Teaching Algorithms and Growing STEM with Sphero

Ammar Mehicevic
Georgia Gwinnett College
Lawrenceville, GA, U.S.A.
amehicevic@ggc.edu

Anca Doloc-Mihu
Georgia Gwinnett College
Lawrenceville, GA, U.S.A.
adolocmihu@ggc.edu

Cindy Robertson
Georgia Gwinnett College
Lawrenceville, GA, U.S.A.
crobertson2@ggc.edu

ABSTRACT

The primary goal of this project is to educate our audience about algorithms. We used Sphero, a fun and engaging robot sphere, to pique the interest of our participants with the intent of promoting STEM and Information Technology. By participating in our workshops, our audience experimented with Sphero.edu and block coding, which allowed them to learn about algorithms through easily replicable examples. This project showed our workshop participants that programming can be fun. The workshop results proved that the audience had an increased interest in programming.

CCS CONCEPTS

• **Applied computing → Interactive learning environments**.

KEYWORDS

project-based learning, teaching, block coding, critical thinking, cs, education, it, outreach, programming, online, Sphero

ACM Reference Format:
Ammar Mehicevic, Anca Doloc-Mihu, and Cindy Robertson. 2024. Teaching Algorithms and Growing STEM with Sphero. In *The 25th Annual Conference on Information Technology Education (SIGITE '24), October 10–12, 2024, El Paso, TX, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3686852.3687067

1 INTRODUCTION

The Technology Ambassador Program (TAP) at Georgia Gwinnett College (GGC) strives to break the misconception that the Information Technology (IT) field is difficult and boring by providing fun workshops for students of all backgrounds. The TAP Program students design engaging and fun outreach workshops to encourage interest in IT and STEM [3, 7].

In this project, the audience learned about algorithms through relatable examples and applied their knowledge using Sphero EDU. Sphero EDU [2] uses block coding, which is a simplistic drag-and-drop programming language that uses pre-coded blocks to create computer programs and scripts. This allowed us to teach beginner level concepts of coding such as algorithms, while-loops, if-statements, and methods. Students tested their algorithmic knowledge by programming Sphero to move through a maze via block coding.

The following sections include an introduction to our project and the results of our study designed to gauge student interest in IT after participating in our programming workshop.

2 OUR PROJECT

Using interactive pedagogy, our project teaches basic algorithmic thinking using the block coding functionality of Sphero, a script-directed robotic sphere with visual and motion capabilities. Participants learned the basic concepts of algorithms and applied that knowledge to program Sphero in a fun and engaging way.

The intent of our project is to teach the audience how to use block programming to program a Sphero robot to run through a maze. Participants learn how to program Sphero to turn and avoid obstacles within the maze with the goal of exiting the maze.

We built several mazes out of foam board cut into an L shape and two rectangular shapes with plywood chiseled into puzzle pieces to make the maze portable, each piece having a different level of difficulty as shown in Figure 1. The beginner level (L shape, Figure 1A.) is used to help participants grasp the basic concept of Sphero's movement along with getting comfortable with assembling and executing block code. After mastering the beginner level, the advanced level (Figure 1B.) tests the ability to create an algorithm to correctly reach the end of a more difficult maze without Sphero crashing or taking a wrong turn.

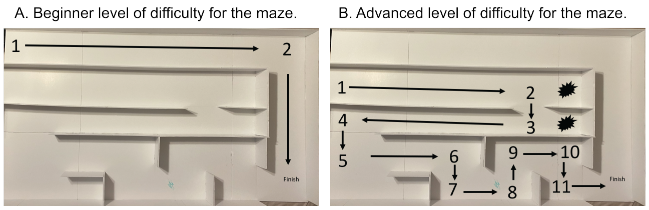
A. Beginner level of difficulty for the maze.
B. Advanced level of difficulty for the maze.

Figure 1: Participants program Sphero through a maze (A) at a simple level or (B) at a more difficult level.

Our project is freely available on the TAP Program website at https://tapggc.org/. We presented our project at several outreach events, including classroom workshops, which we describe next.

3 OUTREACH WORKSHOPS

We held three different hour-long workshops that taught the audience how to program Sphero along with basic algorithmic thinking. The workshop consisted of a lesson on the basics of algorithms and how to use algorithms to program Sphero. In the lesson, we made use of a simple task, such as creating a peanut butter and jelly (pb&j) sandwich, to make an analogy to the steps of an algorithm.

