

Review

Gamifying Computer Science Education for Z Generation

Hadeel Mohammed Jawad ^{1,*} and Samir Tout ²

¹ Department of Computer Science and Engineering, Oakland University, Rochester, MI 48309, USA

² School of Information Security & Applied Computing, Eastern Michigan University, Ypsilanti, MI 48197, USA; stout@emich.edu

* Correspondence: hjawad@oakland.edu

Abstract: Generation Z members use their smart devices as part of their everyday routine. Teaching methods may need to be updated to make learning materials more interesting for this generation. This paper suggests gamifying computer science subjects to enhance the learning experience for this generation. Additionally, many students face difficulty in understanding computer science materials and algorithms. Gamifying computer science education is one of the suggested teaching methods to simplify topics and increase students' engagement. Moreover, the field of computer science is dominated by males. The use of gamification could increase women's interest in this field. This paper demonstrates different techniques that were developed by the researchers to employ gamification in teaching computer science topics. The data was collected at the end of the two different courses. Results show that students enjoyed the suggested teaching method and found it useful. This paper also demonstrates two tools and their gamification elements. These tools were developed by the researchers to help people learn computer programming and information security.

Keywords: gamifying; computer science education; data structures; sorting algorithms; Code Genie; gamifying cybersecurity education



Citation: Jawad, H.M.; Tout, S.

Gamifying Computer Science

Education for Z Generation.

Information **2021**, *12*, 453.

[https://doi.org/10.3390/](https://doi.org/10.3390/info12110453)

[info12110453](https://doi.org/10.3390/info12110453)

Academic Editor: Christo Dichev

Received: 8 August 2021

Accepted: 29 October 2021

Published: 1 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Members of the Z generation have already been born into a digitalized world where smart tools are part of their everyday routine and online games are popular among them. On the other hand, the teaching methods of the 20th century that are still being used in the current educational system may not be efficient enough for these digital natives. The introduction of gamification could provide a solution for the difficulties in educational systems for this group as they are familiar with the various aspects of the gamification technology.

Computer programming is among the hard subjects that need to be simplified by the instructor. Several studies found that the use of gamified learning increases students' learning motivation and engagement [1–3]. Seaborn and Fels (2015), found that applying gamification in any educational context contributes to increasing the student's engagement in learning while providing feedback from such learning [4]. Gamification is the process of adding game elements into a non-game context [1,5].

In his article, Dodge (2018) mentioned that there is no better way to engage students in a code-oriented lesson than through a game. In this way, they hardly feel like they are studying at all [6]. He also listed and described a variety of coding games that help learners break into the world of computer programming such as Scratch and Swift Playgrounds, Osmo, Kodable, Tynke, LEGO Mindstorms, Code Monkey, CodaKid, Blockly, CodeCombat, Code.org, Codemoji, Stencyl, Lightbot, Bitsbox, Code Karts, Roblox [6,7] Scratch and Swift Playgrounds are shown in Figure 1. Most of those tools were designed for K12, however, several studies found that gamification was useful for college students as well [1–3,6,8].

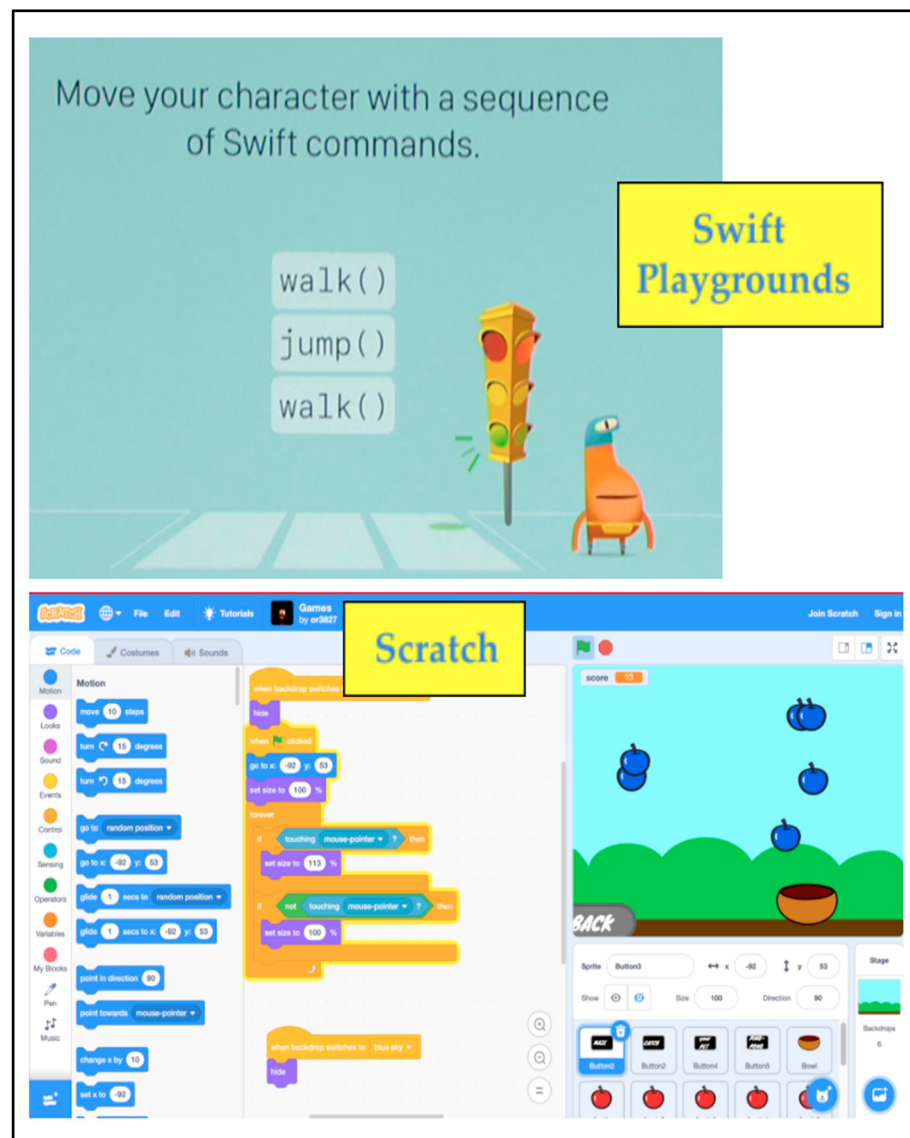


Figure 1. Tools for code learner: Swift Playgrounds from Apple and Scratch from MIT.

Varannai, Sasvari, and Urbanovics (2017) conducted an empirical study on 86 university students where they asked the sample subjects to use Kahoot gamified environment as part of their lectures. The result shows that the positive attitude, good experience, and tool ease of use contributed to improving student performance [9]. Piteira, Costa, and Aparicio (2017) designed and implemented an online course of fundamentals of programming that includes gamification elements. The course was attended by 108 freshmen. Students' attitude toward the progress bar, leader board, badges and points was surveyed. Students found those gamification elements useful and motivating [10].

This paper demonstrates three different techniques of using gamification in computer science education for college students. In the first one, activity games were developed and used in teaching different data structures algorithms and concepts in a university-level course. In the second, game coding examples were added to the teaching materials of the mobile app development course. Both courses were designed for both undergraduate and graduate students. The third technique is to add gamification elements into two different educational tools to enhance the learning experience. Unlike the first two techniques, those two tools could be used by anyone; however, they also contain university-level course materials.

2. Materials and Methods

Computer science algorithms and topics are not always easy to learn. Educators recommend using games in teaching a complex subject to make it easier and more interesting to students. In this study, different kinds of gamification techniques were used to help teach different computer science topics as follows:

- The Use of Gamification in Teaching Data Structure Algorithms;
- The Use of Gamification in Teaching Mobile App Development;
- The Use of Gamification in Code Genie Tool and a Cybersecurity App.

