by a pre-test/post-test format. However, as the participants of this study were recruited from the general public, it is unclear if the results are applicable to the higher education context. In another study, it was found that university students preferred using CodeCombat to the “Robot Turtles” board game in learning basic programming concepts (Kurniawan, Cheung, & Ng, 2019).

We conducted a one-week workshop in 2017 that employed two online learning platforms, CodeCombat and CodeAvengers. CodeCombat is a gamed-based online platform, making use of a game to teach programming, while CodeAcademy is a traditional tutorial platform that has gamified features such as levels, badges, and leaderboard. This workshop was conducted prior to the beginning of The Digital World course, and thus aimed to provide novice learners with some prior knowledge in programming. These online platforms thus act as scaffolding to help these students move through their zone of proximal development in the learning of basic programming concepts (Anderson & Gegg-Harrison, 2013).

We are interested to find out the impact of the two online platforms, game-based and gamified learning platforms, on students’ learning experience. We refer to learning experience as user experience, preference, interest, rationale for using and achievements of learning outcomes. In particular, we want to (1) compare participants’ learning preference of the two platforms, (2) investigate if students continue to use the platform after the workshop on their own and the reasons for their use, and (3) investigate whether there is any significant difference between those exposed to the online platforms in the preparatory workshop and those who are not with regards to the learning outcomes in the official programming course. Our null hypothesis is that there is no significant difference between the two platforms in terms of their learning experience.

Methodology

Education Context

This study was conducted prior to and during first-year introductory programming course “The Digital World” at the Singapore University of Technology (SUTD) in 2017. The majority of students taking The Digital World do not have programming experience. Most students joining SUTD have the GCE-“A”-level qualifications offered by the Junior Colleges (JC), but only a few of such students have taken Computing as a subject. A minority of our students join us from the polytechnics with Engineering or IT diplomas and would have varying degrees of programming ability.

We conducted a five-day, preparatory workshop named “Introduction to Computational Thinking” which introduces students to computational thinking concepts using the Python programming language. This workshop was aimed at students who have little or no programming skills in order to provide them with some programming background at the start of The Digital World. We sent an email to all first-year students inviting them to take part in the workshop. Participation in the workshop is voluntary. The criteria for participation were (1) that students have little or no programming background and (2) students need to attend all the sessions.

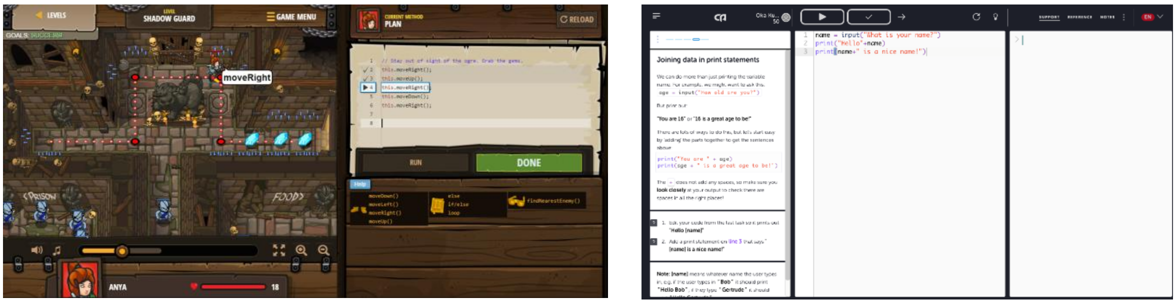
Participants

A total of 457 first year students were enrolled in the programming course The Digital World in the third term of the academic year 2016/2017. From these students, 81 students joined the workshop, and out of these, 48 students gave consent to participate in this study, with 28 students completing the Reflection survey at the end of week six (see

Table 1). Students were aged 18 to 21 years old, with 50% of them female.

