Incorporating Gamification Elements in Interactive Programming Workshops

Anh Tang, Anca Doloc-Mihu, Cindy Robertson
Georgia Gwinnett College
United States
atang1@ggc.edu, adolocmihu@ggc.edu, crobertson2@ggc.edu

Abstract: The main objective of this qualitative study was to gauge the effectiveness of introducing algorithmic thinking and fundamental programming concepts such as variables, loops, and conditional statements to K-12 and non-IT undergraduate students through interactive classroom workshops. Using a game as the teaching platform, the classroom workshops were designed to approach programming concepts in a fun and engaging way in order to foster interest and increase participation in computing and technology, especially among underrepresented groups in the field. We strove to break down the stigma that programming was dull and boring, and that only people with innate ability could pick up the skill.

Keywords: interactive classroom workshops, gamification, block-based programming, introduction to programming concepts, outreach.

Introduction

The Technology Ambassador Program (TAP) is a service-learning course at Georgia Gwinnett College (GGC) whose mission is to boost student interest and broaden participation in computing. This is achieved through numerous student-led outreach activities both on and off-campus, including interactive classroom workshops in introductory Information Technology (IT) courses offered at GGC and outreach events for K-12 students such as the Atlanta Science Festival.

This project was developed and brought to life when we were enrolling in the TAP course. It was designed to align with the values and commitments of the TAP Program and was built upon the results presented in (Dekhane et al. 2018). These publications emphasized the positive effect our service-learning courses had on students from underrepresented minorities in the field of IT, which was encouraging them to explore more opportunities to engage in computing and technology. This project aimed to introduce essential computing concepts such as variables, loops, and conditional statements to an audience with little to no prior experience in programming, including K-12 and non-IT undergraduate students. Furthermore, we wanted to study the impact of engaging students from underrepresented groups by incorporating gamification elements in the teaching of programming concepts.

We hoped to accomplish our goal by approaching these programming concepts through a hands-on, beginner-friendly platform: Scratch, a space-themed role-playing game. By incorporating gamification elements into the learning process, we wanted to showcase the creativity side of programming as well as make the experience enjoyable and engaging for our workshop participants.

Methodology

Our project aimed to give a crash course on programming concepts such as variables, conditional statements, and loops, as well as how to use them to develop an algorithm. The project utilized two technologies: Scratch (MIT 2020) for developing the game and teaching programming, and Makey-Makey (MIT 2024) for making the workshops more fun and engaging, as well as showcasing the creativity of technology.

We chose Scratch (MIT 2020) because it had a drag-and-drop interface that allowed the workshop participants to drag premade blocks of code to the workspace and arrange them to build an algorithm. This simplified the learning process as the workshop participants could focus on understanding the concepts first without worrying about the syntax, which we considered to be secondary in this context.

Makey Makey (MIT 2024) was an invention kit that turned conductive items like Play-Doh and bananas

into a controller in place of the arrow keys on the keyboard. It is a fun and simple way to demonstrate the flexibility and creativity of technology. Tech Dungeon: Roguelite was an asset pack (Trevor Pupkin 2024) that we used to create the game visuals.

Our Game Project

For our project, we created a Scratch game called *Space Mechanic*. The player would role-play as a mechanic working on a spaceship; and each of the in-game tasks like fixing the computer's screen or opening the treasure chest corresponded to a programming question. The player could move around using the arrow keys or the Makey Makey. To make the game more immersive and engaging, we also incorporated features like wall collision (the character could not walk through physical objects), a scoreboard, and a reward system with celebration animation every time the player answered a question correctly. A more detailed description of the game can be found in (Tang et al. 2024) and on the TAP website (https://tapggc.org). We presented our project at several outreach events, including classroom workshops, which we described next.

Outreach Workshops

Our hour-long workshop comprised three sections:

- First, participants were introduced to programming concepts such as variables, conditional statements, and loops, and were encouraged to experiment with different methods of programming character movement in Scratch implementing what they just learned.
- Next, participants learned how to create their own sprite animations in Scratch using loops.
- Finally, participants entered the game with their newly equipped knowledge. By solving in-game programming questions, participants had the chance to reinforce the concepts they previously learned and put their understanding to the test.