Data were collected from students who attended the data structure and the mobile app development courses. These are university-level courses for both graduate and undergraduate students. Students were asked about their experience in these two courses by the end of the semester.

The following section explains the group activity games that were developed and used to help teach different data structures algorithms and concepts. The next section demonstrates the game-coding examples that were used in teaching a mobile app development course.

The third and fourth sections explain the gamification elements that were added to two different educational tools: the Code Genie and the “Be Aware of the Hacker” app. Code Genie is a web-based IDE (Integrated Development Environment) that was developed to learn the JavaScript language and the second educational tool is an app that was developed to educate people about different information security subjects. These two tools were developed by the leading researcher of this paper.

2.1. *The Use of Gamification in Teaching Data Structure Algorithms*

In this study, practical games or class activities were developed and used to help students learn different data structure sorting algorithms in a university-level course. The 15-week course was designed for both graduate and undergraduate students. The activities included engaging role-playing for each student in the context of traditional Computer Science algorithms such as Bubble Sort, Insertion Sort, and Selection Sort; in addition to the use of depictions to reinforce key concepts such as Stacks and Queues. After getting the human subjects’ approvals from the university, students were asked to fill out an online questionnaire by the end of the course. The results showed an agreement on the usefulness of the activities and how such activities simplify rather complicated algorithms and make them easier to understand.

The bubble sorting game is one of the activities that were used in this study, and it is explained in the following paragraphs. For the other game activities that were used to explain other algorithms, the reader is referred to [11]

The bubble sorting algorithm includes two nested for-loops as shown in Figure 2 [12]. In the inner loop, the algorithm compares the first two numbers in the array. If the one on the left is greater, the numbers need to be swapped. The algorithm moves one position right to compare the next two numbers and continues comparing, swapping, and moving to the right until the largest number reaches the right end. When the highest number is pushed to the far right, the outer loop starts over at the left end of the line to sort the rest of the numbers. The class activity that was used to explain this algorithm is shown in Figure 2. The notation N## is used to refer to a student and the number that he/she is holding; for example, N55 refers to a student with the number 55. Six students were asked to participate in this game. When these students left their seats and stood in front of other students, they were given paper plates with numbers. An instructor can also use T-shirts with numbers to let students feel like a team.

```

public void bubbleSort()
{
    int out, in;

    for(out=nElems-1; out>1; out--)    // outer loop (backward)
        for(in=0; in<out; in++)        // inner loop (forward)
            if( a[in] > a[in+1] )        // out of order?
                swap(in, in+1);          // swap them
} // end bubbleSort()

```

Figure 2. Bubble Sort.

In Figure 3, a student on the left (N55) is asking the student next to him/her (N77): “Am I greater than you?” If the answer was “yes” the two students need to swap their position, otherwise, the second student (N77) needs to ask the students next to him/her (N99) the same question and so on until the student with the highest number goes to the right end. The process is repeated four times ($out = 6-1$ to $out = 2$) to sort the six numbers. The instructor could have two teams with two T-shirt colours. In this case, the two teams would compete on sorting the numbers as fast as possible. A small gift could be given to the winning team with a photo for memory to increase students’ motivation.

2.2. The Use of Gamification in Teaching Mobile App Development

In an Android mobile development course, several game examples were used to explain different topics. The data were collected by the end of the course to explore the effect of using this technique on students’ interest in programming. The results suggest that Android development could increase students’ interest in programming and the use of the game examples was effective.

The data were collected from 17 college students ($N = 17$) at the end of a 15-week course and in the classroom setting from all students at the same time using Google Forms. The course name was Android Development Using Java and the leading researcher of this study was the instructor. During this course, students learned how to create Android mobile apps using Java language and how to publish their apps on Google Play. The sample was composed of college students who are trying to get their associate degree. Android Studio was the main IDE that was used to create apps in this course. Currently, Java and Kotlin programming languages are used to develop apps with Android Studio. Java language was used for this course. However, some students submitted their final projects in the Kotlin language.

A study shows that integrating art and animation in teaching programming increases students’ interest in writing more code [13]. The course material of this study was designed by the leading researcher to include art, animation, and game examples. It included a variety of gaming examples such as “Flappy Bee”, “Matching the Pictures”, and “Pop the Balloon” as shown in Figure 4. It also included examples that utilized most of the available graphical user interface (GUI) components or widgets such as radio buttons, checkboxes, images, buttons, slider, and different types of Android layouts. Examples also included the use of lists, dropdown menus, web view, SQLite database, and search queries. Figure 5 shows other app examples.

1	N55: Hey Neighbor N77: Yes N55: Am I greater than you? N77: No	
2	N77: Hey Neighbor N99: Yes N77: Am I greater than you? N99: No	
3	N99: Hey Neighbor N66: Yes N99: Am I greater than you? N66: Yes N99: We need to swap	
4	N99: Hey Neighbor N11: Yes N99: Am I greater than you? N11: Yes N99: We need to swap	
5	N99: Hey Neighbor N88: Yes N99: Am I greater than you? N88: Yes N99: We need to swap	
1	N55: Hey Neighbor N77: Yes N55: Am I greater than you? N77: No	
2	N77: Hey Neighbor N66: Yes N77: Am I greater than you? N66: Yes N77: We need to swap	
3	N77: Hey Neighbor N11: Yes N77: Am I greater than you? N11: Yes N77: We need to swap	
4	N77: Hey Neighbor N88: Yes N77: Am I greater than you? N88: No	
1	N55: Hey Neighbor N66: Yes N55: Am I greater than you? N66: No	
2	N66: Hey Neighbor N11: Yes N66: Am I greater than you? N11: Yes N66: We need to swap	
3	N66: Hey Neighbor N77: Yes N66: Am I greater than you? N77: No	
1	N55: Hey Neighbor N11: Yes N55: Am I greater than you? N11: Yes N55: We need to swap	
2	N66: Hey Neighbor N11: Yes N66: Am I greater than you? N11: Yes N66: We need to swap	

Figure 3. Bubble Sort Class Activity.



Figure 4. Games examples that were used in the teaching mobile app development course.

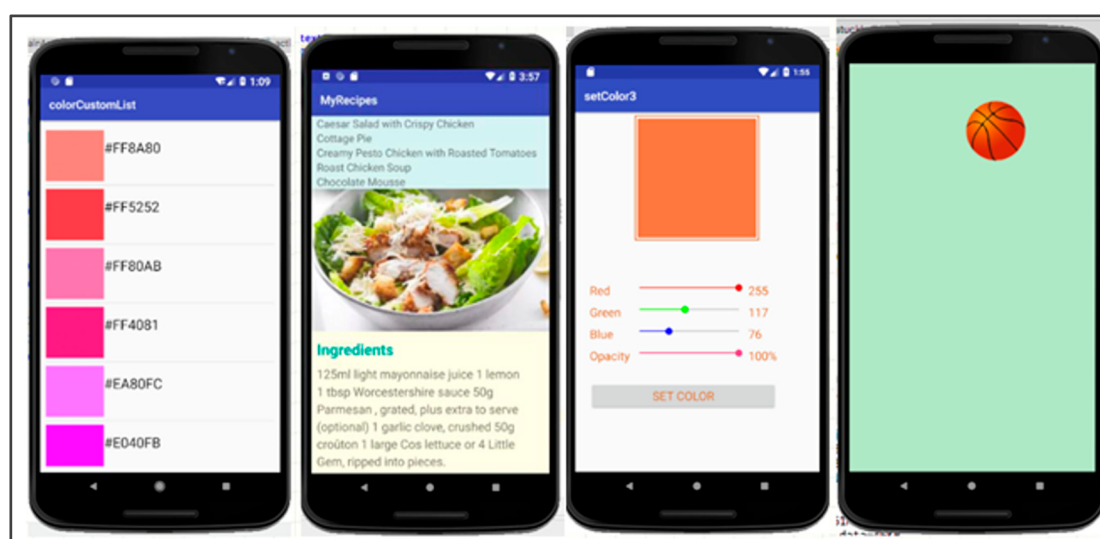


Figure 5. Other examples that were used in teaching the mobile app development course.

