CodeSwarm-AR: Tangible Programming Education with Swarm Robotics and Augmented Reality

Diwij Dev, Kai Huang, Jae Lee, Dongjie Liu, Tian Xia
University of Calgary
Calgary, AB, Canada
{diwij.dev,kai.huang1,jae.lee2,dongjie.liu,tian.xia2}@ucalgary.ca

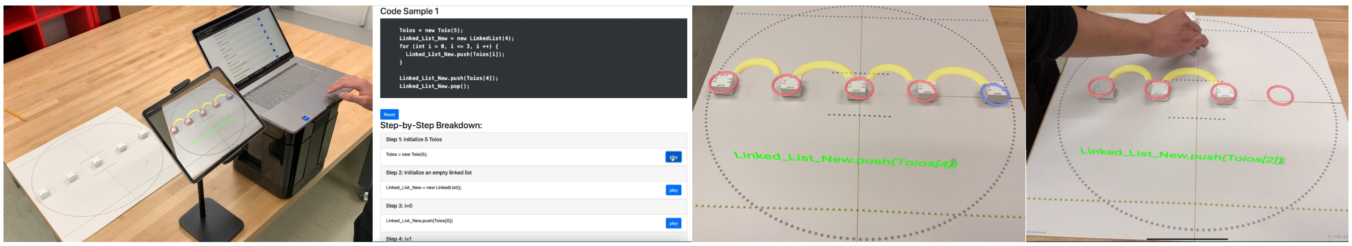

Figure 1: CodeSwarm-AR explores bi-directional interaction between a code sample shown on the webpage UI and the multiple physical robots with the help of AR to enhance the programming education experience. On the computer screen, the user sees the code alongside the step by step execution of the sample code (second image from left). Each step (or the entire process) can be played from the webpage UI and the robots will execute their corresponding behaviours. Each robot is augmented with AR, providing the user with additional information about their behaviour and the state of the code. Each robot represents a different node in a linked list, each connected via a yellow AR line once they are initialized and brought to the center. The user can interact with these robots to physically add another node in or remove a node (last image).
Demo video of the system prototype: https://youtu.be/tbPVSlrchQc

ABSTRACT

We present CodeSwarm-AR, an interactive system for programming education that utilizes swarm robotics and augmented reality (AR). While other projects have explored the use of AR and robotics for general STEM education purposes or block-based programming education, CodeSwarm-AR offers a unique approach with the use of multiple robotics, bi-directional input methods, and directly learning programming concepts like data structures. Through bi-directional code interaction, visualization, and step-by-step demonstration, CodeSwarm-AR provides users with a comprehensive learning experience. Existing code examples are demonstrated through the movements of swarm robots, allowing users to both follow and visualize the construction of a linked list data structure, as well as interact with it using robots to generate new code by manipulating the linked list. CodeSwarm-AR presents a promising new approach to programming education through the integration of swarm robotics and AR.

UIST '23, Oct 23–28, 2023, San Francisco, CA, U.S.A.

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/123456.123456

CCS CONCEPTS

• **Human-centered computing → Mixed / augmented reality**.

KEYWORDS

augmented reality; swarm robotics; programming education; robotics; tangible education

ACM Reference Format:
Diwij Dev, Kai Huang, Jae Lee, Dongjie Liu, Tian Xia. 2023. CodeSwarm-AR: Tangible Programming Education with Swarm Robotics and Augmented Reality. In *UIST '23: The ACM Symposium on User Interface Software and Technology, Oct 23–28, 2023, San Francisco, CA, U.S.A.*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/123456.123456

1 INTRODUCTION

With the widespread integration of technologies into our lives, there has been an increase of interest in programming and STEM education. Traditionally, an individual who is looking to learn programming concepts must do so using screen-based learning, via web applications (e.g. PythonTutor) [17] or online video resources (e.g. YouTube) [9]. However, with the continuous maturity of augmented reality (AR) and robot technology, they have been identified as promising tools that can promote interactive and fun learning environments. [16, 34]. AR enables the overlay of virtual environments onto real-world environments allowing individuals to bypass the physical limits of their environment while maintaining interaction with the physical reality. Robotics on the other hand, allows the physical embodiment of concepts to be mapped out in the real world, in which the user is allowed to interact with. To address this, recent research has been conducted in using AR and robotics for STEM education, often targeting to develop computational thinking in the users. [15, 16]

However, the frameworks and systems suggested remain largely limited when it comes to programming education. Even though these researches identify that computational thinking skills and familiarity to programming are important, their educational goals remain as general concepts as they are more geared towards a younger audience. It often blurs out the programming language portion of the the experience via block-based programming. This way, the proposed systems and frameworks can focus on improving one's ability to express solutions of problems as certain computational steps or algorithms, also known as computational thinking, without worrying about the syntax of programming languages [15, 19]. However, we believe that this is a large waste of opportunity, given the new powerful technologies available in education such as AR and robotics, and could help move past the traditional screen-based web-applications such as PythonTutor [17] and create a more powerful tool for programming education. Hence, we propose a system that not only builds one's computational thinking skills and familiarity to programming language formats, but also to have the skills learnt to directly translate and assist in an individual's understanding of programming language structures and concepts such as the linked list data structure.

