MicrokARts: An Interactive System for Co-Located AR-IoT Interactions with Children

ANONYMOUS AUTHOR(S)

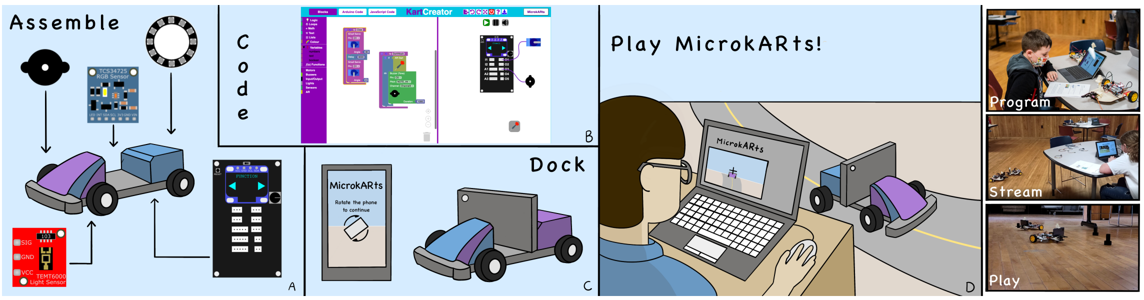

Figure 1: MicrokARts is an Augmented Reality platform designed to help children design and program electro-mechanical devices, while collaborating on tasks in a dynamic AR-IoT environment. Users (A) decide which electronics they want to put on their MicrokARt, (B) program their MicrokARt using our block-based live programming website, (C) dock the phone onto the MicrokARt, and (D) control their MicrokARt and play with others through AR-IoT interactions.

Augmented Reality (AR) is a popular tool for youth to engage with technology in exciting ways; coupled with the Internet of Things (IoT), these domains create unique opportunities for youth to explore, learn, and play. However, many systems today enable such experiences without allowing end-users to customize the contents' looks or behavior. To this end, we developed MicrokARts, allowing children to (1) create an IoT kart, clad with actuators and sensors, (2) program using our block-based programming environment, and (3) interact wirelessly in a dynamic AR-IoT environment. We tested our initial system with 4 graduate student experts and 15 youth (age=11–18), before testing MicrokARts with 22 users (ages=9–15). We observed how children designed their interactions using AR and IoT devices, and their engagement with the technology. With MicrokARts, we contribute a system that supports youth in designing new, creative experiences with AR-IoT interactions.

CCS Concepts: • **Applied computing → Interactive learning environments**; • **Human-centered computing → User interface toolkits**; *Collaborative and social computing systems and tools*; *Collaborative and social computing devices.*

Additional Key Words and Phrases: Children, Augmented Reality, Internet of Things, Live Programming, Shared AR

ACM Reference Format:

Anonymous Author(s). 2022. MicrokARts: An Interactive System for Co-Located AR-IoT Interactions with Children. In *CHI '22: ACM CHI Conference on Human Factors in Computing Systems, April 30–May 6, 2022, New Orleans, LA.* ACM, New York, NY, USA, 29 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Augmented Reality (AR) is a technology that enhances the world around us by superimposing digital information within our physical space through the lens of a cell phone/tablet camera, or a Head-Mounted Display. Such digital information can include images, avatars, 3D models and much more [88]. The core value of AR is that it dually exists in two formerly separate realms – the *physical* and *virtual* realms – and effectively integrates what is powerful about each of the ideas that it draws from. With the advent of Google's ARCore [29] and Apple's ARKit [1], the technology is now being placed in the hands of consumers through companies like Samsung's AR Zone [32], Niantic's Pokemon Go [40], and other popular mobile apps [79, 80, 92]. Children, in particular, are able to use these applications to create an exciting world that takes their understanding of the physical realm and infuses their own imaginations into it. However, current technological advances are starting to allow for more seamless integration between AR and the real world through the Internet of Things (IoT) (e.g. Mario Kart Live [61] and meSchup [47]). By connecting devices to a network, it becomes possible to create more dynamic environments where physical devices and virtual components can co-exist.

Arduino, Micro:bit, and other popular microcontrollers enable their users to connect their devices to others via WiFi or Bluetooth Low Energy (BLE) through the use of shields, extensions, and some built-in functionalities [3, 82]. These devices are particularly useful due to their ability to connect to peripheral devices such as sensors and actuators. Due to the many technological advances in programming techniques, circuit modeling software, and more, children are starting to utilize these physical computing devices for more engaging affairs. For instance, Scratch Link is a way for users (namely children) to connect a Micro:bit device to the computer and build creative IoT projects through programming [78]. Bird Brain Technologies has also developed a kit involving a Micro:bit device that aids users in quickly creating IoT based projects through a simplified electronics interface [84]. However, opportunities for engaging with technologies like AR and IoT simultaneously are generally under explored.

Interactions between AR components and IoT devices are beginning to be explored more actively, particularly with children. Earlier work [96, 97] enables virtual content to interact with physical objects, but those physical objects aren't programmable. Villar, et. al. explore the use of AR interactions with physical devices, but with static, pre-defined objects that can have the Microsoft Hololens device respond when the user interacts with it [91]. Even Nintendo™utilizes AR-IoT interactions with its IoT karts that can be wirelessly connected to a Nintendo Switch device for a remote controlled Mario Kart game [61]. However, all the prior technologies have a critical gap. They do not allow users to create, program, or alter the physical devices (or the software that comes with them in some cases) in an active way.