Besides assignments and exams, students had to develop their own apps as final projects. Students' projects were of different categories. Most of the students submitted game apps; others submitted other kinds of apps such as map app, an app for saving favourite music and videos, an app for chemical engineers, and other apps that included the use of the SQLite database. Students also had to present their app projects by the end of the semester. With the rich course materials, students were exposed to many topics of Android development and gained the needed skills. Having adequate exposure to the material increased the study validity.

2.3. The Use of Gamification in Code Genie Tool

The Code Genie programming environment was built by the leading researcher of this paper to encourage high school students to code in the JavaScript language. This environment was built to integrate art and animation in teaching computer programming [14]. The environment includes a variety of coding templates including simple games such as flappy bird and matching the picture as shown in Figures 6 and 7. This tool was tested

and used in three coding workshops. Pre-test-post-test experimental methodology and the data were collected from 65 high school students participated in the workshop. The results show that the students' confidence in answering the programming questions increased in the post-test for both genders [15]. The participating students had several suggestions in their feedback comments. One of the common suggestions was adding a tutorial section to the environment. Gamification elements and the suggested tutorial section were added to the Code Genie web Integrated Development Environment (IDE) in the following update after the workshops [16]. The following flowchart, see Figure 8, explains the gamification elements that were added to the tool and demonstrates how a learner could earn points and badges while coding and running his/her code.

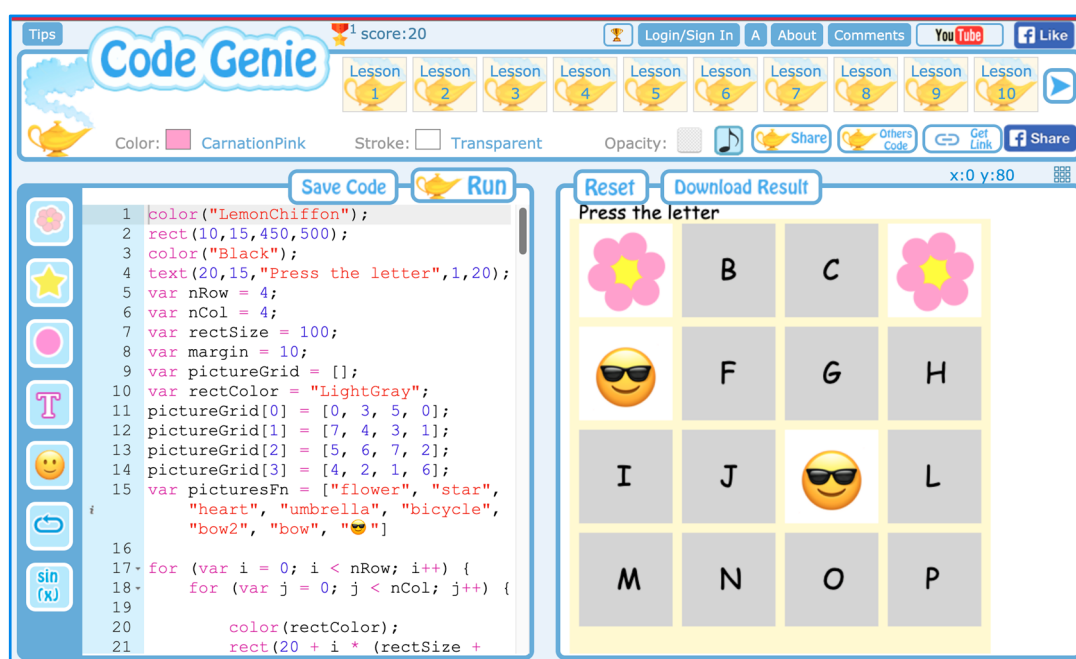


Figure 6. Matching the picture game in the Code Genie Web IDE.

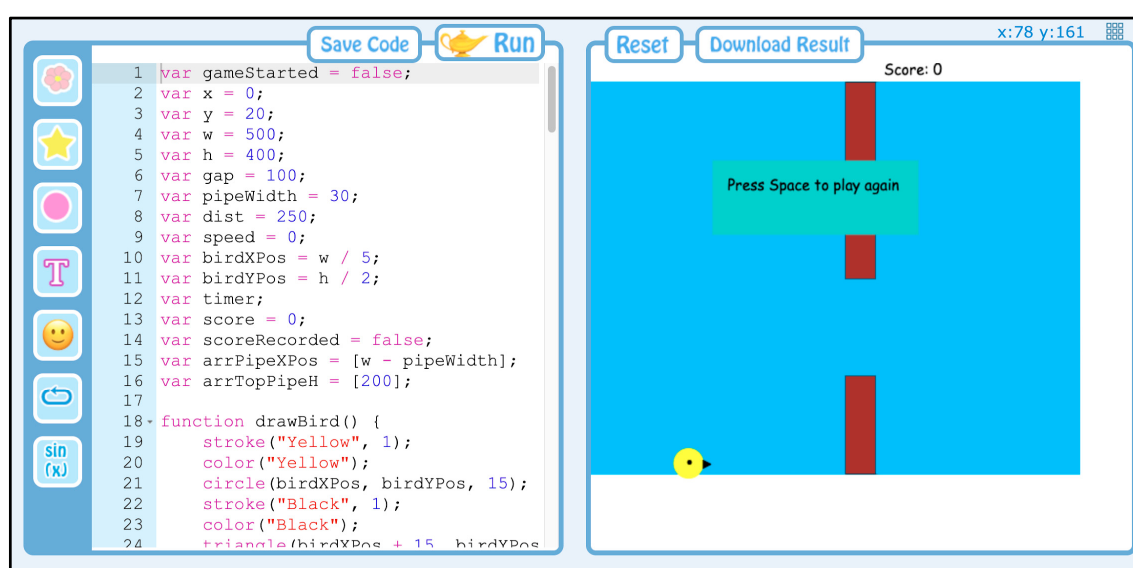


Figure 7. Flappy Bird game in the Code Genie Web IDE.