In this research paper, we introduce a novel method for enhancing the education of programming concepts, particularly data structures, by utilizing augmented reality (AR) and swarm robotics. Our approach enables bi-directional interactions between the user and robots, allowing for a more intuitive understanding of data structures. Bi-directional interactions leverage two input methods: 1) Code-to-Robot input method where the user can select code samples to view demonstrations by the robots; 2) Robots-to-Code input method where the user can manipulate the robots to make changes to the existing code sample and generate new code based on the robots' movements. By providing a visual user interface for existing code samples, an AR overlay for observing the effect of the code on the robot (Sony Toio), and multiple Toios to represent the data structure, we aim to facilitate a more effective educational experience. This approach allows for our system to help individuals interact with specific data structures, in the same style a programming language would, improving individuals' familiarity to data structures.

We have developed a prototype that features a code sample with a linked list initialization and population through the web page interface [Figure 1-b], making five Toios visualize the population of a linked list, adding to and popping from said linked list. This behaviour is augmented with AR to visualize the current code segment and to connect the representation of a linked list structure with individual Toios.

In our current prototype, the code-to-robot input method enables step-by-step demonstration. Under the step-by-step demonstration section in the web page, the user goes through each step by clicking on the play button next to them, and at the same time, the robot performs demonstration on each incremental movement with AR objects attached to it [Figure 1-c].

As for the robots-to-code input method, we have an interaction terminal that records what each Toio is doing, detecting how the linked list is initialized, populated, and interacted with. The resulting code is shown to the user, which can be optimized using a button that adds loop structures to the generated code sample, if possible [Figure 1-d]. We believe that using Toio robots and AR technology will make the learning experience more interesting and meaningful. This paper makes the following contributions:

(1) Introduces a novel, specialized, tangible tool for programming education that focuses on teaching intangible programming concepts such as data structures and loops instead of STEM education or computational thinking skills.
(2) Propose a novel method of bi-directional (code-to-robots and robots-to-code) input method that can be utilized for better understanding of programming concepts.
(3) Propose a novel method of using AR and swarm robotics technology to visualize intangible concepts, bypassing the limits of physical space to represent a complex structure in augmented reality with coordinated robotics.

2 RELATED WORK

2.1 Robotics in Education

In prior works, the physical presence of robots have been identified as a powerful method that can increase cognitive learning gains [22] and help teach youths about science, technology, engineering, and mathematical concepts (STEM) [10, 18, 21]. Based on these research, many attempts have been employed to better use robotics in the field of education. Oftentimes, robotics in education are separated into two main domains: using social robots as a tool to increase engagement in learners [20, 29, 35] and using robots as interactive visualization tools [15, 25]. The social robots have been used to increase engagement in learning a secondary language [20], capture attention of deaf infants in order to expose them to sign language [29], teach children about artificial intelligence concepts [35], and to increase interest and engagement in children when learning about the digestive system (STEM) [26]. Unlike these researches, our proposed system fall into the latter category, as we do not employ a social robot for our programming education. For studies where robots are used as tools for visualization and interactive tools, it has been used as a game state visualization tool [25] and a tool for evaluating and visualizing optimal paths [15].

Even though the main focus of educational robots have largely remained for STEM [10, 18, 25, 35], computational thinking skills [11, 15, 35], and language skills [20, 29], the extent in which these researched systems teach about programming language structures and strategies are limited. Oftentimes, robots are used to teach individuals about ways in which a program would approach a problem, without actually showing what that would look like in a programming language structure. As opposed to these systems, our proposed system focuses on the programming language aspect, allowing individuals to explore how a programming language is structured and used when given a scenario.

2.2 Techniques in Programming Education

In recent years, there has been a growing interest in using innovative technologies such as Augmented Reality (AR) and tangible programming tools to teach programming to children and beginners. This is in response to the increasing demand for computer programming skills in various industries. Prior research [12, 19, 24, 30] proposes a visual programming environment that uses blocks to represent programming concepts or various mini-games that engages the audience in interactive, tangible programming, hands-on learning. Studies have shown that children can learn programming concepts more effectively through physical manipulation of blocks that represent different programming elements. [13, 19, 23, 31, 33]. In contrast to the studies mentioned earlier, our system offers a novel approach to programming education, called "Bi-Directional Input Method," which aims to make programming more engaging and interactive for beginners and children. Our system consists of two methods: "Code to Robot" and "Robot to Code," which are demonstrated using AR to provide a step-by-step visual guide. It offers a more accessible and enjoyable way to learn programming concepts.

Research has shown that visual programming languages are more effective than traditional text-based languages because they provide users with a more intuitive programming interface. This is especially beneficial for beginners, children, and those with limited programming experience, as a visual programming language through augmented reality (AR) can help simplify complex programming concepts and make them easier to understand[14, 15, 28, 31]. Unlike the aforementioned studies that focus on visual programming languages, our system takes a different approach by utilizing augmented reality (AR) to visualize traditional programming concepts such as linked lists. Our systems are designed to enhance a traditional programming or teaching experience by incorporating AR technology, making it more engaging and interactive for users.

2.3 AR and Robotics Assisted Education

Recent research has shed light on using AR and robotics to facilitate educational activities. Robots can be used as tangible embodiment of intangible objects, such as the movements of the solar system and atoms, to teach complex and abstract STEM concepts [27, 30]. They can also assist in art education by training learners to play piano [37] and provide learners the opportunity to co-create in choreography [36]. These educational activities are also dependent on and immensely enhanced by the immersive virtual environment [27, 30] and visual feedback [24, 27, 31, 36, 37] enabled by AR.