The next section provided the data obtained from the pre and post-surveys we administered to our participants.

Results

We conducted a total of four classroom workshops in our introductory Information Technology courses at Georgia Gwinnett College. To collect data, we administered pre and post-surveys to our participants before and after each workshop. Then, we analyzed the data obtained from the same participants.

In total, we had 56 workshop participants; however, only 37 of them filled out both pre and post-surveys. Therefore, we only included the results obtained from these 37 participants. One of the categories of data we were interested in collecting and analyzing was the different demographic categories among the workshop participants.

Demographics

As per the results, 67.6% of the workshop participants were female (25) and 32.4% were male (12); there were no other gender categories of participants reported in our surveys. The majority of the workshop participants were non-IT STEM majors (43% or 16) and non-STEM majors (41% or 15). In both categories, the majority were female: 68.8% and 80%, respectively. As shown in Figure 1, we had 6 IT majors and minors participating in our workshops (16%). Among them, only 2 were female, thus making IT the only category in which there were more male participants than female (66.7%).

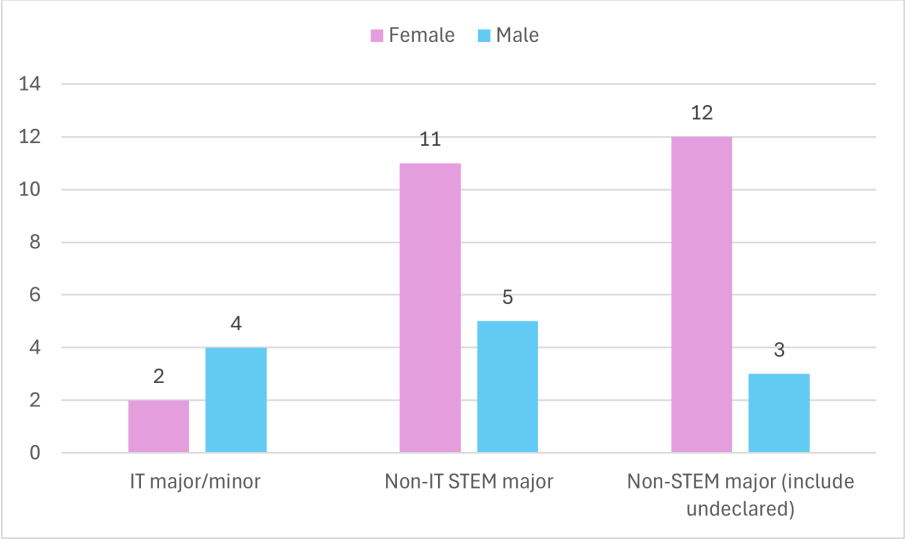

Figure 1. Majors of the workshop participants by gender.

We wanted to gauge the effectiveness of the classroom workshops both in terms of teaching the programming concepts and in terms of boosting interest and enthusiasm in technology among the participants. So, we quizzed the participants on two types of knowledge questions: general concepts and programming questions. For the purpose of this study, we limited our scope to just the programming questions (see two examples in Figure 2), which were more indicative of the effectiveness of this teaching approach, thus making them more relevant to assessing the impact of our workshops on different demographic categories. Note that all the results on both general concept and programming questions themselves can be found in our previous work (Tang et al. 2024) for the entire pool of participants.