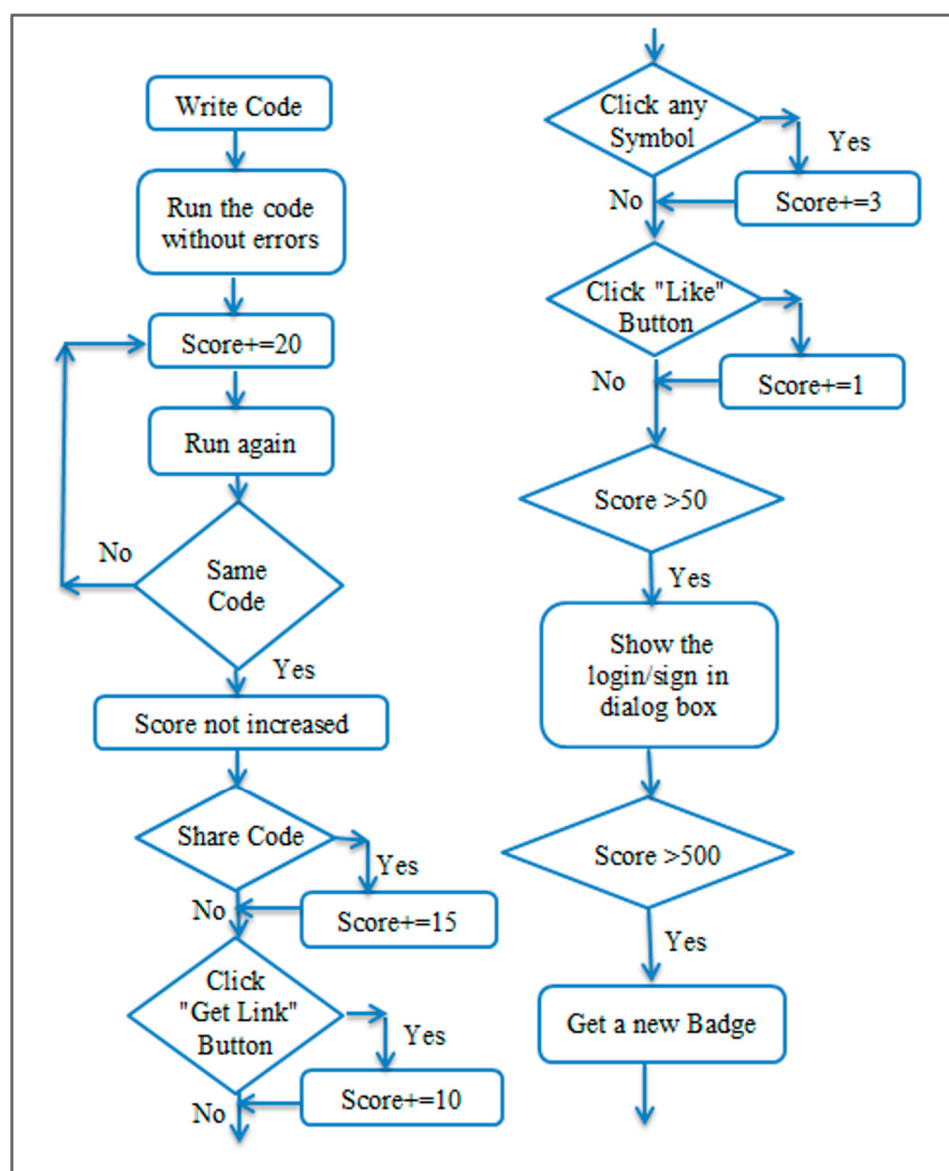


Figure 8. Code Genie Gamification Flowchart.

2.4. The Use of Gamification in Cybersecurity App

Gulf countries such as Saudi Arabia and the United Arab Emirates are increasingly becoming the targets of sophisticated cyberattacks that are aimed at stealing personal data and, in some cases, exposing state and trade secrets [17,18]. As many of these people fall victim to cyber-attacks, more awareness about cybersecurity is needed to protect their social media, bank, and email accounts. This section explains the gamification elements that were added to an app developed by the researcher to educate Arab-speaking people in the Middle East and North Africa (MENA) about cybersecurity and to increase their awareness of information assurance and cybercrimes [19]. The app was developed for Android, and it was called “Be Aware of the Hacker”. Figure 9 shows screenshots of the app.



Figure 9. The Question View and the side menu of the “Be Aware of the Hacker” App.

The app is currently available at Google Play, and it includes three information assurance sections: multiple choice questions, terms, and articles. Currently, it includes 20 multiple-choice questions, 15 term definitions, and 30 article links.

The questions section includes different situations that a user needs to respond to while being online. Such as what to do when you receive a link from your manager, how to respond when you see a warning dialog box asking you to click a button while browsing a website, when you need to be suspicious about some online activities, how to be aware of the social engineering, what is Ransomware, and what situation is considered phishing from different choices, what is the difference between http and https. The questions were translated to Arabic from the Quizizz learning platform [20,21].

The terms section includes definitions of some common cybersecurity terms, such as Ethical Hacking, Social Engineering, Two-Factor Authentication, Honeypot, Firewall, Rootkit, Worm, Trojan, Virus, Botnet. The term and the article sections’ data were collected from different technical websites such as Kaspersky Middle East and The Arabic News Gate for Information Technology [22,23].

The article section educates users about different security topics and to increase their awareness about different situations, such as how to create strong passwords, or how to delete the saved ones from their computers, how to deal with two-factor authentication, what is biometric authentication, what is phishing emails, what are DDoS attacks, and what are the different ways that hackers could use to steal your bank account while shopping online.

Adding gamification elements could increase users’ engagement and their motivation to learn. The app included multiple-choice questions and when a user answers correctly his/her scores will increase by 10 points. An encouragement message is displayed briefly with each right answer. The scores are also increased by five when the user reads an article or a term definition. The scores are also decreased by 10 with each wrong answer.

When a user answers a question, the answer button is disabled to allow one answer. However, a user can visit the same question and answer again using the previous and the next buttons. The answer options are shuffled with each visit and the questions are shuffled each time the user opens the app. An array of the correct questions was used to make sure that the scores were increased once in case the user tried to answer the same question multiple times. The scores were saved and retrieved from the phone’s local storage whenever the app closed and opened again. Figure 10 shows the gamification flowchart of the app and Figure 11 shows the code for the onPause and onResume methods that were used for this purpose.

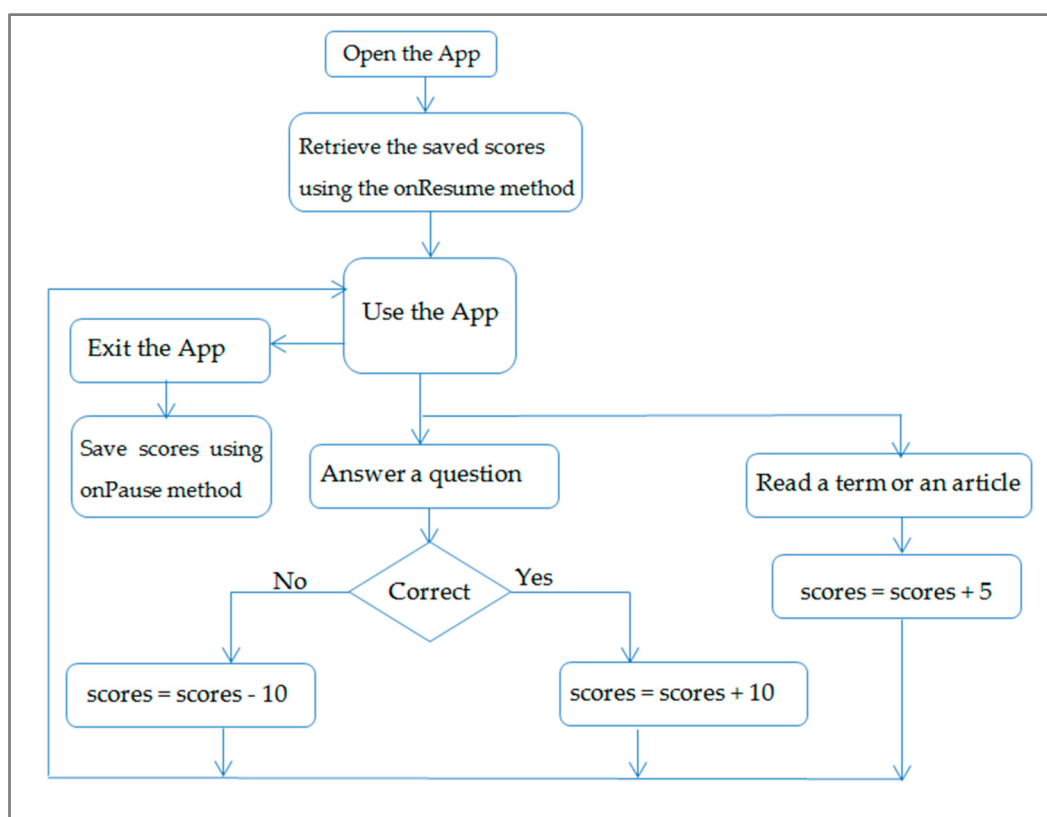


Figure 10. Gamification in “Be Aware of the Hacker” Android App.

```

SharedPreferences savedValues;

protected void onPause() {
    SharedPreferences.Editor editor = savedValues.edit();
    editor.putInt( s: "scores", scores);
    editor.putInt( s: prevCorrQues + "_size", noOfQuestions);

    for (int i = 0; i < prevCorrQues.length; i++)
        editor.putInt( s: prevCorrQues + "_" + i, prevCorrQues[i]);
    editor.commit();
    super.onPause();
}

protected void onResume() {
    super.onResume();
    scores = savedValues.getInt( s: "scores", i: 0);
    int size = savedValues.getInt( s: prevCorrQues + "_size", i: 0);
    for (int i = 0; i < size; i++)
        prevCorrQues[i] = savedValues.getInt( s: prevCorrQues + "_" + i, i: -1);
    scoresTV.setText(" :نقاط" + scores);
}

```

Figure 11. The onPause and onResume methods of the App.