Similar to the aforementioned research, CodeSwarm-AR leverages robotics and AR for educational purposes, but its focuses and interaction techniques are different. Unlike prior works focusing on simplifying the processes of programming and robot control [24, 30, 31] or developing computational thinking skills [16], this project aims to teach general programming concept like data structure. Furthermore, CodeSwarm-AR enables bidirectional input and uses the robots to both demonstrate and generate code, while prior research explores the demonstration of educational materials with robots [24, 27, 30], creating versatile and immersive environment [30], supplementing paper-based educational materials [27], and facilitating choreography [36].

3 DESIGN CONCEPTS AND INTERACTION TECHNIQUES

3.1 Tangible Programming Education with AR and Swarm Robotics

CodeSwarm-AR expands the conventional on-screen programming education into a system that enables the user to physically manipulate the tangible objects that act as proxies to the code samples displayed on the screen.

To properly explain and demonstrate the programming examples, the on-screen programming education tools use geometric shapes like circles and dots with lines and arrows to present coding concepts in animated diagrams. CodeSwarm-AR leverages multiple tangible actuated objects and virtual lines to demonstrate coding concepts. A swarm of small robots are used to represent the animated dots and virtual lines displayed in AR are used to represent the animated lines.

While interacting with the robots, the user's field of view is directed away from the screen and onto the robots. To account for the loss of context, AR contents displayed in the user's field of view are attached to the swarm robots and provide explanation to the user through animated virtual objects in AR.

AR Virtual Object Display: Virtual texts are displayed at the lower centre of the Toio tracking mat in the user's AR view so that the user can focus on the robots' demonstration once a step demonstration session is started and the user's field of view moves away from the stationed screen. The robots' positions, movements, and connections are also indicated by the virtual geometric shapes attached to the robots in the AR view.

3.2 Bi-Directional Input Method

The user can experience tangible programming education through bidirectional input methods:

(1) Code-to-Robot Input Method: From the on-screen code demonstration web page, the user can select a code sample to play a step-by-step demonstration made by the robots.
(2) Robot-to-Code Input Method: The user can physically manipulate the robots to generate new code based on the movement and positions of the robots. The newly generated code is displayed in the on-screen code generation web page.

3.2.1 *Code-to-Robot Code Demonstration.*

In the code demonstration web page, the user can select a code sample to play a step-by-step demonstration. The robots act as the intangible components in the code sample and make movements to demonstrate such code samples. At the same time, the user can view annotations in AR that facilitate the robots' demonstration.

3.2.2 *Robot-to-Code Code Generation.*

When the user manipulates the robots on the tracking mat, new code is generated in the code generation web page based on the user's manipulation and the robots' resulted movements and positions. After such manipulations are made, the user can view the reflected code in the web page and the virtual objects in the AR view. The user can also choose to optimize the generated code to a more efficient format.

4 SYSTEM PROTOTYPE APPLICATION

We demonstrate the design concept of CodeSwarm-AR by implementing a system prototype leveraging an on-screen web page on a laptop, a web-based AR interface on a tablet, and Sony Toio robots with a tracking mat. With this system prototype, the user can experience tangible programming education and learn programming concepts through code demonstration and generation enabled by CodeSwarm-AR.

In this application scenario, the user learns the concept of a data structure, linked list, through bidirectional input enabled by CodeSwarm-AR.

4.1 Code-to-Robot Demonstration on Linked List

In the code demonstration web page terminal, the learner can view the assigned code sample from the instructor. In this application scenario, the following code is demonstration how a linked list is initialized, a list item is added and removed:

```
Toios = new Toio(5);
Linked_List_New = new LinkedList(4);
for (int i = 0, i <= 3, i ++) {
    Linked_List_New.push(Toios[i]);
}
Linked_List_New.push(Toios[4]);
Linked_List_New.pop();
```

This code sample is broken down into steps based on the index of code lines and the index of loop interables. In this scenario, the linked list was initialized with four list items by using a for loop to add an item four times. This for loop is broken down into four steps. The user can click on the play button next to each step to

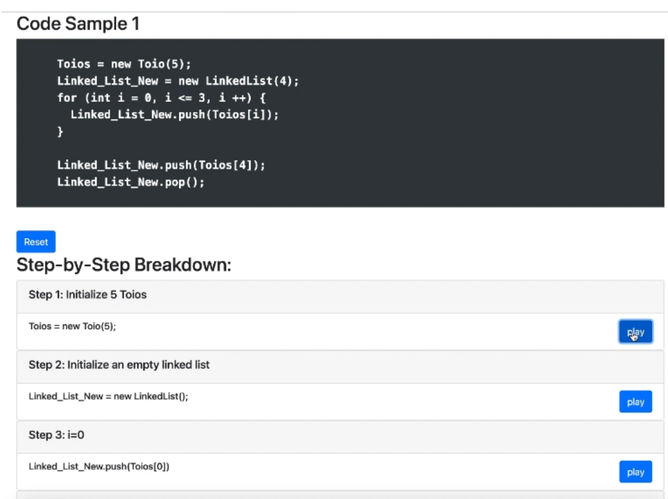

Figure 2: The code demonstration terminal in a web page with step-by-step demonstration consisting of each step and its play button.

see the demonstration from Toio robots. When Toio robots make designated movements at designated positions to demonstrate each step of the code sample, the AR display shows animated virtual objects and texts attached to Toio robots and their tracking mat to supplement the code demonstration.