MicrokARts fills that gap as an AR-IoT platform that empowers children with the ability to create dynamic environments where AR content can trigger reactions from wirelessly connected programmable devices in the real world. With MicrokARts, users (1) decide which components from our electronics toolkit to use with their MicrokARt and plug them into our customized IoT board, (2) program them with Kart Creator, our block-based live programming web application, (3) connect an AR-enabled cell phone to the computer, and (4) create dynamic AR-IoT environments with other users. Figure 1 highlights the aforementioned steps that users take to create their own MicrokARt device for wireless play. To evaluate the MicrokARts system, we first conducted an expert pilot study with four users to evaluate Kart Creator. We then studied Kart Creator with 15 youth users to further inform the design of the AR-IoT interactions of MicrokARts. From the feedback gathered from these studies, we conducted a final user study of MicrokARts with 22 youth users to understand how youth use the system to enhance creative play. All studies were conducted in-person, and precautions were taken to ensure safety against COVID-19. We offer the following contributions with this system:

- The MicrokARts system's end-to-end workflow, which features plug-and-play electronics, a block-based programming application with a live simulation tool, and a multi-user AR-IoT environment to control physical devices with AR-IoT interactions,
- A solution for children to create dynamic AR-IoT environments through block programming, and

- A series of Recommendations for Future Systems based on our study results to inform further work on interactive and creative play through physical computing and Augmented Reality.

Thus, the contributions of this paper are directly linked to the MicrokARts system, workflow, and its various parts.

2 RELATED WORK

In this section, we discuss prior works related to MicrokARts in several areas, including digital-physical interactions, shared AR experiences, and tangible systems for embedded interactions.

2.1 Digital-Physical Interactions

A number of systems that are available commercially, or in research, attempt to utilize the dual nature of AR. By programming the AR components to become interactable with physical devices that are connected to a network, systems begin to unlock more ways for users to engage with the technology. Utilizing this dual nature of AR is of particular interest here because there is room for AR to become more than just a feature on a cell phone, but an integral component to emerging, interactive platforms for youth (which is a major motivating factor that we explore in this paper). Figure 2 highlights some prior works that push the boundaries between visually programming a physical device and creating digital-physical interactions with them. For example, Motionbeam and Hideout are two works that enable digital-physical interactions by using a handheld projector to show their virtual content on a plain surface (e.g., a wall or table) [96, 97]. The AR content in these examples can activate various functions that are built into their physical devices, such as a painting falling off the wall using a camera or infrared sensor connected to the projector. ProtoAR, while pushing the boundaries of using physical objects to create AR experiences, gives its users a visual interface and mixes it with clay/paper prototypes to generate low-fidelity mock ups of AR interfaces [58].

Narumi, et. al. developed ConductAR, which is another system that helps blend the physical and digital realms by using conductive ink and AR to show its users how electricity will flow through a drawn circuit [57]. Systems like ConductAR highlight how powerful AR can be for learning and engaging users with meaningful content while also being accessible to them. Finally, prior works such as ARBlocks [75], AR-Maze [43], AR Scratch [71], CollabAR [95], and BlocklyAR [59] all use image recognition of QR codes, character pictures, and other images to help enable their AR experiences. This combination of digital and physical components is helpful for engaging in learning activities, creating certain AR experiences, and giving the user agency to author those experiences through some form of visual programming interface. However, all of the aforementioned works are simply the foundation upon which digital-physical interactions rest today and are not reflective of the current state of technology.

Heun, et. al. created the Reality Editor, which places a digital layer on top of commonly used physical items, thus making them interactable by users [35]. This workflow is powerful because it helps dismantle any existing barriers between the physical and digital realms through its simple interfaces and AR interactions. When applied to children, Reality Editor also embodies a visual programming interface that helps users create a network of devices that can interact with each other; however, the AR here serves more as an interface to program interactions between physical and digital content, rather than the interactable content and/or the content to be interacted with. Putting this idea of creating digital-physical interactions into the hands of children has also been explored in the past. One example is from Glenn, et al. who developed StoryMakAR, which demonstrates a block-programming environment, an authoring tool for user-defined AR-IoT interactions, and an AR application where the interactions actually happen [26]. However, this work does not support multi-user AR-IoT experiences, and the interactions between physical and digital content are specific to storytelling. Lastly, Nintendo recently released a new game called Mario Kart Live: Home Circuit, which provides users with a physical kart with an AR camera attached to it [61]. Users can play the game through the lens of the camera on the kart and race against AR characters or other players in the same space. Users can pick and place AR components onto their tracks and have the AR obstacles cause Mario's kart to stop (e.g., after being hit with a shell). Although the aforementioned prior works enable digital-physical interactions for users, they do not enable the user to author such interactions themselves. This is where MicrokARts separates itself from other platforms by enabling the user to author such digital-physical interactions. MicrokARts gives multiple users the ability to author their own digital-physical interactions, replace components, and program electronics; thus, giving users the decision making power for the experiences they want to have while playing with each other.

2.1.1 ***Shared AR Experiences:*** Mobile AR applications such as Pokemon GO™, WallaMe [92] and Snaappy [79] use inbuilt GPS to track users and render the AR content based on their geographical locations. Recent industrial AR applications have allowed users to view AR content in-situ, while being located remotely [69]. Similarly, remote AR/VR applications have been explored in research by Loki [85]. With recent developments in AR hardware and software, users can now create a virtual avatar and collaborate remotely with others in a physical space with Spatial.io [42]. Although such shared AR experiences overcome geographical boundaries and closely represent in-person interactions, they don't provide similar digital-physical interaction and DIY features that are akin to the creative endeavours of maker-based projects. MicrokARts addresses this challenge by providing a shared AR experience in a co-located space with features to create and program an interactive IoT MicrokARt.