We asked the audience to give us a list of steps for how to create the sandwich and then we explained that those steps were in fact

SIGITE '24, October 10–12, 2024, El Paso, TX, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1106-0/24/10
https://doi.org/10.1145/3686852.3687067

the steps of an algorithm to create a pb&j sandwich. We then asked the audience for similar examples of algorithms and their steps. The whole session was very interactive and the audience had fun coming up with examples and adding steps to help each other with their various algorithms. The session took about 15 minutes.

Then, participants were asked to provide the steps that would navigate Sphero to go from the starting to the ending point in a maze. For this part, we used the easy maze shown in Figure 1A. In the figure, the path is marked 1-2-3. This allowed participants to brainstorm the steps of the algorithm that they would program.

Next, we introduced them to block coding and to each block code associated with the above steps. We explained each of these block codes in detail and we had them practice programming Sphero to run in a straight line to figure out the appropriate lengths to code for the various steps in their algorithm. This interactive session was about 15 minutes long.

After they successfully programmed Sphero to exit the easy maze, participants were challenged with the advanced maze shown in Figure 1B. At this time, we split them into groups of 2-3, and each group got a different maze to practice their programming skills. This session gave participants hands-on practice with block coding and allowed them to see its effects on the robot. This experience allowed our participants to gain a better understanding of coding and helped them figure out how to debug their code because they could see their mistakes in real time. Common mistakes included Sphero stopping too early or hitting the wall. We allocated around 25 minutes for this session.

We administered pre and post-surveys before and after, respectively, these workshops to assess the success of our work. The next section introduces the results of these surveys.

4 RESULTS

4.1 Demographics

We analyzed the data of the 51 participants in our three workshops who responded to both pre and post-surveys. Although we had more participants in the workshops we chose to analyze the data only from those who completed both pre and post-surveys. 21.6% of the participants were IT majors, 39.2% were non-IT but STEM majors, and 39.2% were non-STEM majors, as shown in Figure 2. In each workshop the majority of the audience was non-IT.

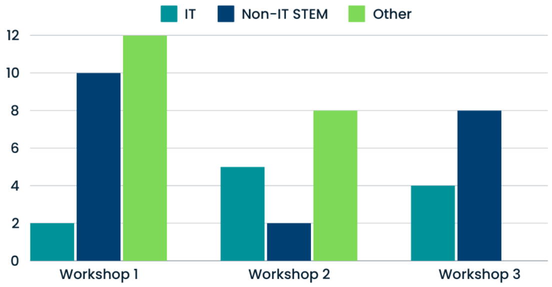

Figure 2: Workshop Participants per Major

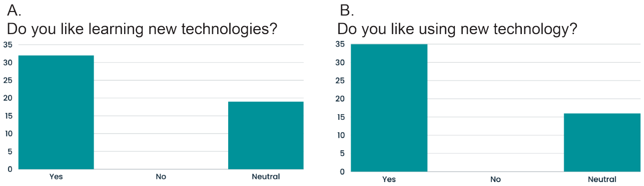

Figure 3: Results for questions related to new technology: (A) Do you like learning new technologies? (B) Do you like using new technology?

4.2 Results for Learning and Using Technology

In our pre-survey, we asked the participants whether they like learning new technology and whether they like using new technology, and we show their answers in Figures 3A and 3B, respectively. 62.75% of participants responded "yes" when asked if they liked learning new technology, while 37.25% responded "neutral" and 0% responded "no". 68.63% of participants responded "yes" that they like to use new technology, while 31.37% responded "neutral" and 0% responded "no".

We were also interested in finding out how often our participants use technology. 98.04% of the participants also reported that they use technology 7 days a week. 1.96% reported they use technology 5 days a week. None of the participants reported anything less.

96.08% of the participants reported that they spent at least 3 hours a day using some sort of technology. While 3.92% reported using technology for only 2 hours a day. None of the participants reported anything less.

These results indicate that most of the participants use some sort of technology almost every day. While that is true, only 62% enjoy learning new technology. Our workshop teaches them about a new technology with the hopes of improving those numbers.