The first part of the code demonstrates initializing a list of five Toio robots that will be used to initialize and operate in a linked list. Once the user clicks on the play button in the first step, five Toio robots spin at their original sideline positions indicating they have been added to the list of five Toio robots.

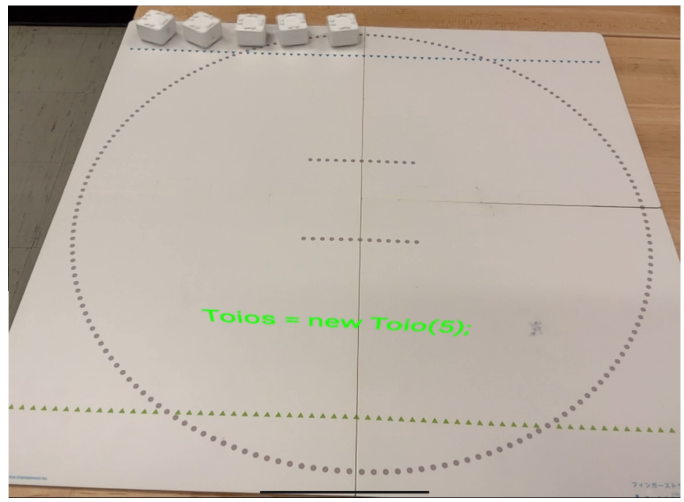

Figure 3: Toio robots spin as they are selected to be included in a list used to populated a linked list

Initializing a linked list is the second and main part of this code demonstration. The first step of linked list initialization is to make space in the memory for an empty linked list with four items. Once the user clicks on the play button in Step 2, the AR display shows four virtual red circles indicating the potential positions of the four Toio robots when they join the linked list. The initial linked list consists of four Toio robots and the initialization is completed by using a for loop to repeat adding a Toio robot four times, which are shown in the code demonstration web page. When the user clicks on the play button next to each step, the Toio robot moves to the central part of the tracking mat. Each of them is placed at a designated location so that after these steps were played, the four Toio robots form a horizontal line leaving equal space between the robots and their neighbours. After each Toio robot moves to its designated position, a virtual line shows in the AR display and connect the two Toio robots and their virtual location circles.

The last part of the code sample is to add and remove an item from the linked list. To add an item to the linked list, once the user clicks on the play button next to Step 7, the AR display shows a blinking empty spot for the fifth Toio robot at the end of the linked list, and the fifth Toio robot moves to its designated position while other four Toio robots move left to make room for it. At the end of this step, the horizontal line of Toio robots has equal space between the robots next to one another, and the AR display shows a new link between the newly added Toio robot and its neighbour.

The last step demonstrates removing an item from the linked list. When the user plays Step 8, the last Toio robot in the linked

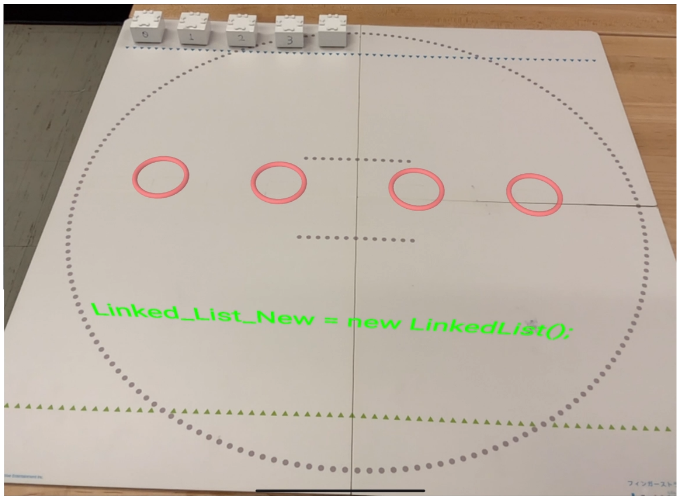

Figure 4: AR display shows blinking red circles that indicate the positions of the Toio robots in the linked list.

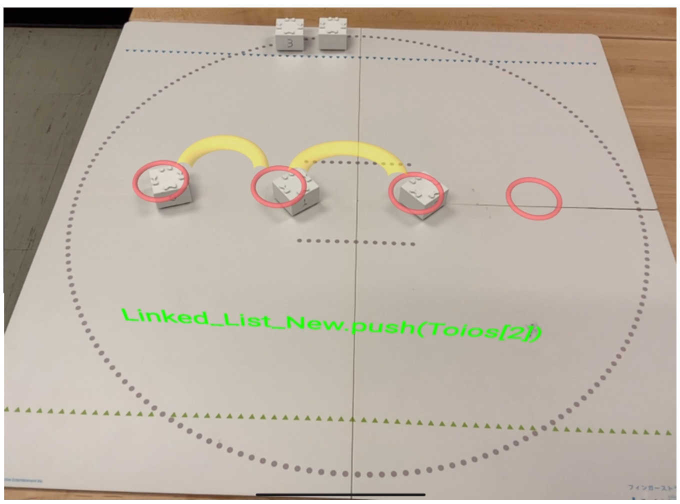

Figure 5: Once a Toio robot moves into the linked list, a virtual line is formed in AR display to connect the two Toio robots next to one another.

list moves back to the sideline. At the same time, its virtual location highlight disappears from the AR display, as well as its virtual link with the Toio robot next to it.