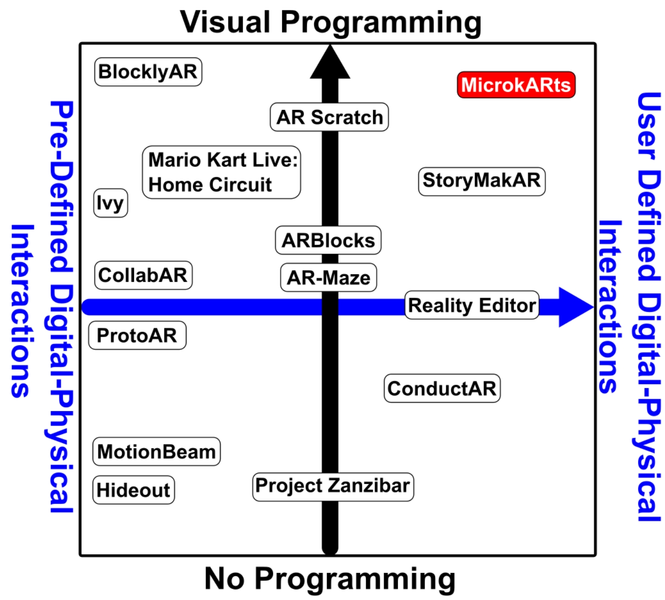

Figure 2: *A novel Quad Chart depicting several works related to MicrokARts. The x-axis shows the spectrum of works that include pre-defined interactions between digital and physical components vs. user defined interactions. The y-axis shows the spectrum of works that do not involve programming vs. works that have a visual programming interface. Citations: [21, 26, 35, 43, 57–59, 61, 71, 75, 91, 95–97]*

Transvision [72] allows co-located users to collaborate by projecting 3D objects on a table-top surface and viewing it via a hand-held display. This aids in improving engagement and participant satisfaction because collaborators can see the actions and gestures of their teammates, while still allowing them to communicate naturally. Shared AR experiences have found a wide range of applications from crime scene investigation [16] to authoring and peer discussion of hands-on STEM sessions [90]. Such experiences also vary in the scale of the environments from a table-top [72] to a large room [38] to even extending up to city scales [76]. Shared AR for co-located and isolated participants create two distinct kinds of experiences and have their advantages and limitations. MicrokARts attempts to take advantage of co-located shared AR experiences to perform collaborative tasks, while still enabling digital-physical interactions.

2.2 Tangible Systems for Embedded Interactions

Tangibles are an integral component to engaging children with technology. There are several works related to MicrokARts that create tangible devices for children to interact and play with. To start, Resnick et. al. helped introduce the idea of physical computing devices [73]. The idea of a toy that can be programmed by children falls in line with the devices that we are familiar with today [3, 22, 23, 82]. Extending these, researchers have taken this idea and begun to further this research in several ways.

2.2.1 ***Tangible Electronics Toolkits:*** Blikstein surveys various toolkits that examined the design principles of several physical computing platforms [6]. The author suggests that physical computing offers numerous opportunities for children to use and learn from these devices. For example, platforms like LittleBits™ lets children attach electronics modules together using built-in magnets [4]. Users connect the modules together in a logical order to create their own circuits. MakerWear builds on this idea by allowing children to attach small electronics modules (e.g., actuators, sensors, and displays) to their clothes [45]. Proxino is a virtual circuit platform that helps users prototype circuits using physical proxies, which also invokes digital-physical interactions [99]. These tangible electronics toolkits offer high engagement with electronics for users and have some similarities with MicrokARts. MicrokARts adds a new dimension of AR-IoT interactions to create electro-mechanical devices and program them while retaining the plug-and-play electronics board and low barrier to entry.

2.2.2 ***Live Programming:*** Researchers, software developers, and large corporations are showing great interest in live programming as a way to enhance the programming experience. According to Nilson, live coding is *"the art of programming a computer under concert conditions, [and] can enable a novel engagement with the notions of algorithm, and the mapping from code to musical resultant"* [13, 60, 93]. Although music and live coding is not the main focus of the MicrokARts system, the act of "liveness" is attractive for engagement. Live programming refers to when the user's actions receive immediate feedback that shows the effects of what the user did. This powerful notion is displayed in numerous places, which Kubelka, et. al explore in their review paper [46]. Apple and Facebook have also developed their Swift and React platforms, which give developers options to see what will happen when they implement certain features to their software in real-time [20, 83].

Microsoft offers a block programming software [49] on their MakeCode website for the Micro:bit device, which displays a wiring diagram of the breadboard, electrical components, and even the Micro:bit itself [8]. Modkit, developed by Milner, Baafi, Qiu et. al., introduces live programming to a physical Arduino board by allowing users to program their electronics with blocks, then see the affects of their code in real-time on the physical board [53, 70]. How could the lessons learned from prior works such as these be applied to other areas (e.g. AR-IoT interactions)? We explore this question by evaluating the MicrokARts system in three studies as indicated in later sections. MicrokARts separates itself from other platforms due to the intersection of the areas covered in this section.

2.2.3 ***Connected Toys:*** Children's play objects have become a popular tool for them to also learn with. However, what can children learn about IoT by playing with connected toys? Connected toys is an area that is being explored more because smart environments are more prevalent due to the abundance of computers, mobile phones, smart objects, and online communities that children have access to [36, 39]. Zaman, et. al. sought to answer questions similar to this when they explored the Internet of Toys (IoToys) [100]. These studies showed that IoToys can give children insight into both the physical and digital realms when designed with this intention in mind. We aimed to keep these design considerations in mind while developing MicrokARts. The following section describes our motivation to design the initial MicrokARts prototype, our pilot studies, and the consequential design goals that we developed for the system.

3 MOTIVATION & DESIGN GOALS

Our approach to designing a multifaceted system like MicrokARts considered challenges faced by children when using physical computing devices at a novice level (e.g. syntax and incorrect wiring). In this section, we review prior research and commercial products that informed the design goals for our initial MicrokARts prototype. We also describe our initial pilot studies with the prototype and explain how the system evolved.