Online Platforms

The screenshots of the two online platforms used in this study, CodeCombat and CodeAvengers, are shown in Figure 1. A game-based online platform like CodeCombat makes use of a game-like environment to help users learn programming. Users are engaged in an immersive game activity. This can also be considered as "serious gaming". In the process of game and play, users learn programming starting from the basic syntax and can be up to different programming concepts such as conditionals and iteration. On the other hand, a gamified online tutorial like CodeAvengers uses traditional structured online lessons with gamification elements such as rewards, levels, badges, and a leader board. Thus, instead of an actual game, the platform uses game elements to engage and motivate students. For our study, we purchased the necessary subscriptions for CodeCombat and CodeAvengers, and verified that the content covered by the two platforms are similar and met our requirements.



```
// Stay out of sight of the ogre. Grab the gems.
this.moveRight();
this.moveUp();
this.moveRight();
this.moveDown();
this.moveRight();
```

Joining data in print statements

We can do more than just printing the variable name. For example, we might want to ask this:

`age = input("How old are you?")`

But print out:

"You are 16" or **"16 is a great age to be!"**

There are lots of ways to do this, but let's start easy by 'adding' the parts together to get the sentences above

```
print("You are " + age)
print(age + " is a great age to be!")
```

The + does not add any spaces, so make sure you **look closely** at your output to check there are spaces in all the right places!

1. Edit your code from the last task so it prints out **"Hello [name]"**
2. Add a print statement on line 3 that says " **[name] is a nice name!"**

Note: [name] means whatever name the user types in, e.g. if the user types in "**Bob**" it should print "**Hello Bob**", if they type "**Gertrude**" it should

```
name = input("What is your name?")
print("Hello"+name)
print(name+" is a nice name!")
```

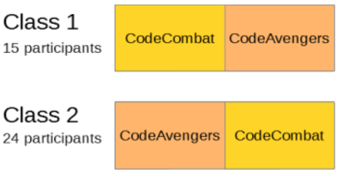
(a) (b)

Figure 1: Interface of (a) CodeCombat, and (b) CodeAvengers.

Study Protocol

We divided the participants of the workshop randomly into two classes, and both classes were taught by several undergraduate teaching assistants (UTA). On the first day of the workshop, all participants did a background survey to assess their programming background and were invited to take part in this study. Those who agreed to take part in the study then did a pre-test programming quiz. Each class started with one of the two platforms, before switching to the other in the middle of the five-day duration, as illustrated in Figure 2.

At the end of the workshop, participants completed a post-test programming quiz and a survey on the two online platforms. The post-test programming quiz is similar in concept and difficulties as the pre-test quiz. We then informed the participants that they will continue to have access to the two online platforms for the next few months, and that a follow-up survey will be held in week six of The Digital World course. This process is summarized in Table 1.


Class 1
15 participants
CodeCombat
CodeAvengers

Class 2
24 participants
CodeAvengers
CodeCombat

Figure 2: The sequence of how the platforms are used during the workshop

Table 1: Measurement Instruments and Its Timing

Timing	Instrument	What It Measures
Start of workshop	Background survey* (10 questions)	Participants programming experience and background
	Pre-test programming quiz^	Programming content and skills before workshop
End of workshop	End of workshop survey (10 questions)	Participants' perception on the workshop and online platforms
	Post-test programming quiz^	Programming content and skills after workshop
Start of course	Background survey of non-workshop participants* (10 questions)	Students programming experience and background
Week six of course	Reflection survey (8 questions)	Participants' perception on the long-term use of online platforms and impact of workshop
Mid-term of course	Mid-term programming assessment (5 questions)	Students learning outcomes

* same survey questions
^ similar content and difficulties

Analysis

Quantitative Analysis

For the statistical analysis of quantitative data from the surveys, quizzes, and assessments, we used programs such as Excel, Tableau, and some Python statistical libraries. We calculated the learning gain (Hake, 1998) and conducted a paired t-test between the pre-test and the post-test scores. We then compared the mid-term programming scores between students who attended the workshop and students identifying as novice learners who did not attend the workshop using an independent samples t-test.

Qualitative Analysis

We asked the participants what they liked and disliked about the two online platforms in the Reflection survey conducted at week six of the official course. A total of 28 students responded on the open-ended questions. Students' open-ended responses were categorized, and frequency counted to understand their sentiments.

Results

Programming Background

Table 2 shows the self-assessment of the participants' programming background. Most participants consider themselves to have zero programming background or novice learners, with four students claiming to have written more than 50 lines. We cross check this background with a programming test which is conducted at the beginning of the workshop. The average for the pre-test was 0.103 out of 1.000 (normalized). There were only three participants who have scores greater than or equal to 0.5 in the pre-test. This agrees with the self-assessment of participants' programming background that most of the participants do not have any programming experience, and only a few of them had some programming background. This is also similar to our other previous studies students' profile (Kurniawan et al., 2019).