4.2 Robot-to-Code Generation on Linked List

The second part of this application scenario is to demonstrate robot-to-code code generation. The user physically manipulates the Toio robots on their tracking mat, and the code is generated based on the movements and positions of the Toio robots. The user can view the generated code in the code generation web page and has the option to optimize the code displayed in the terminal.

The user first taps on the five Toio robots on the sideline to initialize them as a list of Toio robots. The code generation terminal

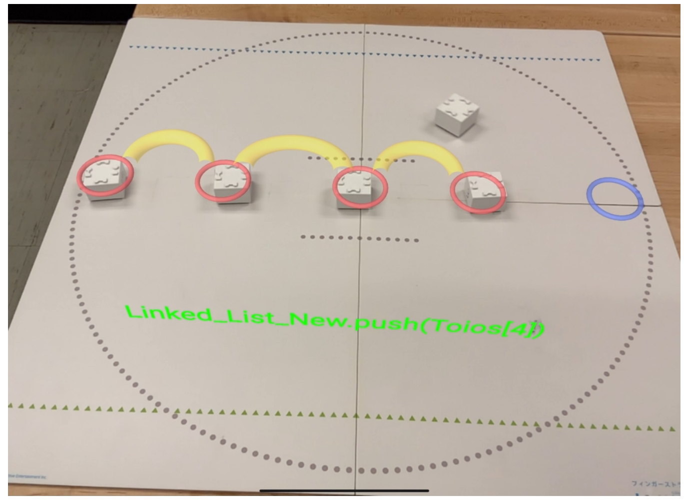

Figure 6: A blinking circle is shown in the AR display indicating the added Toio robot's position, while the fifth Toio robot moves to its designated position.

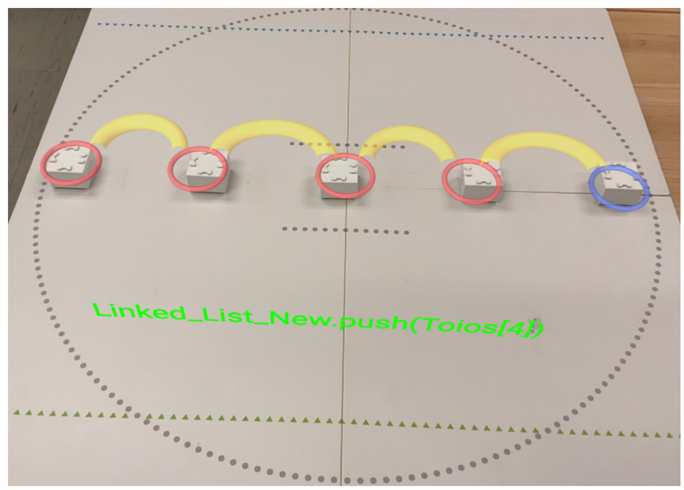

Figure 7: When the fifth Toio robot is added into the linked list, a virtual line is added to link the fourth and fifth Toio robots in the AR display.

then shows the code of this action of Toio robot initialization. Next, the AR display shows blinking circles to indicate the potential positions of the Toio robots in the linked list at the centre of the tracking mat, while the code generation terminal shows the code of initializing an empty linked list. The user then moves four Toio robots from the sideline to the centre of the tracking mat one by one matching the red circles in the AR display.

After each Toio robot is moved to the centre, a virtual line is shown between each pair of the Toio robots next to one another. At this time, each Toio robot's moving action is translated into one line of code in the code generation terminal.

The user then adds a Toio robot to the linked list by moving a Toio robot from the sideline to the centre of the tracking mat at

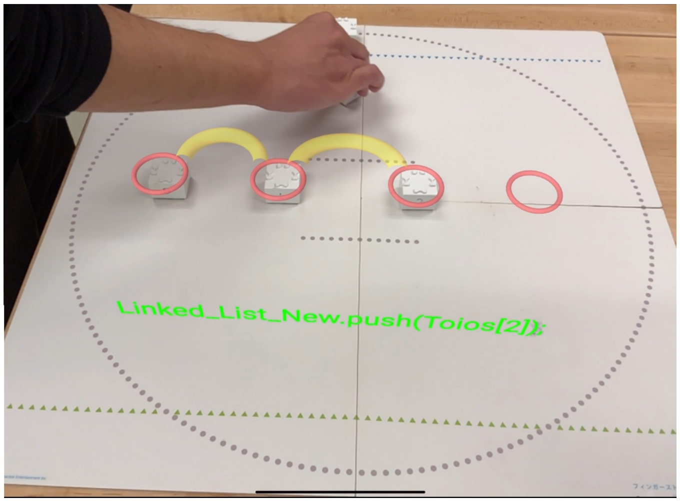

Figure 8: The user moves each Toio robot from the sideline to the centre of the tracking mat.