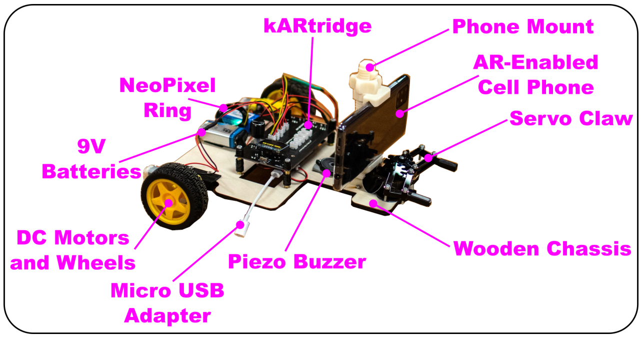

Figure 3: *One MicrokARt chassis that participants used during our user study. The MicrokARt consists of a chassis, two yellow DC motors, the claw and the kARtridge from our Electronics Toolkit (see Section 4.1), a ball castor wheel, two 9V batteries, an AR-enabled cell phone, and a mount to dock the phone onto the chassis. Users control this device wirelessly from a computer and receive an AR video stream from the MicrokARts app (see Section 4.2).*

Earlier work suggests that children tend to encounter usability issues with software, which can ultimately lead to frustration, disinterest, or boredom [12, 18, 25, 37, 65]. These same issues can occur if there is a significant learning curve to using the software efficiently, or if opportunities for scaffolding are scarce [15, 24, 64, 74]. As stated in Section 2.2.2, Live Programming is appealing because it removes the arduous aspects of programming (e.g. compile times) and allows the user to focus on the functionality of the program they are creating while receiving almost immediate feedback. With the Blockly platform from Google, developers have broken down these barriers by reducing the cognitive load commonly associated with computer programming [49, 54, 62, 66]. Thus, we can devise that combining block programming and live programming can significantly improve youth's experiences of using such systems [8, 53].

Physical computing devices have benefits and challenges associated with their software and hardware aspects. For example, wiring diagrams tend to become messy on screen and can further add to the frustration of a child without prior electronics knowledge [89]. Some toolkits have begun to simplify this process by adding some plug-and-play features to their electronics interfaces, promoting easier connection points for users [26, 41, 84]. The Thymio robot, for example, is an open source educational platform that is designed for multiple ages; it promotes creativity, encourages learning, and provides a wide range of interaction possibilities, from built-in behaviors to text programming [55]. However, the hardware subsystem of Thymio presents a closed structure with its circuitry/sensors being embedded inside of its chassis, restricting its access to users. Consequently, there are opportunities to channel the plug-and-play functionalities of these electronics toolkits into a toolkit of our own design rather than use a pre-existing toolkit/programming environment. Thus, we developed the Kart Creator web application as our initial MicrokARts prototype.

3.0.1 ***Kart Creator:*** Kart Creator allows anybody to program and customize their own IoT-enabled kart. Kart Creator is built using Google's Blockly™ programming interface, which is a web-based drag-and-drop environment, to enable users to create code by dragging and connecting blocks in a visual way [49, 54]. The blocks are broken down into two main sections: Reserved programming terms (e.g. loops, conditionals, and variables), and hardware components (Referenced in Section 4.1). Each block has its own customizable properties so every device is unique. As users drag their code into the window, they can watch their Arduino™ code build in the code tab so they can learn how their code is structured and investigate how it works. Kart Creator was designed with a live simulator tool to allow users to see how their Kart will function in the real world. As users create functions for their devices (e.g. a function that flashes LEDs and spins a servo motor), they can run this function in the simulator to show the order and length of the events, and whether the function runs as the user envisioned it.

3.1 Pilot Study: Initial MicrokARts Prototype

We tested Kart Creator with 4 expert users for feedback and validation of the system, and 15 youth users to learn how youth might benefit from the system. Given our curated design goals, we set out to develop an initial prototype of the MicrokARts system without the AR. We first designed and tested our kARtridge PCB along with the initial version of Kart Creator. This was done intentionally to test the physical subsystems *before* testing the digital ones. Our findings were used to improve our final system design for MicrokARts and to inform the design of our final user study.

3.1.1 ***Expert Pilots:*** Our pilot study started with 4 participants (3 male, 1 female) who are all graduate students at a large Midwestern university, as well as experts in (1) Building & Using Electro-Mechanical Devices, (2) Electronics & Circuitry, or (3) Programming Physical Computing Devices. We consider an expert to be a person with 3 years or more of academic and/or professional experience working in that area. The objective of this study was to gain validity of the system and our methodology for the study (described below) before testing the system with youth.

3.1.2 ***Expert Pilot Setup:*** We designed a within-subjects study to determine any changes or modifications that were necessary in our electronics toolkit and Kart Creator. Through this study, we aimed to explore the potential for Kart Creator to be a functional platform to test and program electro-mechanical devices. Participants were required to wear masks throughout the entirety of the study, and all equipment was sanitized before and after each study. Finally, participants were given $20 compensation after the study.

3.1.3 ***Expert Pilot Methodology:*** While designing our pilot study methodology, we recognized that our methods should be grounded in research related to evaluating new systems and technology. Ledo, Houben, Vermeulen, et. al. investigated such evaluation methods by reviewing 68 HCI toolkit research papers [48]. In this review paper, they derived 4 evaluation strategies for HCI toolkits: (1) demonstration, (2) usage, (3) technical evaluation, and (4) heuristic evaluation, which can be used individually or in some combination. Our pilot studies were designed based on the *usage* evaluation strategy to verify usability *and* utility of our preliminary system by youth (see section 3.1.4). Ledo, Houben, Vermeulen, et. al. also provided *ways to conduct usage studies*, of which we used the following: (1) usability studies, (2) walkthrough demonstrations, (3) observations, (4) Likert scale questionnaires, and (5) open-ended interviews.

All participants were given a pre-survey with questions about their experience levels and demographics before starting the study. Pre-survey questions were adapted from the Internet Proficiency Scales by Eastin and LaRose [19] (See Appendix A). A quick tutorial of Kart Creator and its functionalities was given before being introduced to the main activity, which was a set of coding challenges. The coding challenges were conveyed to participants via an in-workshop pseudo-code packet with instructions on how to progress from one step to the next. The instructions were written using a scaffolding teaching practice that is employed in Computer Science and Physical Computing Education [7, 101] so that our participants could build on what they learned as they progress. The coding challenges were as follows: (i) Rotate a yellow motor clockwise for 5 seconds and counter-clockwise for 5 seconds, (ii) activate the Neopixel when the light sensors detects no light, (iii) activate the Neopixel when the RGB sensor sees something red, (iv) use a `for loop` to rotate a servo motor from 0^{o} to 180^{o}, and (v) build a kart that you can control with our "Test My Code" debug tool (See Figure 5). The participants in both the expert and youth pilots used a version of the chassis as depicted in Figure 3 without the phone mount, phone, and servo claw, all of which were added as a result of our pilot studies. Coding challenges 1–4 ask users to follow specific guidelines; whereas, coding challenge 5 is open-ended to get the participants to apply what they've learned (see Supplementary Materials for a copy of the pseudo-code document). The study lasted approximately 2 hours for each participant. A post-survey was given and a post-study interview was conducted to gather feedback from the experts.