Table 2. Participants' Self-reported Programming Background .

Zero*	Novice

0 lines	1 to 10 lines	1 to 10 lines	10 to 50 lines	> 50 lines
24	13	2	4	4

* one participant under “Zero” category choose Others and put a comment instead. Total number is 48.

Learning Experience of Platforms and Workshop

At the End of Workshop survey, we asked participants to respond on a 5-point Likert scale regarding the ease of use, motivation to use, and challenge in using the two platforms. The average score of this result is presented in

Table **3**. In general, participants found both platforms to be easy to use and motivating. The average scores for both platforms on all questions are above 3.0. However, the results suggest that students find CodeAvengers more challenging compared to CodeCombat.

Table 3: Average Likert Score on Participants’ Learning Experience of The Two Platforms

	CodeCombat	CodeAvengers
Easy to Use	3.80	3.76 (↓)
Motivate to Learn	3.61	3.73 (↑)
Challenging	3.32	3.76 (↑)

In the Reflection survey conducted at week six of The Digital World course, we asked the participants for open-ended responses on what they liked and disliked about the two platforms. Overall, the respondents seemed to prefer CodeAvengers over CodeCombat. The most frequent phrases among the comments are shown in Table 4.

Table 4: Frequent Words in Open Ended Responses on The Two Online Platforms

	Liked	**Disliked**
CodeCombat	fun interesting visual graphics interactive	not clear for the concept confusing repetitive too simple or easy
CodeAvengers	clear instruction step by step easy instructions or easy to follow materials or topics	long lessons or too many challenging nil

An analysis of the responses showed that, while 66 % of the students liked the game CodeCombat citing that it was fun, interesting and interactive and 25 % of the students noted that the game helped them to visualize, 25% of them mentioned that they could not relate the game to the concepts learnt in class, and that the game was too abstract and confusing to them. That is, there is a gap in relating the game to academic content and learning. About 37% of the students also found the game to be simple and repetitive.

On the other hand, 100% of the respondents indicated that the gamified tool CodeAvengers was very helpful in teaching programming concepts and programming skills in a structured manner. One major drawback cited was the long time taken with Code Avengers, but this need not necessarily be a disadvantage as learning does require time and practice. Nearly 20% of respondents wanted additional challenging activities with CodeAvengers, adding further support to the earlier inference that students were deeply engaged and wanted more of the learning experiences with CodeAvengers.

Continued Use and Reasons

In the same Reflection survey, we asked whether the workshop has helped them in their course, The Digital World, and increased their interest in the course. We also asked if they continued to use the online platforms after the workshop during this course. Figure 3 shows the results. Students who attended the workshop found that the workshop has helped them in their official programming course (Q2). Moreover, the workshop has increased their interest in learning the programming course (Q3). However, not everyone continues to use the online platform during the official course (Q4). We will present on students' reasons for this below.

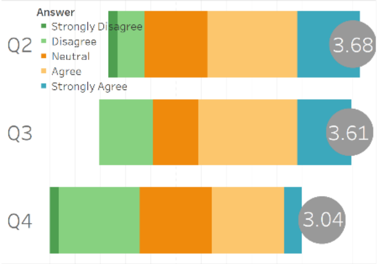


Figure 3: Students found the workshop has helped them in their course, The Digital World, (Q2) and increases their interest in learning The Digital World course (Q3). Only some, however, continue to use the online platform during the course (Q4).

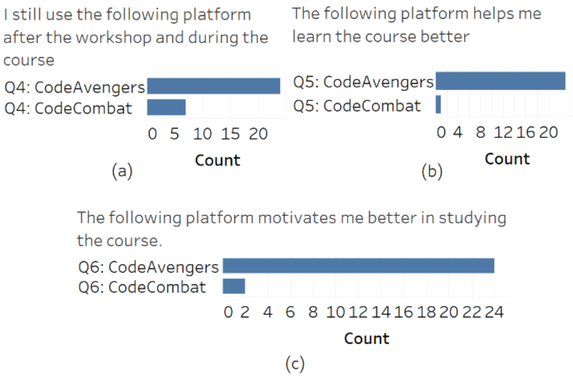


Figure 4: Students still use CodeAvengers after the workshop and continue to use it during the formal course (a). They found CodeAvengers helps them to learn better (b) and motivates them to study the course better than CodeCombat (c).