Figure 9: After each Toio robot is moved by the user from the sideline to the linked list, one line of code is generated to reflect such a movement and position change in the code generation terminal.

the end of the existing linked list. At the same time, the AR display shows a blinking empty spot for the user to place the Toio robot. A line of code to push the Toio robot is then generated in the code generation terminal. Lastly, the user moves the Toio robot from the line in the centre to the sideline, in order to remove the Toio robot from the linked list. The virtual line between the removed Toio robot and its neighbour then disappears. At this time, a line of code to pop the Toio robot from the linked list is generated in the code generation terminal.

At the end of the code generation, the user can click on the button to optimize the code by combining all the individual lines of adding Toio robots to the linked list into a for loop where the linked list adds one Toio robot repeatedly four times.

Figure 10: Before the generated code is optimized, lines of code generated from adding Toio robots to the linked list are separate.

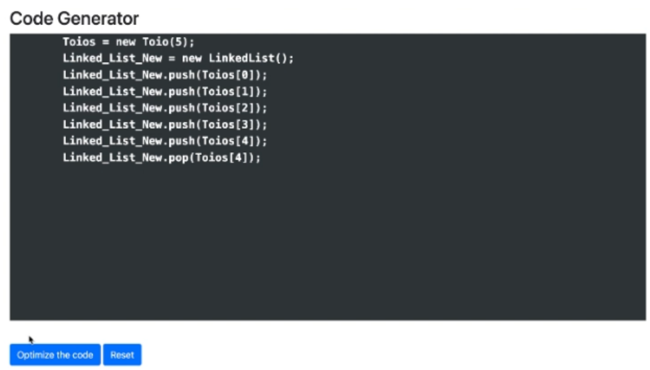

Figure 11: After the generated code is optimized, the separate lines of code showing Toio robots addition to the linked list are transformed into an equivalent for loop.

5 SYSTEM IMPLEMENTATION

5.1 Server Connection, Data Transmission, and Hosting

CodeSwarm-AR relies on a server-client connection to enable the synchronization of its system components, including a desktop web interface, an AR interface, and Toio robot control and tracking system. Node.JS [5] server framework is used to construct the server environment, and Socket.IO [7] is used to enable data transmission among all the aforementioned system components. With this setup, the desktop web page's code samples, the AR interface's virtual objects, and Toio robots' positions and movements can be synchronized based on the user's inputs through the desktop web page or the Toio robots. During each input action from the user, data is transmitted in the form of JSON among all the system components and enables them to invoke the intended contents and behaviours. Glitch.com [4] is used to edit and save the entire code base of CodeSwarm-AR consisting of HTML, CSS, and JavaScript. It is also used to host all the system components in a cloud environment.

5.2 Desktop Web Page Interface

CodeSwarm-AR displays its code samples and step-by-step demonstrations in two web page interfaces. The interfaces are implemented with Bootstrap [2] front-end framework and run on a 2018 15-inch MacBook Pro via Chrome web browser. The first web page interface displays the pre-built code samples that can be demonstrated by the Toio robots when the user executes the sample demos. The user can also manipulate the Toio robots to generate new code samples that are displayed in the second web page interface.

5.3 AR Interface

The AR display component of CodeSwarm-AR is built with A-Frame [1] web AR framework and run on a 2021 13-inch iPad Pro tablet via the web browser Mozilla WebXR Viewer [8]. The tablet is placed on a tablet stand in front of the Toio tracking mat with its camera pointing to the middle of the mat at approximately 45 degrees of angle. With this setup, the tablet camera's field of view covers the entire tracking mat. An AR scene is created to store virtual objects that animate based on the user's inputs in the desktop web page or the Toio robots. The virtual objects in the AR scene and on the tablet's screen assist in displaying the code samples and their translated behaviours in the Toio robots. The positions of the virtual objects are calculated to match the physical objects in the tablet's view, the Toio robots and their tracking mat; therefore, the virtual objects are seen as being attached to the moving Toio robots and located at the centre of the tracking mat.

Figure 12: The overall system setup with Toio robots, tablet (AR device), and computer (web terminals).

5.4 Connection, Control and Tracking of Swarm Robotics

The swarm robotics system component consists of five Sony Toio robots, a tracking mat, and a web-based program run on a 2018 15-inch MacBook Pro via Chrome web browser to control the Toio robots via Bluetooth technology. Each Toio robot is 3.2 cm x 3.2 cm x 2.5 cm in dimension and has a ground speed of up to 20 cm/s. The Toio tracking mat is 55 cm x 55 cm with a 1-mm tracking precision. The control program is built with p5.toio library [6] in JavaScript and can connect, control, and track Toio robots based on the user's inputs from the first web page interface or transmit data to generate new code in the second web page interface based on the user's inputs from the Toio robots.

6 DISCUSSION OF LIMITATIONS AND FUTURE WORK

Due to the time constraint of this research project, several limitations are present in our design concept and system application and implementation.

6.1 Unidirectional Input Methods

The concept of bidirectional input method in CodeSwarm-AR was explored with two isolated scenarios of unidirectional input methods of opposite directions rather than a truly complete bidirectional input method. To limit the scope of this project to only the interaction techniques, the system implementation centres around the individual demonstrations of the two input methods. In the future, an improvement is to modify the current prototype to enable the users to enact both input methods in the same coding terminal where code can be demonstrated, modified, and generated in the same workflow.