3. Results

The data that were collected from students in the two courses, the data structures course and the Android mobile development course, are analysed in the following two sections.

3.1. The Data Structures Course

For the data structures course, the data was collected from 40 college students by the end of the course. Most of the students were males (75%) and 25% were female students.

17.5% were graduate students and 20% were seniors. Most of the students were White and 20% were Asian. Figure 12 shows the demographic data of the participants.

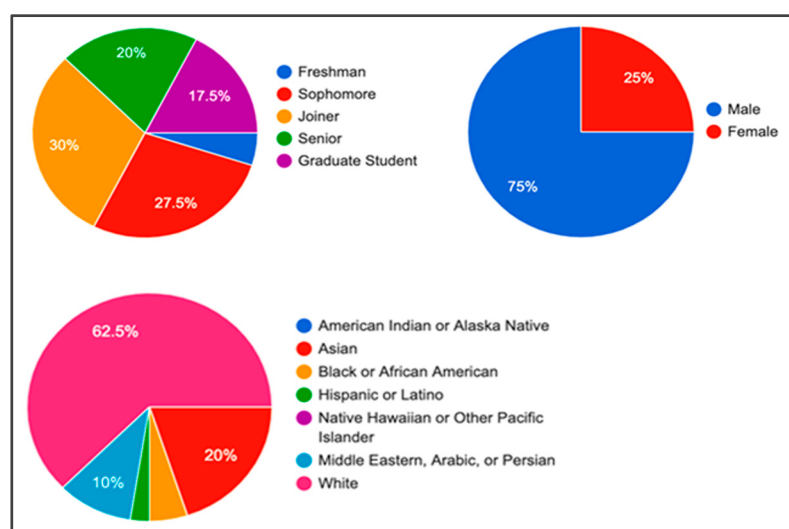


Figure 12. The demographic data of the participants.

When students were asked if they enjoyed the class activities in teaching different algorithms, 80% agreed, 17.5% were neutral, and only one student disagreed, see Table 1.

Table 1. Students' responses to the survey questionnaire for the Data Structures course.

	Agreement %	Neutral %	Disagreement %
Enjoyment	80%	17.5%	2.5%
Usefulness	85%	10%	5%
Simplifies the algorithms	95%	5%	0%
Engagement	70%	22.5%	7.5%
Interest in CS	47.5%	32.5%	15%

The agreement percentage was better (85%) when students were asked about the usefulness of the class activities; only two students (5%) disagreed. When the question was more specific, almost all students agreed (95%) that the use of the class activities simplifies the algorithms and makes them easier to understand. Only 5% were neutral about this question and the disagreement percentage was zero. The agreement percentage decreased (70%) when students were asked whether the class activities enhance students' engagement, 22.5% were neutral, and 7.5% disagreed on this question. Since the students are already in computer science programs and they have an interest in their major, the disagreement percentage was 47.5% when they were asked if the class activities increase students' interest in the computer science field. 32.5% of the students were neutral about this question, and 15% disagreed.

As shown in Figure 13, the agreement percentage of the activities' enjoyment for the female students was 100% while it was 73.3% for the male students. Similarly, the agreement percentage of the activities' usefulness was also 100% for females while it was 80% for males. This indicates that females enjoyed the game activities and found them useful more than male students.

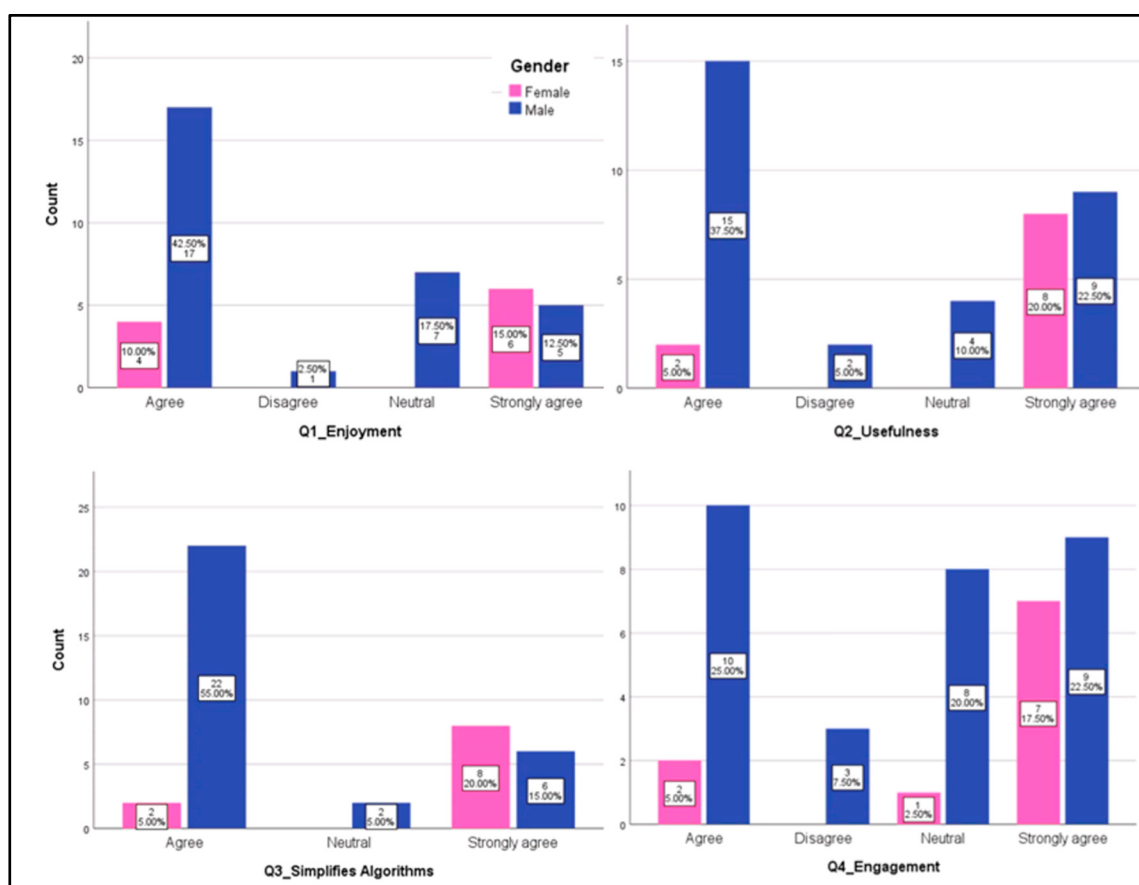


Figure 13. Students' responses to game activities for both genders.

In the same course, different teaching techniques were used to teach linear data structures such as Stack and Queue. The class camera was used to explain those two data structures. The instructor used real-life objects such as fruit and small shelves to simulate memory locations and how those two data structures are working. Students enjoyed this other technique and found it useful. However, the class activities for the three sorting algorithms where students participated themselves were preferred in students' responses, as shown in Figure 14.