3.1.4 ***Youth Pilots:*** Following our expert pilots, we conducted several individual and group workshops for our Youth Pilot with children in our target age range (age = 11–18). Our age range encompasses the age at which students are in middle school (10–14) and high school (14–18) in the United States [17]. The objective of this study was to test the workflow and usability of Kart Creator by youth, to assess which hardware components they are most likely to use for a kart, and to further validate our scaffolding pseudo-code methodology. Each user was recruited via word of mouth, as well as postings on our social media pages. Informed consent for the study was received from all participants (if age = 18) and parents (if age < 18), and assent from some participants (if age < 14) as required by our university's Institutional Review Board. Overall, we had 15 participants across seven workshops (4 male, 11 female).

3.1.5 ***Youth Pilot Setup:*** This study lasted approximately 90 minutes for each participant/group and followed a similar format to our Expert Pilot. However, the fourth coding challenge was omitted here due to a lack of use for the servo motor. Some participants conducted their studies in small groups, while others were individual. The decision to conduct studies in small groups versus individually was done on a case-by-case basis since some parents did not want their children to be in groups due to the COVID-19 pandemic. Masks were required and all equipment was sanitized before and after the study to keep participants and researchers safe.

3.1.6 ***Youth Pilot Methodology:*** We decided to precompile the code for challenges 1–4 in this study to reduce the waiting time. The precompilation did not hinder the user experience since the outcome of each coding challenge was the same with the exception of the final challenge. This did not mean that what the children were doing on Kart Creator did not matter, rather the researchers ensured that the code was functioning properly on-screen before uploading the code to the kARtridge. Participants were compensated $20 for taking part in the study. At the conclusion, the children were asked to participate in a short interview and fill out a post-study survey.

3.1.7 ***Results:*** In this section, we report all Likert-scale questions with the mean (M), standard deviation (σ), and median (m). The subscript E depicts experts and Y depicts youth. All names are reported as pseudonyms with (gender, age). Overall, there were 15 youth users (mean age = 15.73 years) and 4 expert users (mean age = 28.25 years). The expert users' post survey consisted of three Likert Scale questions and seven open-ended questions, while the youth users' post survey had 14 Likert-scale questions and two open-ended questions. Both expert and youth participants were able to explore the various features of Kart Creator, as well as build and program a custom electro-mechanical device. Our pre-survey results showed that the experts were mostly familiar with live programming tools (M_E= 3.25, m_E = 2.5, σ_E = 0.473) and had ample experiences in physical computing devices and electronic circuitry with **high confidence** on such devices. Experts stated that the system met their expectations (M_E = 4.5, m_E = 4, σ_E = 0.500) and that Kart Creator does a good job overall of helping build electro-mechanical devices (M_E = 4.75, m_E = 4.5, σ_E = 0.433). However, we asked the Experts where young students might struggle when using Kart Creator, to which Xander (Male, 24) stated, *"When things don't work out, they may not be able to find the problem easily using the current Blockly UI. Especially when the function has more variables, and logics,"*. He also said, *"For experts, the provided functionalities may be limited. And when things turn to be complex, blockly may not be a good way to program,"* when asked if Kart Creator is attractive/usable by novices and experts. Our current focus is novice users in our target age range; thus, the feedback received from experts motivated us to proceed with the youth pilot given the aforementioned study modifications.

The youth participants were not very familiar with live programming tools (M_Y = 2.133, m_Y = 2, σ_Y = 0.247). Their lack of experience (if any at all) in those areas was seen in their **low or moderate confidence** on such devices. Participants reported to us that the pseudo-code helped them understand the blocks more (M_Y = 6.2, m_Y = 7, σ_Y = 1.166) and that they found it to be clear and easy to follow step-by-step (M_Y = 6.33, m_Y = 7, σ_Y = 0.943). It is also worth noting how adept the youngest participants in our study were at completing the coding challenges. Our original target demographic was 14-18 years (high school age students [94]); however, when advertising the study, parents asked if their younger children could also participate. This motivated us to investigate how the different age groups might respond to and interact with the technology. After analyzing the results, we found no noticeable differences amongst the varying age groups, making it suitable to combine our sample of 11-18 year old participants in the study. The kARtridge was found to be easy to connect external electronics to (M_Y = 6.4, m_Y = 7, σ_Y = 0.712), which made participants happy that they weren't required to have prior knowledge of electronics to use it (M_Y = 6.067, m_Y = 6, σ_Y = 0.772). Kart Creator was also highly rated amongst youth, stating that the on-screen simulator helped them visualize what would happen when the code was uploaded to the kARtridge (M_Y = 6.7, m_Y = 7, σ_Y = 0.718). In fact, participants said programming their devices with the blocks was easy (M_Y = 5.93, m_Y = 7, σ_Y = 0.929) and that they would like to use Kart Creator to create something more complex (M_Y = 6.33, m_Y = 7, σ_Y = 0.869). We also found that most youth participants (N = 12) used the electronics that were presented to them during the coding challenges as opposed to exploring the other devices that were present (e.g. 360^o servo, LED matrices, sound buzzer, etc.).