6.2 AR Display Modality

This project achieves a less time-consuming system prototype with tablet and mobile AR; however, our vision of AR display supplementing the robotics control can be better realized by head-mounted display (HMD). As [32] points out, AR can be displayed to the user through different approaches, and our system leverages "on-body handheld mobile AR", which is in the format of a tablet positioned near the user's eyes. A noticable improvement is the use of HMD instead of tablet so that the user will not be limited to the less flexible position of the tablet on the desk. HMDs can enable the users to view AR contents while having considerable amount of freedom of head and body movement while interacting with the swarm robotics.

6.3 Scale of Robotics Movement and Tracking

Sony Toio robots have the advantage of easy setup and short compilation thanks to the [6] library. However, Toio robots' tracking is limited to the size and position of the tracking mat(s). Like this project, Toio robots are limited to moving and tracking on a small portion of the tabletop with their full tracking capabilities. Our future work involves the development of a better tracking infrastructure that enables the full table-top and even room-scale tracking for swarm robotics. With this future exploration, our system prototype can be expanded into a bigger scale, which enables multiple users to interact with swarm robotics to teach and learn programming concepts.

6.4 AR-Facilitated Interactions

The current system leverages the AR display to demonstrate animated geometric shapes and texts to better explain Toio robots' movements, positions, and connections. However, the AR display is limited to unidirectional display without any interaction between the user and the AR contents. In the future, our system will be enhanced with the capabilities of AR interaction so that the freedom from the stationed computer screen can be realized.

7 CONCLUSION

Here, we present CodeSwarm-AR, an interactive system for programming education that utilizes swarm robotics and augmented reality (AR). Offering a bi-directional code input method via Toio robots, CodeSwarm-AR focuses on making programming concepts like data-structures visual, interactive, and more intuitive for the audience to conceptualize. Various interaction techniques for CodeSwarm-AR were introduced, such as the conventional on-screen programming education system along with physical manipulation of the robots acting as proxies, code-to-robot interaction with a code sample, and robot-to-code interaction, in which changes in one environment leads to a result in another with a line-by-line AR display of code execution, robots' movement, and its demonstrations. We discussed the challenges we faced along the way and the future work required to address each of the issues, such as the modality of AR display and Toio tracking limitations. But overall, the combination of Toio robots, augmented reality, and swarm robotics creates a highly interactive and engaging learning environment that can motivate students and promote collaboration and teamwork. The ability to see and interact with physical robots can provide a more immersive and memorable learning experience, allowing students to understanding programming concepts and robotics principles, while also honing their problem-solving and critical thinking skills.

8 ACKNOWLEDGEMENTS

Our team thanks the CPSC 584 course instructor Dr. Ryo Suzuki and the course teaching assistant Shivesh Jadon for their guidance in this research project.

During the ideation phase, Dr. Suzuki and Shivesh provided the high-level idea of tangible programming education with swarm robotics for our team. At the beginning of the implementation phase, Dr. Suzuki provided guidance on the tracking of Toio robots.

Part of this report was improved by using ChatGPT [3] for proofreading, general editing, and grammer correction.