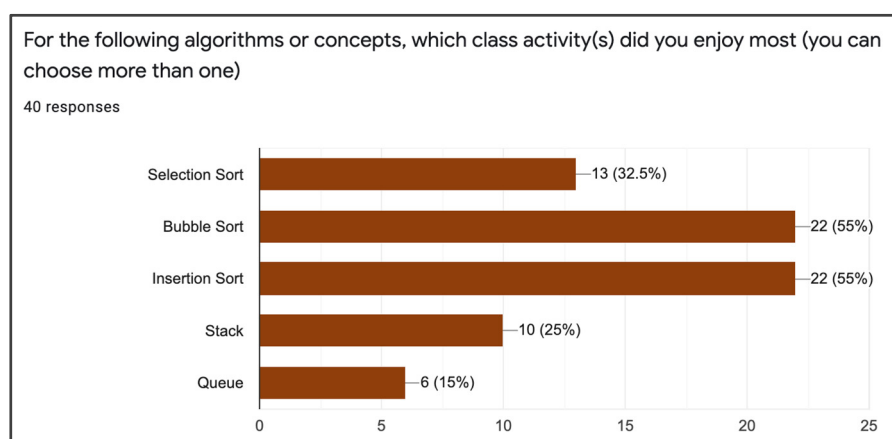


Figure 14. Students' enjoyment of different activities.

3.2. The Android Mobile Development Course

The actual number of students who attended the Android mobile development course was 21 students, but the data were collected from 17 who attended the last class. The majority of the participants were White with one Asian student. At the beginning of the course, there was one African American student, but he left the course in the second meeting. Most of the participants were male students. The number of female students who were enrolled in the course was only two (12%), and this indicates the low interest of women in the mobile development field and in computing majors in general. Adding students who enrolled in the course but did not fill in the surveys, the female percentage drops to 10%. Figure 15 shows the demographic data of the participants in the Android course.

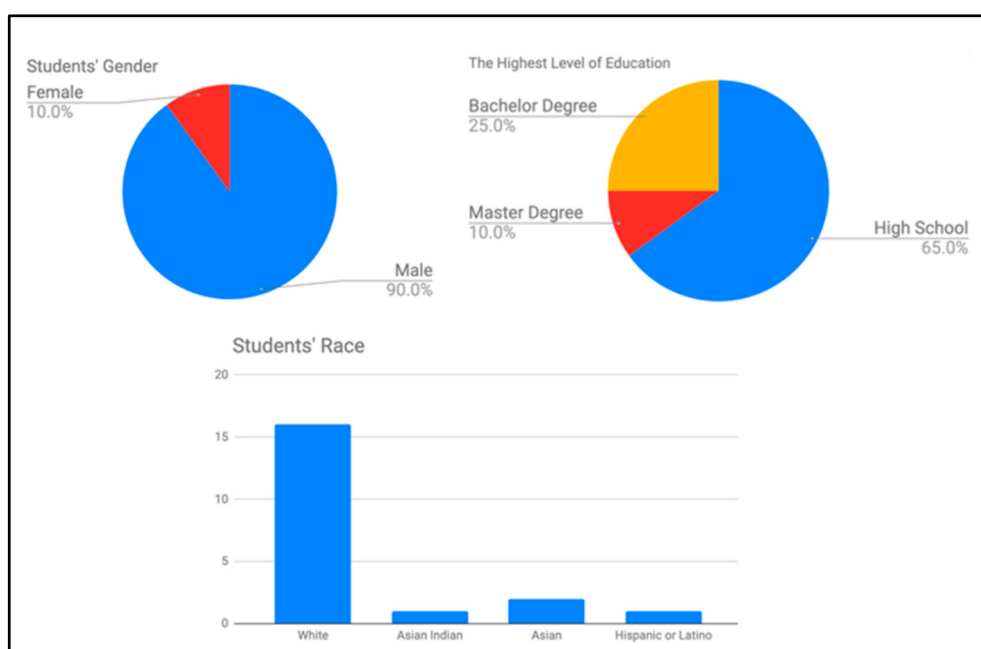


Figure 15. The demographic data of the participants.

Most of the students (76.5%) agreed that mobile development increased their interest in computing degrees, as shown in Table 2. Figure 16 shows students' responses to some questions of the survey. Most of the students (76.4%) also agreed that mobile development increased their motivation toward programming in general. The agreement percentage that android development increases creativity was high (88.3%). More than half of the students (64.7%) agreed that mobile development gives them a sense of accomplishment and makes them feel happy. 82.4% of the students found mobile development easy to learn. The researchers claim that enjoyment of the game examples and the use of the art and animation examples simplified the course materials.

Table 2. Students' responses to the survey questionnaire for the Mobile App Development course.

	Agreement %	Neutral %	Disagreement %
Enjoyment	64.7%	35.3%	0%
Creativity	88.3%	11.8%	0%
Simplicity	82.4%	11.8%	5.8%
Interest in Programming	76.4%	23.5%	0%
Interest in CS major	76.5%	23.5%	0%

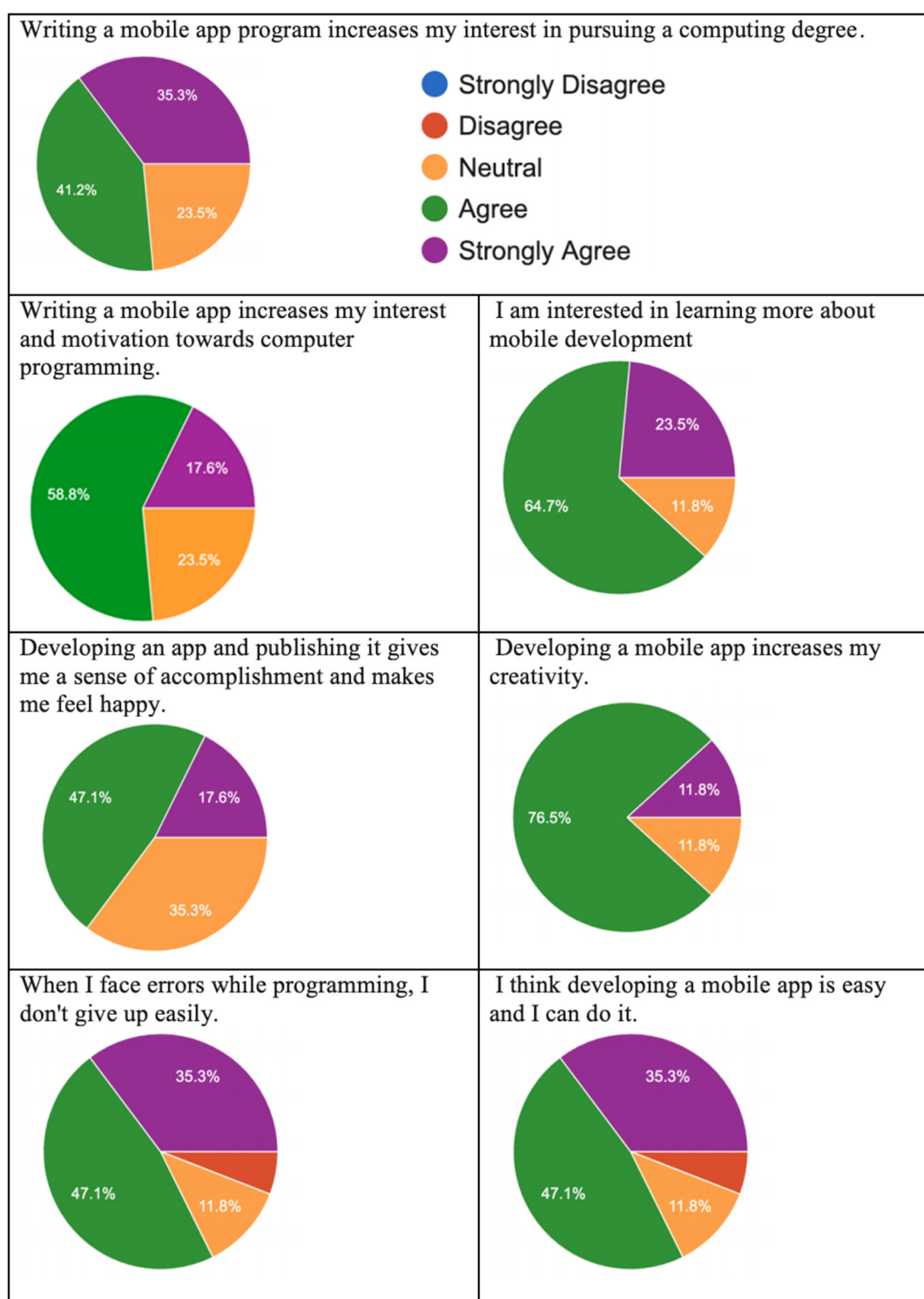


Figure 16. Students' responses to part of the mobile app development course survey.