The results also show strongly students' preference toward CodeAvengers (Figure **4**). They still used it after the workshop and during the official course. Compared to CodeCombat, more participants indicated that CodeAvengers helps them to learn the course better. Hence, these results seem to be consistent with the survey done at the end of the workshop (Table 3).

From Figure 5, we found that some students continue to use the online platforms after the workshop, using it for revision, practice and to learn new topics. This could be why students prefer CodeAvengers to CodeCombat, as CodeAvengers' interface makes it easy for students to do so, while CodeCombat's game-based interface makes it hard to revise any particular topic. The students who no longer use the online platforms mainly cite lack of time and being able to obtain information elsewhere as the reason. These results are also reasonable as we observe that the third term tends to have a high workload for many students.

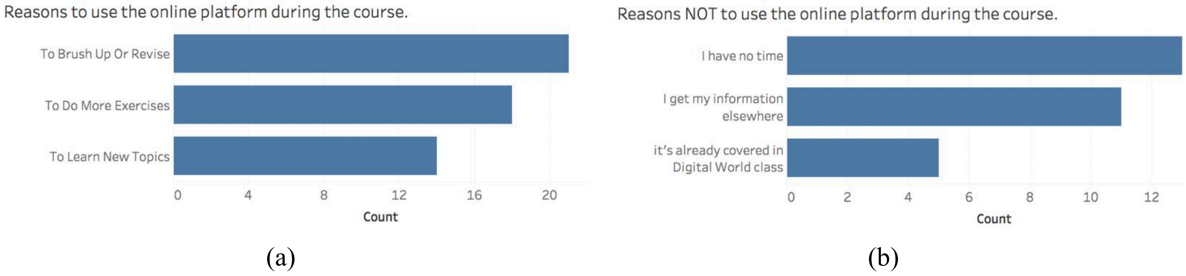


Figure 5: During the course, (a) some participants participants continue using the online platforms for revision and practice, (b) some participants report not using due to the lack of time

Learning Outcomes

Pre- versus Post-test

From the pre-test to the post-test, the mean score of the test increased from 0.103 to 0.405 (out of a normalized value of 1.0). A paired t-test suggested that there is a significant difference between the two test results with $p < 0.01$. Using the formula by Hake (1998), the learning gain was calculated to be 33.1%. Lastly, the number of students with mean scores greater than or equal to 0.5 increases from only three participants in the pre-test to thirty-four participants in the post-test. This shows that more students were able to write Python code at the end of the workshop. We found it is important to check on this result as students only learn Python for the first time in a very limited time frame, i.e. they spent only three hours each day in this five-day workshop.

Workshop Participants versus Non-Participants

Table 5: Mid-Term (MT) Exam Statistics from The Official Programming Course

	Attended Workshop	Did Not Attend Workshop
Mean	0.516 (↑)	0.455
Minimum	0.086 (↑)	0.029
25%	0.371 (↑)	0.314
50%	0.543 (↑)	0.486
75%	0.671 (↑)	0.568
Maximum	0.843 (↓)	1.000

Table 5 shows the statistics of the students' grade in The Digital World course for the Mid-Term exam (MT), and compares the results between those who attended the workshop and those who did not. For the students who did not, we used grades from students who reported that they had zero or little programming background at the start of The Digital World course. Results using independent samples t-test on the mean score of the programming questions suggest that there is no significant difference between the two group of students. This means that the workshop did not translate to a gain in academic performance as compared to those who did not attend the workshop. At the same time, looking at Table 5, we can see that those attended the workshop tend to have higher scores in the five-number summary statistics except for the maximum score.

Discussion

Learning Experience

Most students found the two online platforms, CodeAvengers and CodeCombat, easy to use. However, it seems that students find CodeAvengers slightly more motivating, and surprisingly, more challenging compared to CodeCombat. This challenging factor could be one of the sustained motivating factors for students as we see in the Reflection survey at week six. CodeCombat is more fun, but those who attended this workshop have The Digital World course in mind. Therefore, they wanted something that can help them in their official course. CodeAvengers seems to be the better platform for this purpose. Both the quantitative data from the end of workshop survey and qualitative data from the Reflection survey attested to this.