REFERENCES

[1] [n.d.]. A-Frame Documentation. https://aframe.io/
[2] [n.d.]. Bootstrap Documentation. https://getbootstrap.com/
[3] [n.d.]. ChatGPT. https://openai.com/blog/chatgpt
[4] [n.d.]. Glitch.com. https://glitch.com/
[5] [n.d.]. Node.JS Documentation. https://nodejs.org/
[6] [n.d.]. p5.toio Documentation. https://tetunori.github.io/p5.toio/
[7] [n.d.]. Socket.IO Documentation. https://socket.io/
[8] [n.d.]. WebXR Viewer. https://labs.mozilla.org/projects/webxr-viewer/
[9] [n.d.]. YouTube. https://www.youtube.com/
[10] Bradley S Barker and John Ansorge. 2007. Robotics as means to increase achievement scores in an informal learning environment. *Journal of research on technology in education* 39, 3 (2007), 229–243.
[11] Valerie Barr and Chris Stephenson. 2011. Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *Acm Inroads* 2, 1 (2011), 48–54.
[12] David Bau, Jeff Gray, Caitlin Kelleher, Josh Sheldon, and Franklyn Turbak. 2017. Learnable programming: blocks and beyond. *Commun. ACM* 60, 6 (2017), 72–80.
[13] Brandon Antonio Cárdenas-Sainz, María Lucia Barrón-Estrada, Ramón Zatarain-Cabada, and José Mario Ríos-Félix. 2022. Integration and acceptance of Natural User Interfaces for interactive learning environments. *International Journal of Child-Computer Interaction* 31 (2022), 100381.
[14] Marcos Devaner do Nascimento, Igor Moreira Félix, Bernardo Martins Ferreira, Lucas Mendonça de Souza, Douglas Lima Dantas, Leônidas de Oliveira Brandão, and Anarosa de Oliveira Brandão. 2019. Which visual programming language best suits each school level? A look at Alice, iVProg, and Scratch. In *2019 IEEE World Conference on Engineering Education (EDUNINE)*. IEEE, 1–6.
[15] Salomi Evripidou, Angelos Amanatiadis, Klitos Christodoulou, and Savvas A Chatzichristofis. 2021. Introducing algorithmic thinking and sequencing using tangible robots. *IEEE Transactions on Learning Technologies* 14, 1 (2021), 93–105.
[16] Anna Fuste and Chris Schmandt. 2019. HyperCubes: a playful introduction to computational thinking in augmented reality. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*. 379–387.
[17] Philip J Guo. 2013. Online python tutor: embeddable web-based program visualization for cs education. In *Proceeding of the 44th ACM technical symposium on Computer science education*. 579–584.
[18] Kate Highfield. 2010. Robotic toys as a catalyst for mathematical problem solving. *Australian primary mathematics classroom* 15, 2 (2010), 22–27.
[19] Qiao Jin, Danli Wang, Xiaozhou Deng, Nan Zheng, and Steve Chiu. 2018. AR-Maze: a tangible programming tool for children based on AR technology. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*. 611–616.
[20] James Kennedy, Paul Baxter, Emmanuel Senft, and Tony Belpaeme. 2016. Social robot tutoring for child second language learning. In *2016 11th ACM/IEEE international conference on human-robot interaction (HRI)*. IEEE, 231–238.
[21] Dennis Komm, Adrian Regez, Urs Hauser, Marco Gassner, Pascal Lütscher, Rico Puchegger, and Tobias Kohn. 2020. Problem solving and creativity: Complementing programming education with robotics. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. 259–265.
[22] Daniel Leyzberg, Samuel Spaulding, Mariya Toneva, and Brian Scassellati. 2012. The physical presence of a robot tutor increases cognitive learning gains. In *Proceedings of the annual meeting of the cognitive science society*, Vol. 34.
[23] Tianren Luo, Mingmin Zhang, Zhigeng Pan, Zheng Li, Ning Cai, Jinda Miao, Youbin Chen, and Mingxi Xu. 2020. Dream-experiment: a MR user interface with natural multi-channel interaction for virtual experiments. *IEEE Transactions on Visualization and Computer Graphics* 26, 12 (2020), 3524–3534.
[24] Stéphane Magnenat, Morderchai Ben-Ari, Severin Klinger, and Robert W Sumner. 2015. Enhancing robot programming with visual feedback and augmented reality. In *Proceedings of the 2015 ACM conference on innovation and technology in computer science education*. 153–158.
[25] Oussama Metatla, Sandra Bardot, Clare Cullen, Marcos Serrano, and Christophe Jouffrais. 2020. Robots for inclusive play: Co-designing an educational game with visually impaired and sighted children. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
[26] Joseph E Michaelis and Bilge Mutlu. 2019. Supporting interest in science learning with a social robot. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*. 71–82.
[27] Ayberk Özgür, Séverin Lemaignan, Wafa Johal, Maria Beltran, Manon Briod, Léa Pereyre, Francesco Mondada, and Pierre Dillenbourg. 2017. Cellulo: Versatile handheld robots for education. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. 119–127.
[28] Daisuke Saito and Tsuneo Yamaura. 2013. A new approach to programming language education for beginners with top-down learning. In *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*. IEEE, 752–755.
[29] Brian Scassellati, Jake Brawer, Katherine Tsui, Setareh Nasihati Gilani, Melissa Malzkuhn, Barbara Manini, Adam Stone, Geo Kartheiser, Arcangelo Merla, Ari Shapiro, et al. 2018. Teaching language to deaf infants with a robot and a virtual human. In *Proceedings of the 2018 CHI Conference on human Factors in computing systems*. 1–13.
[30] Gordon Stein and Ákos Lédeczi. 2019. Mixed reality robotics for stem education. In *2019 IEEE Blocks and Beyond Workshop (B&B)*. IEEE, 49–53.
[31] Masanori Sugimoto, Tomoki Fujita, Haipeng Mi, and Aleksander Krzywinski. 2011. RoboTable2: a novel programming environment using physical robots on a tabletop platform. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*. 1–8.
[32] Ryo Suzuki, Adnan Karim, Tian Xia, Hooman Hedayati, and Nicolai Marquardt. 2022. Augmented reality and robotics: A survey and taxonomy for ar-enhanced human-robot interaction and robotic interfaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–33.
[33] Anastasios Theodoropoulos and George Lepouras. 2021. Augmented Reality and programming education: A systematic review. *International Journal of Child-Computer Interaction* 30 (2021), 100335.
[34] Ana M Villanueva, Ziyi Liu, Zhengzhe Zhu, Xin Du, Joey Huang, Kylie A Peppler, and Karthik Ramani. 2021. Robotar: An augmented reality compatible teleconsulting robotics toolkit for augmented makerspace experiences. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.
[35] Randi Williams, Hae Won Park, and Cynthia Breazeal. 2019. A is for artificial intelligence: the impact of artificial intelligence activities on young children's perceptions of robots. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.

[36] Ryan Wistort and Cynthia Breazeal. 2011. TofuDraw: A mixed-reality choreography tool for authoring robot character performance. In *Proceedings of the 10th International Conference on Interaction Design and Children*. 213–216.

[37] Xiao Xiao, Pablo Puentes, Edith Ackermann, and Hiroshi Ishii. 2016. Andantino: Teaching children piano with projected animated characters. In *Proceedings of the the 15th international conference on interaction design and children*. 37–45.