4.3 Results for General Skills Questions

We asked participants to answer the following question: "What is an algorithm?" and select the correct answer from several choices, "A type of language", "A set of instructions or steps to solve a problem or perform a task", "The robot we are programming", or "I don't know". The results are shown in Figure 4. A huge increase in the number of correct answers can be seen for the correct answer, "a set of instructions", with 35.3% correct answers in the pre-survey and 96.1% in the post-survey (a 60.8% difference). However, a few participants 3.9% (2) still gave the wrong answer in the post-survey, so not everyone was able to understand this concept.

We used the χ^2 test [1] to analyze the difference between our control and experimental data results to see if there was any significant difference between these group results. To test the independence of the two survey results, pre and post, we used the chi-square test of independence. The p-values were obtained using $scipy.stats.chi2\_contingency$ function in SciPy library [8] in Jupyther Colab. We obtained $\chi^2 = 45.91$, $df = 3$, and a p-value $p = 5.907183694589282e - 10$ which is less than ($p < 0.05$) showing that there was a significant difference between the answers of the pre-survey versus the answers of the post-surveys.

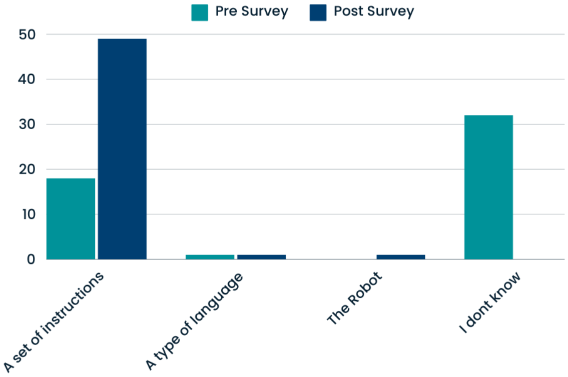

Figure 4: Results for the question: What is an algorithm?

Another question we asked, "What is block coding?", had the following 5 options to choose from: "A type of drag and drop coding", "Coding with your mind", "A type of copy and paste coding", "A type of C++ coding", or "I don't know". The results are shown in Figure 5. Again, a huge increase in the number of correct answers can be seen for the correct answer, "drag-and-drop coding", with 21.6% correct answers in the pre-survey and 84.3% in post-survey (a 62.7% difference). We can also see that everyone gave an answer to this question in the post-survey, but there were a few 15.9% (8) who did not answer this question correctly. We obtained $\chi^2 = 53.023$, $df = 4$, and a p-value $p = 8.42455008518723e - 11$ which is less than ($p < 0.05$) showing that there was a significant difference between the answers of the pre-survey versus the answers of the post-surveys.

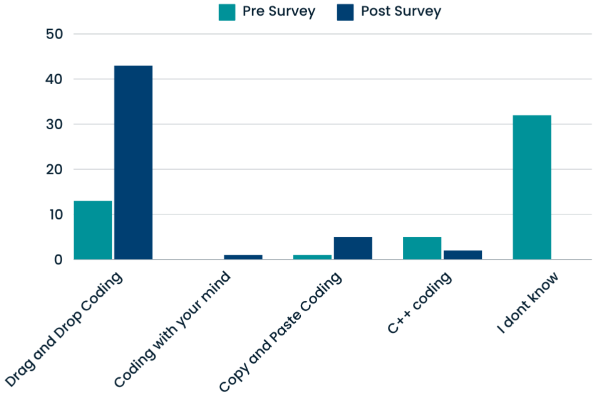

Figure 5: Results for the question: What is a block coding?

4.4 Results for Coding Skills Question

We asked the block coding programming question shown in Figure 6 to assess the level of coding our audience learned during the workshop. 98.03% of the participants answered the question correctly while 1.97% answered incorrectly (1 participant). This is a great, unexpected result for our study.

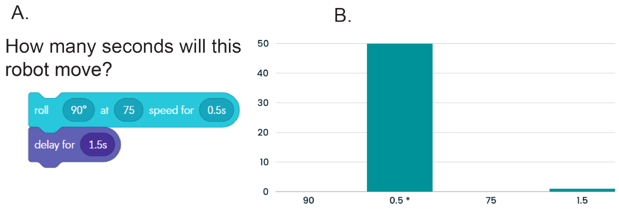

Figure 6: Block coding question about the delay of the sprite movement. (*) Shows the correct answer. (A) Block coding corresponding to the question. (B) Results of the post-survey.