The qualitative data gave insights into why students preferred CodeAvengers for learning. Participants cited that the gamified platform was structured, sequential and progressive, relevant to the academic content, and easy to use. Although participants found CodeCombat easy to use and visually interactive, but they found it repetitive and harder to relate the game to the academic content of The Digital World. This observation seems to indicate that CodeAvengers was in the comfortable zone of proximal development of students (Anderson & Gegg-Harrison, 2013) whereas CodeCombat was too simple. According to the concept of zone of proximal development, students' learning happens when they are comfortably stretched; if too easy, students get disengaged and bored, and if too challenged, feel overwhelmed and dejected. In our case, students who preferred the gamified platform seem to be comfortable being challenged and some students even wanted more challenging tasks, and this suggests that the online platform needs to cater to diverse learners at both extremes as well.

An interesting observation was that students found CodeAvengers to be relevant and useful for academic learning while CodeCombat was difficult to relate to. This could be because our students are novice learners and novice learners typically need more structuring and scaffolding (Awbi, Whalley, & Philpott, 2015). Also, the gamified-learning used deductive teaching approach while game-based learning uses inductive teaching. Typically, it is found that novice learners find deductive mode of teaching and learning easier to follow than inductive teaching and inductive approach can be complex and abstract for them (Felder & Silverman, 1988). Transferability of knowledge through deductive approach is higher in deductive methods than inductive methods. As the workshop was standalone and conducted prior to the official programming course, students were left on their own to translate the workshop content to the programming course. Hence, one implication is that if we are to use the game-based platform in our teaching, we will need to scaffold students' learning to help them relate to the academic content, e.g. using reflection questions or other suitable activities.

We found that only 30% of participants continued to use the online platforms and this was the gamified platform, CodeAvengers. The reasons suggested was that this helped them to revise, practice and prepare for academic learning in The Digital World. This suggests two things – one is that initial perception and interest can be temporary and therefore we must be cautious in interpreting student perception right after exposure. It is good to also measure perceptions throughout the learning experience and at the end of the learning journey or even long after that. This is referred to as "situational interest" (Rotgans & Schmidt, 2011).

Also, in our context, it seems that game-based learning may lead to "surface- learning" and gamified learning leads to "deep learning" (Biggs, 1988). According to Biggs, deep learning refers to engaged, sustained and proactive learning for the purpose of learning. Surface learning is superficial and taking part in the learning activity for the sake of it or get through the course. The observation that students continued to use the gamified platform rather than the game-based platform over an extended period of time and on their own, when given a free choice indicates that the gamified platform was perceived to be more valuable and leading to sustained and self-directed learning. Hence, this implies that we need to see what sort of tools lead to scaffolding and engages students in deep learning.

Learning outcomes

We had found no difference in the mid-term exam scores between the workshop participants and those who did not participate in the workshop. However, the workshop participants reported in the survey that the workshop helped them to learn programming in the official course and increased their interest in learning the course (Figure 3). Hence, we can conclude that the workshop had a positive impact on the participants, even if it did not translate to a significant gain in their academic performance at the mid-term exams. On top of knowledge, learning outcomes also comprises of skills and attitudes. Given that the gamified online platform, CodeAvengers, added to the interest to learn, we infer that it is useful to use this in our teaching of programming to our novice learners.

The reason why the workshop did not result in a difference in the mid-term exam scores deserves some consideration. Firstly, the content examined in the mid-term exam was much larger than scope of the workshop, hence, the instruction given during the course could have helped to bring both groups of students to a similar academic level. The questions in the mid-term exams may have been too difficult and cognitively demanding, but assessing the questions is beyond the scope of the current study. Lastly, the workshop was run as a standalone event and not integrated into our course, The Digital World, which may have reduced its impact.

Lessons Learnt

The findings suggest that while games and gamification can be both engaging, they may engage students in different ways. While games can be fun, students may not be able to relate the gaming and fun element to the academic content. The context of game and learning may be distant that students are not able to link the two. This concept is called situational learning that learning happens in context. An implication of this is that as teachers, if we are to use games in teaching, we will need to build in opportunities for students to discuss and link the games to the actual lesson. Another inference is that perhaps the game may be more useful as a tool to interest learners in programming compared to being a teaching or scaffolding tool. Also, this game may be more useful for younger learners. It is also possible that the context of The Digital World course creates an academic environment that hinder people to play games. If game-based learning is to be used together with any academic course, it should be integrated together inside the course and some of the assessments should include achievements done in the game.