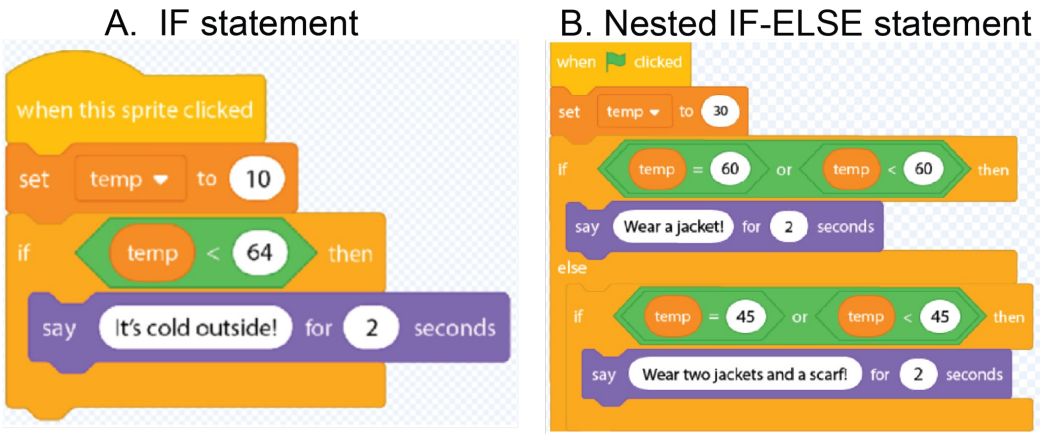

Figure 2. Examples of programming questions in Scratch that we used to test our participants' understanding of programming skills. (A) An IF statement. (B) A nested IF-ELSE statement.

Workshop Effectiveness

Female participants liked learning new technologies introduced in our workshops. In our pre-surveys, we asked the workshop participants whether they liked learning new technologies. Here we present the responses from our 25 female workshop participants. As shown in Figure 3A, only 7 of the female participants answered yes, while the remaining 17 answered maybe, and 1 female participant responded with no.

To gauge the effect that the workshop activities had on our female participants, in the post-surveys, we asked them if they enjoyed learning the technologies presented in the workshop. As shown in Figure 3B, the majority of the participants 16 answered yes, 9 answered maybe and this time, no one responded with no. This

demonstrated a shift of 8 answers (32%) from 'maybe' to 'yes' after the workshop.

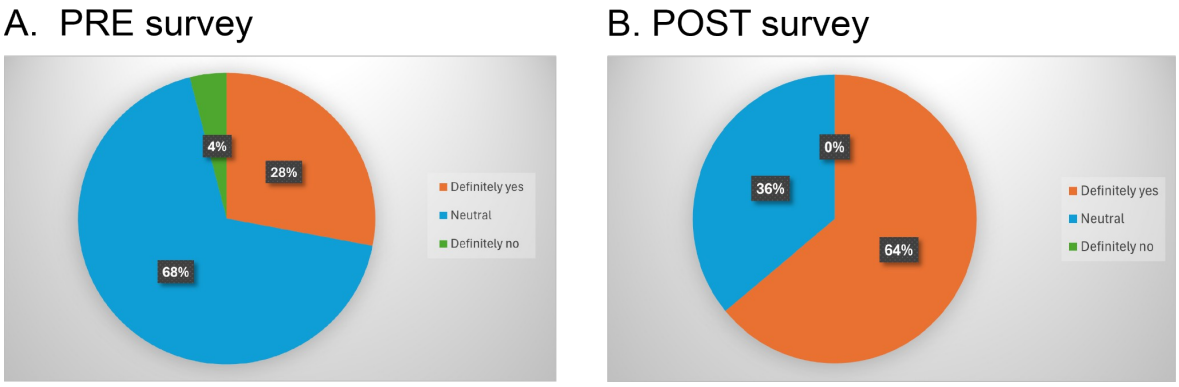

Figure 3. Female participants' interest in learning new technologies. (A) Pre-survey answers. (B) Post-survey answers.

Results for coding questions.

We asked our participants the same three programming questions in both the pre and post-surveys. The questions were designed to assess the understanding of the participants on programming concepts such as variables and conditional statements. The questions also got progressively harder, with question one being the simplest and question three being the most complex.