4.5 Results for Workshop Engagement and Difficulty Questions

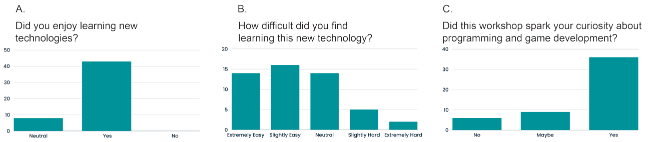

Figure 7: Results for questions related to the workshop: (A) Did you enjoy learning new technologies? (B) How difficult did you find learning this new technology? (C) Did this workshop spark your curiosity about programming and game development?

We asked our participants if they enjoyed learning the new technology. The results from the post-survey are shown in Figure 7A. 84.3% responded "yes", and 15.7% responded "neutral". There was no negative answer.

We asked our participants how difficult it was for them to learn this new technology. The results from the post-survey are shown in Figure 7B. 27.5% responded "Extremely Easy", 31.4% responded "Slightly Easy", 27.5% responded "Neutral", 9.8% responded "Slightly Hard", and 3.8% responded "Extremely Hard". That is, 58.9% found learning this technology somewhat easy, while only 13.6% found it somewhat hard.

Participants were also asked if the workshop sparked any interest or curiosity about programming and game development. The results from the post-survey are shown in Figure 7C. 11.8% responded "No", 17.7% responded "Maybe", and 70.5% responded "Yes".

The workshop participants ranked the difficulty of the workshop a 3.58 out of 5. Participants were asked how fun the workshop was and they ranked it 4.67 out of 5. Participants gave a 4.74 out of 5 rating for how engaging the workshop was to them. The participants were then asked to rate the enthusiasm of the presenter and they gave a 9.25 out of 10. Finally, the participants were asked to rate the overall workshop which was rated a 4.91 out of 5. Our conclusion is that we received very high scores which tells us that we did well but there is also room for improvement in our project and the teaching techniques we used.

We asked the participants to tell us what was the most difficult thing to learn/understand in our workshop and the most common response was how to figure out the timing required by the robot for it to make each turn throughout the maze without crashing or stopping earlier than expected. This is not a programming issue, but rather an issue related to the technology we used. Sphero tends to be inconsistent due to its battery life and the strength of its Bluetooth connection.

5 DISCUSSION

In this work, we achieved several goals. We successfully created a project from the beginning to the end that aimed to teach basic programming and algorithmic thinking in a fun and engaging way using a robot called Sphero. We held 3 workshops in which we taught our audience this technology as well as basic programming skills. We also gave several outreach presentations/demos of our project to K-12 audiences at several events such as the Atlanta Science Festival, a K-12 event at GGC, and at a local elementary school. Based on our participant responses, our outreach events were a great success. Our participants were engaged while programming Sphero and asking a lot of questions. Several of the participants even asked how to acquire a Sphero of their own so that they could program at home.

Our workshops were held in general education IT classes at our undergraduates only college, where the majority of the population is non-IT. Since these classes are prerequisites for our beginning programming classes, most of the IT students who participated in our workshops were either new to programming or may have learned a little bit of programming in high school. 63% had no previous programming experience, 24% reported themselves as newbies to programming, and 13% said they had intermediate programming experience. This indicates that most of our participants got their first programming lesson during our workshop.

In our pre-survey, we asked the audience if they like learning new technology and most of them (62.75%) said yes and the rest (37.25%) were neutral. In the post-survey, we asked them a similar question but specific to the technology that we used. Most of them (84.3%) said that they liked learning how to program Sphero, and the rest (15.7%) were neutral. Although the two questions were slightly different, we think they are comparable. This shows that 21.55% of participants shifted from "neutral" to "yes" after our workshop, indicating that peoples' opinions of new technology can improve if the technology is introduced to them in an engaging way as we did in our workshop.

This work continues previous work [6, 7] on finding the best strategies to introduce programming skills to audiences. Unlike [5], we taught block coding by using a different tool, the Sphero robot, which allows participants to see the effect of their coding blocks via ball movements through a maze. This is a hands-on-only robot although it is programmable through an interface with block codes similar to the one used by Scratch [4]. While the participants wanted to continue to play with the robot after being introduced to Sphero, the intent of this workshop was to teach programming skills. So, we had to limit the amount of time the participants could freely play with Sphero to 15 minutes to be able to include the 30-minute programming instruction session within the hour-long workshop. This gave us less than one hour to learn and practice the coding part of the project, and thus our coding questions may need to be a little less challenging.