3.1.8 ***Outcomes:*** Overall, we found that Kart Creator and our electronics toolkit helped eliminate some unnecessary barriers to physical computing and for creating complex electro-mechanical devices. For example, Erica (Female, 14) stated in her post study survey that *Kart Creator is different from other programming software because the model on the screen helps visualize functions without needing to actually carry them out.* Erica stated that she was "moderately familiar" (Likert 3/5) with live programming, but not familiar at all with physical computing devices, electronics, or electro-mechanical devices. Erica's unfamiliarity in these areas highlights the benefits of the system in simplifying the process of creating such complex systems. This experience with the youth through the pilot also enlightened us of several key modifications that we aimed to make with the next version of the MicrokARts system. First, programming AR-IoT interactions should be as easy as programming their devices. Other systems like StoryMakAR [26] create these interactions using a separate interface, but also comment on the disjointed workflow as a result of separate interfaces. Thus, we propose using Kart Creator to author AR-IoT interactions. Next, we want to encourage users to use/learn how to use servo motors since adding it as a coding challenge did not encourage them to use it. We hypothesize that by fabricating a "claw" attachment with the servo motor, participants will be more likely to utilize it for their MicrokARts. Finally, we want to replace one of the coding challenges in the pseudo-code with different sensors and actuators to introduce users to more electronic components and encourage greater diversity in kart design.

3.2 Design Goals

Given the results of our pilot studies, we devised a plan to develop a system that utilizes robust digital-physical interactions. and are ive: Home Circuit is one of the closest examples to the MicrokARts system because of its digital-physical interactions [61]. The problem, however, is that this Nintendo game does not give the user the power to create their own experience and are forced to only have the experiences that are provided to them. To understand the affordances and limitations of this technology, we acquired two Mario kits to study and play with, which helped us motivate our design and further understand the user experiences. After using this system, along with the results from our pilot studies, we developed the following design goals for MicrokARts:

> **[DG1] Open-Ended:** Our system should be open-ended, enabling children to explore options to customize their creations and iterate on them if they desire. MicrokARts has options for customized electronics and AR-IoT interactions through its block-programming interface and electronics toolkit,

[DG2] Low Floors, Wide Walls, High Ceilings: MicrokARts should not be difficult for children to use, should support a wide variety of projects, and be robust enough to scaffold on top of its most basic features. MicrokARts offers plug-and-play electronics, block programming, and live programming to achieve this.

[DG3] Increased Engagement: MicrokARts effectively engages its users through the design, build, and play phases of its use. We designed MicrokARts to be hands-on and dynamic by utilizing the aforementioned barrier reducing technologies, and

[DG4] Quick Iterations: Prior systems may require many steps to get from the initial design of an electro-mechanical device to playing with it. In contrast, MicrokARts offers a simplified way to create modular code that can be compiled and uploaded to our electronics board quickly and easily (see Section 4).

It is standard for new systems to have design goals based on prior knowledge, research, and studies; however, there still exists a question of *why* these goals are desirable for a system like MicrokARts. Olsen [63], Myers, et. al. [56], Greenberg [31], and Ledo, Houben, Vermeulen, et. al. [48] investigate why HCI researchers design new toolkits and explain that, in summary, new toolkits make it easier and faster for users to author complex interactions (DG2, DG4), enable new users to generate new interactive solutions with the system (DG1, DG2), and remain reminiscent of existing systems while building on their foundations (DG3). Thus, MicrokARts' design goals are crucial in developing a system that delivers on our stated contributions (section 1). In the next section, we describe the final MicrokARts system.

4 SYSTEM OVERVIEW

Motivated by prior work (Section 3) and our pilot studies (Section 3.1), we created the MicrokARts system based on the design goals we developed in the previous section. MicrokARts is an end-to-end system that enables children to create a kart, powered by Augmented Reality and connected via WiFi to a network of other karts that can interact with one another in a dynamic AR-IoT environment. MicrokARts is designed to support children in their creative endeavors by offering plug-and-play electronics, a block programming environment with a live programming simulator, and a simplified web interface and mobile application that connects the MicrokARts and smart phones together. In the following subsections, we describe the MicrokARts system's hardware and software interfaces.

4.1 Hardware

The development of the hardware for the MicrokARts platform was centered around facilitating the interaction of children with the handling of electronic devices, in order to initiate cognitive development for entry into technology at an early age. MicrokARts targets children ranging from 9 to 15 years old, and hence, the design and development of its hardware was done to make the system easy to understand and resistant to electrical shock, all whilst enabling quick, easy, and safe replacement or exchange of parts. There are currently several platforms on the market with didactic approaches that fulfill the above functions [41, 55, 84]. In contrast, MicrokARts' hardware presents an open structure that allows direct interaction with the user through the kARtridge PCB, facilitating the connection of any device available in our Electronics Toolkit (see Figure 4). This strength allows MicrokARts to be a potential and viable option for introducing children to electronics, programming and technology.

The kARtridge PCB uses several design methodologies to achieve its intended ease-of-use design. Gestalt's principles of perceptual organization were used to optimize the position of the input header pins, and to easily and intuitively distinguish between the different types of input pins [14]. Various other PCB based toolkits on the market allow for easy connection to the PCB, but also require complicated wiring diagrams to successfully follow programming and design lessons [3, 84]. kARtridge comes equipped with 5 output, 2 input and 3 I2C ports. Sensory devices connect to the input ports, whereas LEDs, and actuators such as motors and servos, connect to the 5 output ports. By providing the capability to connect to I2C devices through the 3 I2C ports, we enable a multitude of modern input and output devices to interface with the kARtridge PCB.

Because our electronics toolkit has multiple components with high power consumption, a significant effort was devoted to designing an effective power delivery system on the kARtridge PCB to endure varying load. Power is delivered to the rotary encoder, ESP32 and OLED display with the help of a 3.7-volt Lithium Polymer battery, while two 9-volt Lithium-Ion batteries uniformly supply power to the sensors and devices connected to the I/O ports.