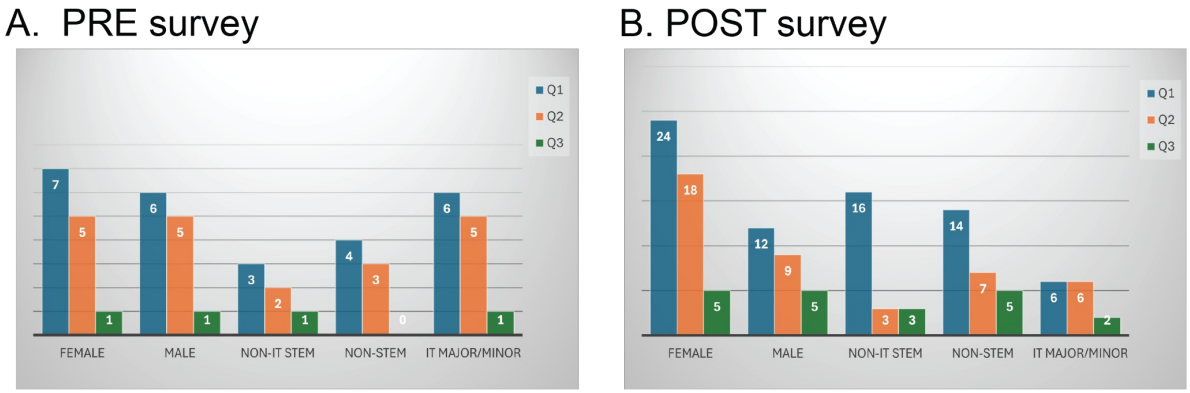

Figure 4. Number of correct answers by demographic categories for each programming question. (A) Pre-survey answers. (B) Post-survey answers.

Figure 4 summarizes the performance results by each demographic category by displaying the number of correct answers for each of the multiple-choice programming questions. The questions themselves along with the overall performance of the entire pool of participants can be found here in our previous work (Tang et al. 2024).

Overall, all demographic categories demonstrated some level of improvement across all three programming questions. We first compared the performance of the two genders.

Female participants showed a promising 68% increase in the number of correct answers in the first question, 52% in the second one, and 16% in the third. Male participants demonstrated the same improvement across all three questions, albeit less of a significant shift compared to their female counterparts, with 50% in the first question, 33.3% in the second, and also 33.3% in the third.

We also looked at the difference in performance results among non-IT STEM majors, non-STEM majors, and IT majors/minors. Among those who were non-IT STEM majors, the most significant improvement was for the first programming question (a staggering 81.3% increase), with all participants in this category answering the question correctly. However, this number quickly dropped in the next two questions, plummeting to just 6.25% and 12.5%, respectively. On the other hand, non-STEM majors showed promising results in their assessment: 66.7% increase for the first question, 26.7% in the second, and 33.3% in the third.

Last but not least, the IT majors and minors among the participants showed interesting trends: there was a

0% shift in the first question and 16.7% in the two remaining ones. Without taking the context into consideration, these numbers could be misleading. In fact, all participants who were IT majors and minors were able to answer the first programming question correctly in both the pre and post-survey, which explained the 0% shift. Both the second and third question had the same statistics, however, each of them painted a very different story: For the second question, there was a shift from 5 correct answers to 6, which meant the entire group, once again, answered the question correctly in the post-surveys. For the third question, however, the increase from 1 to 2 correct answers wasn't quite as positive.

Discussion

The main objective of the TAP Program, as an organization, is to broaden participation in computing, especially among underrepresented groups in the field. Through this study, we hope to communicate what we have found conducting programming workshops in introductory Information Technology courses at Georgia Gwinnett College, both promising results and areas that need improvement, and to demonstrate that computing skills can be taught and cultivated if given the right approach and the right environment.

Statistically, women in the tech sector have always faced numerous challenges and setbacks, despite the ongoing effort to diversify both the workforce and IT or CS post-secondary education. Many researches have been conducted in the last decades to study this issue of the gender gap and to propose solutions to tackle this ongoing challenge. Our study aims to contribute to this effort, even if only on a small scale within our institution.

From the data that we collected, on average, our female participants appeared to have less interest in learning new technologies compared to their male counterparts of the same age range. At the start of our workshop, approximately 28% of the female participants claimed to like learning new technologies (see Figure 3A). Even after taking into consideration the fact that only 8% of them were IT majors or minors (see Figure 1), this statistic paled in comparison to their male counterparts: Approximately 33% of the male participants were IT majors/minors, and according to our surveys, 58% claimed to have an interest in learning new technologies. However, this did not provide the complete picture.