We also learnt that the perception surveys may be subject to situational interest and it is good to monitor interest and learning over time. Also, the results from this study must be taken in relation to the context. For instance, we cannot conclude our results to be saying that gamified learning is better than game-based learning. We need to see the contextual factors for instance, the level and complexity of the subject matter, the extent of time we have for learning, the mode of learning (face to face, blended), the background and prior knowledge of students, learning outcomes and purpose of learning, to name a few factors. We encourage our readers to consider their contexts in extrapolating our findings to suit the context and use a scholarly and evidence-based enhancement of teaching practices.

The analysis also indicates that it is critical to structure the content and introduce the various programming skills in a step by step manner in teaching programming skills to students who may not have sufficient programming background. This observation aligns with Vygotsky's concept of zone of proximal development which suggests that students' learning needs to be scaffolded – especially when students do not have sufficient prior knowledge, and they have to achieve a cognitive jump (Vygotsky, 1978).

The student responses had not mentioned any aspect of the gamified learning such as collecting badges etc as motivating factors and seemed to be drawn to the structured way of learning. This interesting observation indicates that, with well-structured and planned teaching tools/aids such as CodeAvengers, it is possible to interest the students in the actual learning of content by intrinsically motivating them rather than relying on extrinsic motivating factors such as points and badges, which may be short-lived.

Overall, what we have learnt is that it is important to use the right game/gamified tool to aid teaching of programming skills. We need to ensure that the game/gamified tools are pitched at the prior knowledge of students, it provides ample opportunities for the diverse learners catering to both the novice and advanced learners, and that the teacher designs and integrates the use of the tool in the actual lesson so that students are able to relate to the academic content. Since learning through such resources will take up additional time, teachers may also want to use flipped learning so that students could use the class time effectively.

Conclusion

In this study, we ran a one-week workshop to compare two kinds of online platforms, a game-based platform and a gamified online tutorial platform, to help students to learn programming. This was done prior to the students' formal programming course. We found that students need not play a game to be motivated. Participants preferred CodeAvengers, the gamified platform, as it is challenging, provided content in a structured manner and had exercises that help students in their revision for their programming course. Moreover, the way that these platforms are included in the official course may affect their usage. Lastly, the interface of the platform and the nature, either game-based or gamified platform, should be designed in such a way to fulfill the needs of how the students would use it. Hence, in our context, the gamified online platform had features that met their needs. All these considerations should be taken into account in the choice of online platforms by instructors.

References

Anderson, N., & Gegg-Harrison, T. (2013, March 8). Learning computer science in the "comfort zone of proximal development." 495. https://doi.org/10.1145/2445196.2445344

Awbi, N. K., Whalley, J. L., & Philpott, A. (2015). Scaffolding, the zone of proximal development, and novice programmers. Journal of Applied Computing and Information Technology, 19(1).

Ayalew, Y., Tshukudu, E., & Lefoane, M. (2018). Factors Affecting Programming Performance of First Year Students at a University in Botswana. African Journal of Research in Mathematics, Science and Technology Education, 22(3), 363-373. https://doi.org/10.1080/18117295.2018.1540169

Bergin, S., & Reilly, R. (2005). Programming: Factors that influence success. Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, 411-415. https://doi.org/10.1145/1047344.1047480

Biggs, J. B. (1988). Assessing student approaches to learning. Australian Psychologist. https://doi.org/10.1080/00050068808255604

Bittencourt, R. A., Dos Santos, D. M. B., Rodrigues, C. A., Batista, W. P., & Chalegre, H. S. (2015). Learning programming with peer support, games, challenges and scratch. Proceedings - Frontiers in Education Conference, FIE. https://doi.org/10.1109/FIE.2015.7344222

Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education - ITiCSE '01, 49-52. https://doi.org/10.1145/377435.377467

Felder, R. M., & Silverman, L. K. (1988). Learning and Teaching Styles in Engineering Education. Engineering Education, 78(7), 674-681.