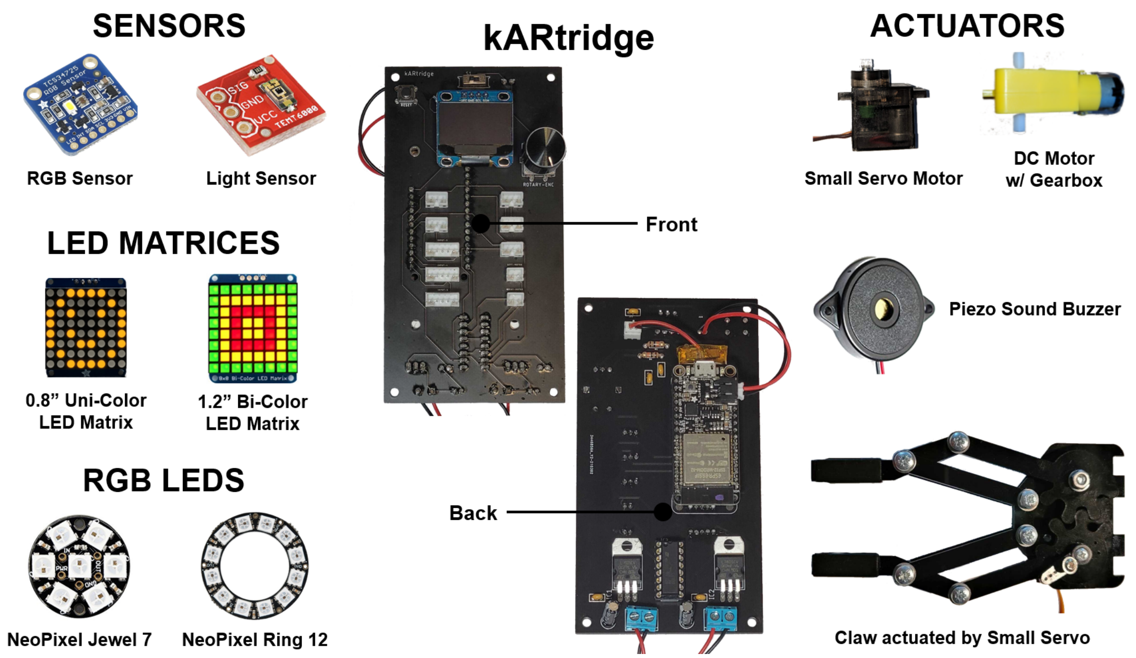

Figure 4: *Our Electronics Toolkit has a plethora of sensors, actuators, and LEDs for users to take advantage of when programming. These electronics are accompanied by a plug-and-play PCB (kARtridge), which has several I/O ports to plug in the electronics. Finally, we designed a small claw that can be used to grab objects in the field and create dynamic games during play.*

4.2 Software Implementation

The MicrokARts system comprises multiple complex moving parts. Each section of the system has been designed to ensure that all parts are synchronized automatically without the user worrying about missing any steps or breaking any unfamiliar technology. Long and arduous set-up processes tend to be confusing and hard to stay engaged with if they don't relate exactly to the system the user is wanting to explore, especially for youth. Using various cloud technologies, our system is able to track what devices are active, who they belong to, and how they will interact with other devices.

4.2.1 Desktop Interface: The web interface for MicrokARts is where users are able to interact with their custom developed MicrokARt. Upon loading the web page, users are greeted with a QR code to sync the mobile interface (section 4.2.2) and the custom kart on the network. Once scanned, the user sees a video feed of their kart, along with various keyboard bindings for controlling the kart. Users can activate their functions programmed in Kart Creator (via automatically mapped keys), change the speed of their kart, and shoot AR darts from this screen.

4.2.2 Mobile Interface: The core component of the MicrokARts system is a phone application that runs on an AR-capable smartphone. We built the application using Unity's latest cross-platform ARFoundation framework alongside Photon's Unity™ Networking package. The application provides users a way to give their MicrokARt a set of Mixed Reality "eyes" [67, 86]. Users can sign into our online multiplayer network and scan the QR code presented on their desktop interface. The software will automatically pair the desktop, phone, and desired MicrokARt on the network simultaneously. The phone pings our various databases to send user kart information to the desktop interface, as well as establish a UDP connection to the correct board. Since our system relies on kart interaction, the AR content on the screen needs to be synchronized in the correct 3D space on all phones. For synchronization, we use Cloud Anchors, a feature in Google's ARCore™ SDK [29]. ARCore provides a coordinate system local to the network's host device, based on unique feature points scanned in the real world, and sends that coordinate system to the other devices. The synchronization of coordinate systems allows a virtual item spawned on one phone to spawn local to the cloud anchor on the rest of the phones. A shared coordinate system in the physical space allows the karts on the network to have a co-located experience when placed in the same environment.

4.2.3 Debug Mode: An additional component of the MicrokARts system is a debug mode that is present in the phone application. This mode allows a user to test all of their custom functions on the kARtridge PCB. The application accesses Firebase to receive the list of user functions along with the IP address of their specific kARtridge PCB. Upon selecting the debug mode, the user can toggle through all of their functions and can activate them in a few unique ways utilizing the various user interface options (Figure 5 depicts Debug Mode with the Action button UI element selected). The features in debug mode allow users to test their solutions beyond using Kart Creator's simulator and helps develop their debugging and problem-solving skills.

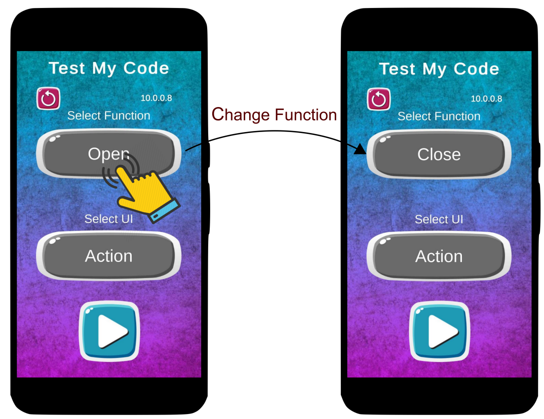

Figure 5: *Our debug tool accesses our Firebase Realtime DataBase to obtain the functions that were written by the users. After obtaining the IP address and list of functions, users can test their code before docking the phone and playing with their MicrokARt.*

4.3 AR-IoT Interactions

One of the challenges faced during development of the system was determining how users could create their own AR-IoT interactions in a quick and intuitive manner. Our solution to this problem is through the inclusion of AR-centric code blocks, so that users can create functions specifically for AR-IoT interactions. These functions make it clear to users in a visual way that if their MicrokARt interacts with a specific AR element, then the code in that function will execute for their electrical components. Currently, our system supports a virtual dart that users can fire from their kart in an attempt to hit other karts and trigger special events. Once the event triggers are programmed in Kart Creator and uploaded to the board, activation just requires the user to get hit with a virtual dart from another player. The darts are spawned within the Cloud Anchor coordinate system (Section 4.2.2), which is how a virtual dart shot from one physical kart can hit another physical kart. When hit, the kart will physically stop, and perform whatever custom function the user programmed with the AR block in Kart Creator. Because our overall workflow uses visual functions, AR-centric code blocks, and cloud anchors to connect the physical world with AR components and synchronize them between the phones, the methodology for AR-IoT interactions is expandable.