Our decision to teach algorithmic thinking in this way proved to be very successful, with more than 60% improvements in the number of correct answers between our pre and post-surveys. These results have provided a good starting point for future studies where we can start to teach more difficult concepts.

Like [5, 7], we asked coding questions similar to the concepts and structures participants experienced during the workshop. Unfortunately, we did not ask this question in the pre-survey to compare the number of correct answers before and after the workshop. This is something we can easily correct for our next workshops. Also, it would be interesting to incorporate more challenging questions in the post-survey. In addition, it would be interesting to introduce a follow-up workshop that would provide a more difficult maze for the participants to navigate through, thereby allowing us to introduce additional coding concepts. These more challenging concepts could include algorithms with more steps, ifs and whiles, and other statements.

We received a very good overall score for the workshop getting a 4.91 out of 5. The fun factor was ranked 4.67 out of 5 and the engagement was ranked 4.74 out of 5. Those are very high scores that surpassed our expectations. They also showed that we achieved our goal of teaching the audience programming skills in a fun and engaging way. This is very encouraging to us for a first trial of such an outreach endeavor. These numerical results are supported by several of the general comments that participants provided including:

- "Fun little introduction to programming. Had lots of fun would do it again."
- "Everything was great, and explained very well and it was really fun."
- "You guys did awesome."
- "fun :)"

In addition to the positive comments just listed, we were also provided with some suggestions on how to improve moving forward with our project:

- "I wish we had 3 mazes so we didn't have to wait so long between runs."
- "Have more projects!"
- "I would like more obstacles/criteria."
- "how to make ball say gregory."
- "Race other players through the maze."
- "Help out with the timing more."

We plan to incorporate this feedback in the next version of our project.

6 CONCLUSION AND FUTURE WORK

In this work, we successfully developed a fun and engaging project that introduces audiences to basic programming skills via a hands-on robot that is programmable using block coding. We held several outreach events for K-12 and college students and their feedback was highly positive. This showed that our project was very versatile with minimum changes needed to be used for teaching new technology to audiences with different levels of technology knowledge, and specifically programming knowledge. Our results indicate that our workshop is headed down the right path but there are some improvements that we plan to implement in the future. We also plan to do more workshops for K-12 audiences as well as college classes.

ACKNOWLEDGMENTS

Our team wants to express our most profound gratitude to the IT Department Chair, the Science and Technology Dean at Georgia Gwinnett College, and the STARS Computing Corps for supporting our project. Supported by NSF grant BPC-DP 2315804.

REFERENCES

[1] William G Cochran. 1952. The χ^2 Test of Goodness of Fit. *The Annals of Mathematical Statistics* 23, 3 (1952), 315 – 345. https://doi.org/10.1214/aoms/1177729380

[2] Sphero EDU. 2024. *Sphero EDU.* https://edu.sphero.com/sphero/home

[3] Sonal Dekhane et al. 2018. *Journal of Computing Sciences in Colleges* 34, 2 (2018), 147–153. https://doi.org/10.5555/3282588.3282609

[4] MIT. 2020. *Scratch.* https://scratch.mit.edu/

[5] Valentina Mosquera-Reina, Ryan Cunico, Josiah Williams, Matthew Bauer, Anca Doloc-Mihu, and Cindy Robertson. 2021. Introducing Programming Concepts through Interactive Online Workshops. In *SIGITE 2021.* ACM, 71–72. https://doi.org/10.1145/3450329.3478319

[6] Cindy Robertson and Anca Doloc-Mihu. 2021. Assessing the Effectiveness of Teaching Programming Concepts through Online Interactive Outreach Workshops. In *SIGITE 2021.* ACM, 123–128. https://doi.org/10.1145/3450329.3476861

[7] Cindy Robertson and Anca Doloc-Mihu. 2023. Understanding College Level Student Learning of Basic Programming at an Open Access Institution. In *Proceedings of the 2023 ACM Southeast Conference.* Association for Computing Machinery, New York, NY, USA, 26–32. https://doi.org/10.1145/3564746.3587007

[8] SciPy.org. [n. d.]. *SciPy.* https://projects.scipy.org/getting-started.html