As demonstrated in Figure 3B, in the post-surveys, when asked whether they had enjoyed learning the new technologies presented at the workshops, there was a significant shift in attitude among the female participants, a 36% increase in the total number of positive responses to the question, which meant more than half of them had expressed their interest and enthusiasm despite the initial reluctance or indifference. This positive result indicated the promising effectiveness that this hands-on and interactive approach in conjunction with gamification elements had on the female participants in breaking down the initial barrier and drawing out interest and engagement. More importantly, this demonstrated that, much like a skill, interest and enthusiasm could be cultivated rather than being solely an innate and permanent quality.

In addition, we also analyzed the effectiveness of our workshops in terms of teaching the programming concepts. On average, participants from all demographic categories demonstrated some level of improved understanding across all three programming questions, which tested their knowledge on concepts like variables and conditional statements. This can be observed through the increase in the number of correct answers shown in Figure 3. While the results concerning this criterion had been promising, we also noticed the discrepancy between the number of correct answers for the first and second programming question and that of the third question.

This discrepancy suggested that we should allocate more time on explaining the difference between two if statements (second programming question) and an if-else statement (third question). While the complexity of the question might have been a contributing factor to the low number of correct responses, we could not help but notice that only 2 out of the 6 IT majors/minors were able to answer the third question correctly in the post-surveys. This meant the IT major/minor category exhibited the same improvement rate (33.3%) as non-STEM majors (5 out of 15 answered correctly). It begged the question: Was the concept all that challenging or did the IT majors and minors who presumably would have more experience with programming miss what in plain sight because their brains were still wired to see the pattern in the previous question? If we accepted this hypothesis, then that could also explain the low number of correct responses overall: The participants had focused too much on the similarity between the second and third programming question and had failed to recognize the one difference. This observation certainly would be useful for us later when we expand the scope and complexity of this study, namely, by adding more levels and programming questions to the game.

Conclusion and Future Work

Overall, the results we have gathered from our study suggest that we are heading in the right direction in our commitment to encourage more participation in computing. The method of using a hands-on, interactive platform like a video game to teach programming fundamentals has proven itself to be efficient and effective. We have presented this project to K-12 teachers in our community and to 50 high school students as part of our outreach effort, who showed the same enthusiasm as our college participants.

At the same time, we are also aware of the main limitation of our study, which is the small pool of participants. Due to this limitation, we cannot claim any definitive conclusions at the moment. In the future, we plan to replicate the study to a larger sample size. Our hope is to be able to demonstrate that given the right approach, anyone can learn programming skills. Having said that, the results we have are both promising and rewarding, and we plan to continue our outreach efforts in the future.

Furthermore, we believe that our study stands to benefit from additional data such as the number of hours per week spent using electronic devices and its correlation to one's interest in learning new technologies, or the relationship between one's self-efficacy and their actual ability. We are interested in conducting more thorough research to determine other factors at play in the process of learning programming.

References

Sonal Dekhane, Xin Xu, Nannette Napier, Rahaf Barakat, Cengiz Gunay, and Kristine Nagel. (2018). Technology focused service-learning course to increase confidence and persistence in computing. J. Comput. Sci. Coll. 34, 2 (December 2018), 147–153.

Anh Tang, Julissa Valdez-Ramos, Anca Doloc-Mihu, and Cindy Robertson. (2024). Space Mechanics: A Beginner's Guide to Programming Fundamentals. In Proceedings of the 25th Annual Conference on Information Technology Education (SIGITE '24). Association for Computing Machinery, New York, NY, USA, 37–42. https://doi.org/10.1145/3686852.3687068

MIT. (2020). Scratch. https://scratch.mit.edu/

MIT. (2024). Makey Makey. https://makeymakey.com/

Trevor Pupkin. (2024). Tech Dungeon: Roguelite- Asset Pack by Pupkin. https://trevor-pupkin.itch.io/tech-dungeon-roguelite

Acknowledgements

Our team wants to express our most profound gratitude toward the IT Department Chair and the Dean of the School of Science and Technology at Georgia Gwinnett College for supporting our project. Sponsored via NSF 2315804.