Hagan, D., & Markham, S. (2000). Does it help to have some programming experience before beginning a computing degree program? Proceedings of the 5th Annual SIGCSE/SIGCUE ITiCSEconference on Innovation and Technology in Computer Science Education - ITiCSE '00, 25-28. https://doi.org/10.1145/343048.343063

Hake, R. R. (1998). Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. American Journal of Physics, 66(1), 64-74. https://doi.org/10.1119/1.18809

Haungs, M., Clark, C., Clements, J., & Janzen, D. (2012). Improving first-year success and retention through interest-based CS0 courses. Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, 589-594. https:// doi.org/10.1145/2157136.2157307

Kim, A. S., & Ko, A. J. (2017). A Pedagogical Analysis of Online Coding Tutorials. Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, 321-326. https://doi.org/10.1145/3017680.3017728

Kurniawan, O., Cheung, N.-M., & Ng, G. S. (2019). University Students' Perspectives on The Impact of Games in Preparing for a Programming Course. Asian Journal of the Scholarship of Teaching and Learning, 9(1), 5- 29.

Lee, E. A., & Messerschmitt, D. G. (1998). Engineering an education for the future. Computer, 31(1), 77-85. https:// doi.org/10.1109/2.641980

Lee, M. J., & Ko, A. J. (2015). Comparing the Effectiveness of Online Learning Approaches on CS1 Learning Outcomes. Proceedings of the Eleventh Annual International Conference on International Computing Education Research - ICER '15, 237-246. https://doi.org/10.1145/2787622.2787709

Leitão, A., Cabecinhas, F., & Martins, S. (2010). Revisiting the Architecture Curriculum: The programming perspective. FUTURE CITIES [28th ECAADe Conference Proceedings], 81-88.

Linn, M. C., & Dalbey, J. (1985). Cognitive Consequences of Progr amming Instruction: Instruction, Access, and Ability. Educational Psychologist, 20(4), 191-206. https://doi.org/10.1207/s15326985ep2004_4

Lister, R., Seppälä, O., Simon, B., Thomas, L., Adams, E. S., Fitzgerald, S., ... Sanders, K. (2004). A multi- national study of reading and tracing skills in novice programmers. ACM SIGCSE Bulletin, 36(4), 119-150. https:// doi.org/10.1145/1041624.1041673

McCartney, R., Boustedt, J., Eckerdal, A., Sanders, K., & Zander, C. (2013). Can first-year students program yet? Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research, 91-98. https://doi.org/10.1145/2493394.2493412

McCracken, M., Wilusz, T., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., ... Utting, I. (2001). A multi- national, multi-institutional study of assessment of programming skills of first-year CS students. ACM SIGCSE Bulletin, 33(4), 125-180. https://doi.org/10.1145/572139.572181

Qian, Y., & Lehman, J. (2017). Students' Misconceptions and Other Difficulties in Introductory Programming. ACM Transactions on Computing Education, 18(1), Article 1. https://doi.org/10.1145/3077618

Rizvi, M., Humphries, T., Major, D., Jones, M., & Lauzun, H. (2011). A CS0 course using Scratch. Journal of Computing Sciences in Colleges, 26(3), 19-27.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. Computer Science Education, 13(2), 137-172. https://doi.org/10.1076/csed.13.2.137.14200

Rotgans, J. I., & Schmidt, H. G. (2011). Situational interest and academic achievement in the active-learning classroom. Learning and Instruction. https://doi.org/10.1016/j.learninstruc.2009.11.001

Sloan, R. H., & Troy, P. (2008). CS 0.5 : A Better Approach to Introductory Computer Science for Majors. Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education, 271-275. https:// doi.org/10.1145/1352322.1352230

Vihavainen, A., Airaksinen, J., & Watson, C. (2014). A systematic review of approaches for teaching introductory programming and their influence on success. Proceedings of the Tenth Annual Conference on InternationalComputing Education Research, 19-26. https://doi.org/10.1145/2632320.2632349

Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes. In M. Cole, V. John-Steiner, S. Scribner, & E. Souberman (Eds.), Mind in society: The development of higher psychological processes. L. S. Vygotsky. Oxford, England: Harvard U Press.

Watson, C., Li, F. W. B., & Godwin, J. L. (2014). No tests required: comparing traditional and dynamic predictors of programming success. Proceedings of the 45th ACM Technical Symposium on Computer Science Education - SIGCSE '14, 469-474. https://doi.org/10.1145/2538862.2538930

Please cite as: Kurniawan, O., Lee, N.T.S., Sockalingam, N. & Pey, K.L. (2019). Game-Based versus Gamified Learning Platform in Helping University Students Learn Programming. In Y. W. Chew, K. M. Chan, and A. Alphonso (Eds.), *Personalised Learning. Diverse Goals. One Heart. ASCILITE 2019 Singapore* (pp. 159-168).