There are several types of apps that are currently available on Google Play and people use in their daily life. These vary from social media and other communication apps such as emails to gaming apps, educational apps, banking, and business apps, news and magazine apps, weather and map apps, and productivity apps where a user can create something and save it. Statistics show that "Game" is the most popular app category at the Apple store while "Education" is the top category at Google Play [24,25]. The results of this study

show that most students were interested in creating games, educational, and productivity apps, as shown in Figure 17.

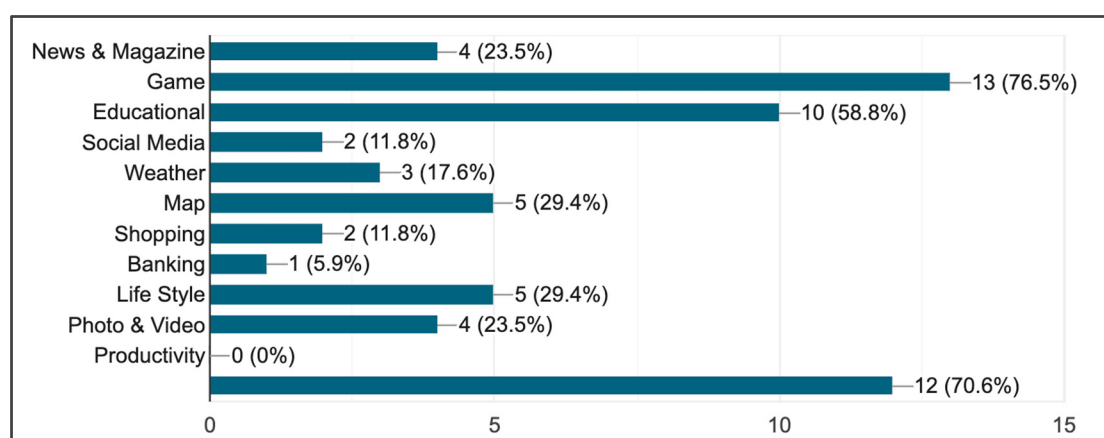


Figure 17. Students' preference to the development mobile app categories.

4. Discussion

According to motivation theory and self-determination theory, motivation is either intrinsic or extrinsic. Intrinsic motivation is about a person's enjoyment, accepting challenges, seeking mastery and curiosity about a certain thing. Extrinsic motivation promotes external reasons for participation, such as competition or collaboration with others, rewards such as scores, points, or badges [26,27]. The results of this study showed that using class activities and gaming code examples to explain computer science (CS) subjects increased students' intrinsic and extrinsic motivation. These findings concur with those of other researchers. Zainuddin (2018) compared the students' performance and motivation for both gamified and non-gamified instructional models and he found that the gamified model increased students' motivation and engagement and the findings were supported by the self-determination theory of human motivation [28]. Ibanez, Di-Serio & Kloos (2014) designed and developed a platform called Q-Learning-G to investigate the effectiveness of gamified activities in learning the C programming language. The results showed that most students continue working even after achieving the learning goal and after earning the maximum points which was an indicator of their cognitive engagement [3]. However, gamification in education could not be useful if it was not designed carefully. Hanus and Fox (2015) conducted a study to assess the effectiveness of gamification elements in the classroom using motivational, psychological, and behavioural measures [29]. Testing was carried out on students during a 16-week semester with gamification and non-gamification curriculum. Results found that students in the gamified course showed less motivation, satisfaction, and empowerment over time than those in the non-gamified course. This suggests that some care should be taken when applying certain gamification mechanics to educational settings.

The motivation theory and the self-determination theory also helped in designing the Code Genie tool. After reviewing these theories, three buttons were added to the tool: "Share", "Like", and "Others Code". These buttons would motivate students to write and share code [10]. In addition, points, and badges were added to the two tools that were demonstrated in this paper. Limantara, Hidayanto, and Prabowo (2019) reviewed studies published between 2014 and 2018 to examine the frequency and effects of gamification elements used in the recent literature [1]. They found that the most effective elements in gamified learning are points, badges, and a leader board. The Code Genie web IDE has the point and badges; however, the leader board could be added in the next update.

The results of these experiments clearly indicate a positive relationship between the performed methods and a better CS education experience. With the limited number of

female participants, it is difficult to conclude that females find gamifying CS education more useful than males do, even though the results show 100% of females agreeing to this statement. This high percentage could be due to the lack of female participation. As a minority group or even a CS novice, activities with high inclusion (such as the bubble sorting activity) may have contributed to the overall positive reaction. A sense of inclusion and belonging most probably led to their overall engagement and understanding. Out of the algorithmic activities, the students seemed to have preferred the sorting class activities the most. This could be possibly due to the level of immersion as well as the level of simplification and application to real-world examples that are easy to follow. Integrating animation, gaming, and another visual stimulation widens the CS education audience as well as the pool of students that can enjoy and understand the concepts explained. Adding visual representation to technical processes such as sorting loops and algorithms allows a broader scope of students to comprehend such complex topics.

The freedom to navigate creatively through Android Studio (with instructions) may have also contributed to the students' positive reaction to the course. Being able to creatively apply learned concepts and seeing clear results (working code) is a strong motivator for students. This also gives them the ability to further explore programming beyond the class requirements. Instant gratification (working code) is a very strong motivator and is likely to increase the likelihood of a student enjoying a course or activity.

Some misconceptions about CS and coding may be a barrier between students and pursuing a degree in a CS subject, introducing programming in an interactive and creative way can negate such assumptions and attract a larger variety of students to the field. Including artistic features in technical subjects can offer them a new perspective, therefore appealing to a subset of the overall student population that was hesitant to enter the field.

Representation and feeling included are important components of education. The methods used in the data structures class embody these concepts. The students participating in the exercises more than likely felt a sense of inclusion while participating which can increase interest and confidence in a subject. The low number of African American students in the field (and the courses) may be due to the lack of inclusion and participation. Methods such as the ones mentioned in this study can help bridge that gap and make room for a more diverse student CS population.

While the number of students and level of diversity may have been limited, the overall consensus is that introducing gaming and more artistic methods into CS teaching creates a positive reaction from most students of all groups. Perhaps further research to tackle specific portions of the student population is needed for more specific conclusions.

5. Conclusions

In this study, class activities were used to teach data structures algorithms, while game coding examples were used to teach the Android course. The students' feedback for both courses was collected at the end of the 15-week courses. Students who were surveyed after the data structures course agreed on the usefulness of the used activities and close to consensus on how such activities simplify rather complex algorithms and make them easier to understand. Similarly, students who attended the Android mobile development course stated that the use of the game coding example was useful in simplifying the materials and in making the course more interesting.

The field of computer science is dominated by males. The use of gamification could increase women's interest in this field. The results of the data structures course showed that agreement percentages of the females on different variables of the study were better than the percentages for male students. In other words, female students enjoyed the game activities and found them useful in learning data structures and algorithms. Similarly, the two women who attended the Android mobile development course were very motivated and they created two great game apps as their final projects. The researchers claim that the use of gamification could increase women's interest in programming and mobile

development. However, the number of women in this sample was too small to draw a conclusion and further study is needed for that.

This paper also demonstrates the gamification elements that were added to two tools as a third technique to employ gamification in teaching computer science. These tools are the Code Genie web IDE and the “Be Aware of the Hacker” app. No data were collected to explore the usefulness of gamifying those two tools and another further study could be conducted for that.

Our conclusion is that students’ learning of computer science algorithms and data structures can be enhanced through practical gamification that helps stimulate active learning. Additionally, the use of game examples in the Android course simplifies the topics and makes the course more enjoyable. It also increased students’ interest in the computer science field. However, further study with a bigger sample may confirm the results of this study.

Overall, a gamified approach reached a wider audience of students and increased computer science comprehension as well as enjoyment.

6. Future Work

The results of this study show that the use of game examples and game activities in the Android and data structure courses simplifies the topics and makes the courses more enjoyable and interesting. However, there are several areas of improvement that may be explored in future work. First, the sample size ($N = 40$, $N = 17$) was not large enough to produce generalizable and conclusive results, and therefore, a bigger sample with more females may better confirm the findings. Furthermore, only three data structures sorting algorithms were gamified in this study, so more activities can be developed for other sorting and searching algorithms. Additionally, virtual reality or augmented reality may be employed to improve class activities. Similarly, more game coding examples can be added to the mobile development course. To avoid any negative results as those found by Hanus and Fox [29], the gamified materials should be carefully designed.

Author Contributions: Conceptualization, H.M.J. and S.T.; methodology, H.M.J.; software, H.M.J.; validation, H.M.J. formal analysis, H.M.J. investigation, H.M.J.; resources, H.M.J. and S.T.; data curation, H.M.J.; writing—original draft preparation, H.M.J.; writing—review and editing, S.T.; visualization, H.M.J.; supervision, H.M.J.; project administration, H.M.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of Oakland University (protocol code IRB-FY2020-68, 29 April 2020).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data are not publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Limantara, N.; Hidayanto, A.N.; Prabowo, H. The elements of gamification learning in higher education: A systematic literature review. *Int. J. Mech. Eng. Technol.* **2019**, *10*, 982–991.
2. Kumar, B.; Kurtana, P. Gamification in Education—Learn Computer Programming with Fun. *IJCDS* **2012**, *2*, 46–53.
3. Ibanez, M.-B.; Di-Serio, A.; Delgado-Kloos, C. Gamification for engaging computer science students in learning activities: A case study. *IEEE Trans. Learn Technol.* **2014**, *7*, 291–301. [[CrossRef](#)]
4. Seaborn, K.; Fels, D.I. Gamification in theory and action: A survey. *Int. J. Hum. Comput. Stud.* **2015**, *74*, 14–31. [[CrossRef](#)]
5. Deterding, S.; Dixon, D.; Khaled, R.; Nacke, L. 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, 28 September 2011; pp. 9–15.
6. CodaKid. Available online: <https://codakid.com/coding-games-for-kids-the-best-way-to-teach-computer-programming/> (accessed on 22 July 2021).

7. Swift Playgrounds. Available online: <https://www.apple.com/swift/playgrounds/> (accessed on 22 February 2020).
8. Games on Scratch. Available online: <https://scratch.mit.edu/projects/144805247/editor/> (accessed on 22 July 2021).
9. Varannai, I.; Sasvári, P.L.; Urbanovics, A. The use of gamification in higher education: An empirical study. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 1–6. [CrossRef]
10. Piteira, M.; Costa, C.; Aparicio, M. A Conceptual Framework to Implement Gamification on Online Courses of Computer Programming Learning: Implementation. In Proceedings of the ICERI 2017 Proceedings, 10th International Conference of Education, Research and Innovation, Seville, Spain, 16–18 November 2017.
11. Jawad, H.M.; Tout, S. Gamifying Data Structures Algorithms. In Proceedings of the 17th Int'l Conf on Frontiers in Education: Computer Science and Computer Engineering (FECS'21) in the 2021 World Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE'21), Las Vegas, NV, USA, 30 April 2021.
12. Lafore, R. *Data Structure Data Structures and Algorithms in Java*, 2nd ed.; Sams Publishing: Indianapolis, IN, USA, 2003.
13. Jawad, H.; Tout, S.; Abualkibash, M.; Xie, Y. Integrating Art and Animation in Teaching Computer Programming for High School Students Experimental Study. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018.
14. Jawad, H.M.; de Laski-Smith, D.; Tout, S. The Code Genie Programming Environment. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018.
15. Jawad, H.M.; Tout, S. Exploring the Impact of Using Code Genie to Enhance the Programming Knowledge of Students and Across Genders: Experimental Study. In Proceedings of the Future Technologies Conference, Cham, Switzerland, 21 October 2020; pp. 916–943.
16. Jawad, H.M. Gamifying the Code Genie Programming Tool. In Proceedings of the 2019 IEEE International Conference on Electro/Information Technology (EIT), Brookings, SD, USA, 16–18 May 2019; pp. 555–559.
17. “The Biggest Data Breaches in the Middle East 2020,” Cio.com. Available online: <https://www.cio.com/article/3410237/the-biggest-data-breaches-in-the-middle-east.html> (accessed on 7 September 2020).
18. Aboul-Enein, S. Cybersecurity Challenges in the Middle East. *GCSP* **2017**, *17*, 5–49.
19. Jawad, H.M.; Tout, S. Introducing a Mobile App to Increase Cybersecurity Awareness in MENA. In Proceedings of the IEEE 2020 International Conference on Signal Processing and Information Security (ICSPIS), Dubai, United Arab Emirates, 25–26 November 2020.
20. “About Us—Quizizz” Quizizz.com. Available online: <https://quizizz.com/about> (accessed on 7 September 2020).
21. Chaudhary, D. Learning Platform Quizizz Raises \$3 Mln from Nexus Venture Partners. *Fortune India*. Available online: <https://www.fortuneindia.com/venture/learning-platform-quizizz-raises-3-mln-from-nexus-venture-partners/102262> (accessed on 7 September 2020).
22. “Kaspersky Middle East” Kaspersky.com. Available online: <https://me.kaspersky.com/about> (accessed on 7 September 2020).
23. “The Arabic News Gate for Information Technology” Aitnews.com. Available online: <https://aitnews.com/about-us/> (accessed on 7 September 2020).
24. “Apple: Most Popular App Store Categories 2018 | Statista”, Statista.com. Available online: <https://www.statista.com/statistics/270291/popular-categories-in-the-app-store/> (accessed on 22 February 2019).
25. “Google Play Most Popular App Categories 2018 | Statistic”, Statista.com. Available online: <https://www.statista.com/statistics/279286/google-play-android-app-categories/> (accessed on 22 February 2019).
26. Ryan, R.; Deci, E. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemp. Educ. Psychol.* **2000**, *25*, 54–67. [CrossRef]
27. Ryan, R.M.; Huta, V.; Deci, E.L. Living well: A self-determination theory perspective on eudaimonia. *J. Happiness Stud.* **2008**, *9*, 139–170. [CrossRef]
28. Zainuddin, Z. Students’ learning performance and perceived motivation in gamified flipped-class instruction. *Comput. Educ.* **2018**, *126*, 75–88. [CrossRef]
29. Hanus, M.D.; Fox, J. Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Comput. Educ.* **2015**, *80*, 152–161. [CrossRef]