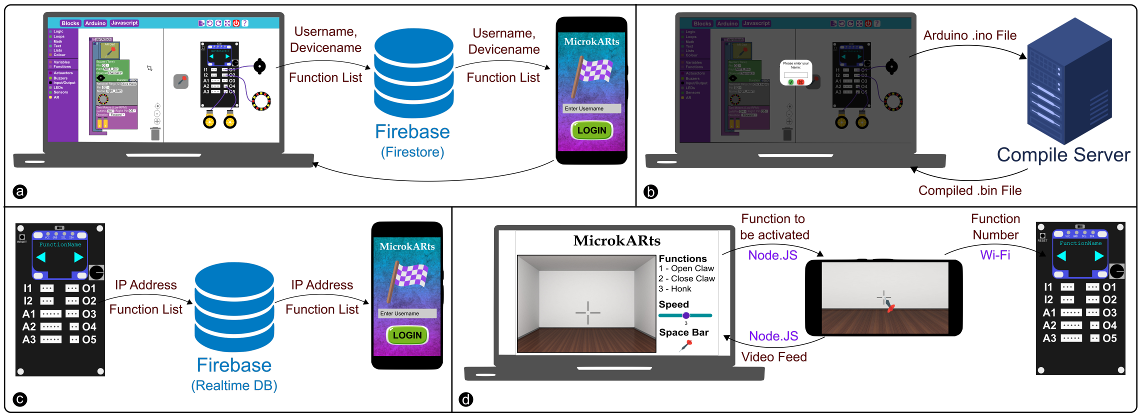

Figure 6: *A breakdown of the system's back-end where (a) the user generated code is automatically saved to Google's Firebase service, where it is hosted in the cloud via Firestore. This information becomes accessible to users when they open the MicrokARts mobile application. (b) Their code is compiled remotely and uploaded to the kARtridge via our Over-The-Air WiFi updating process. (c) The kARtridge automatically sends its IP address and all functions the user created to Firebase's Realtime Database, which is retrieved by the phone when the user logs in. Finally, (d) the user scans a QR code to establish a connection between the phone and the computer through which the AR video stream and list of functions is sent and displayed for the user to see.*

4.4 System Workflow

Our system explores how youth are able to program micro-controllers and have them interact with other physical micro-controllers in an AR-IoT environment. Here is an example of how to use the system from start to finish.

The first step is visiting Kart Creator to design and program their MicrokARt. Users have access to the full suite of hardware mentioned in Section 4.1 to program their functions with. Once a hardware block is dragged onto the screen, users can customize how they want it to function. Each hardware component shows up on the other side of the page with the simulator tool. Users then test each of their functions, seeing exactly how all of their code performs on the board before uploading the code. Users save their device and send it to our remote compiler which will send back a BIN file download to their computer (Figure 6b). This file is then uploaded to a kARtridge via our Over-The-Air uploading page, and when the board restarts, it is then ready to be physically customized to the user's liking.

Users turn on their kARtridge, open the MicrokARts app on their phone, and open the MicrokARts website on their computer. The phone application will pull a list of users from our database, and users can select their name and sign into the application (Figure 6c). Users can then create a room for everyone to join or join an existing room. The website will present them with a QR code that users can then scan with their phone to begin streaming the video feed from the phone onto the computer. Users are then prompted to attach their phone to the 3D printed phone mount on the front of their kart. When the phone is locked in, the kart is now fully controllable via the keyboard on their computer (Figure 6d). Each of their custom functions show up next to the number they need to press to activate it. From there, users can play with their karts however they please. By pressing the Space Bar, users can shoot virtual darts from their karts. If your kart gets hit by a virtual dart, your car will stop for a brief second, and if a custom AR interaction was programmed with the Virtual Dart block in Kart Creator, that function is executed as well.

5 YOUTH USER STUDY

We evaluated MicrokARts in seven workshops with 22 total participants. Each workshop lasted approximately 2 hours and participants' ages ranged from 9 to 15 years. Driven by the learning theory of Constructionism [34], our aim for this user study was to evaluate the MicrokARts system by answering the following research questions:

RQ1 To what extent does MicrokARts serve as a design space for youth to create dynamic AR-IoT environments?

RQ2 What learning opportunities does MicrokARts provide?

To answer these questions, we sought a sample of children to get their feedback on our approach of AR-IoT interactions through physical computing, and to gather data on how children interact with the various sub-systems within the MicrokARts environment.

5.1 Participants

All of our participants were recruited through postings on our social media pages and a local elementary school located 20 miles outside of a large Midwestern University. Informed consent regarding the study was received from both parents and students. There were 22 participants (6 male, 13 female, 3 prefer not to answer) across seven total workshops. The study lasted approximately 2 hours, and at the conclusion, each user was offered compensation in the form of a $20 Amazon gift card. Before starting the study, participants were asked to fill out a pre-study survey asking for demographic information, as well as their experience level in the areas of Augmented Reality, Electronics & Circuitry, Programming Physical Computing Devices, and Designing Electro-Mechanical Devices. Questions regarding their confidence levels in each of the surveyed fields were asked to compare and contrast the differences in self-reporting amongst users. All names of students are reported as pseudonyms and the results from the pre-survey are reported in Figure 8.

5.2 Workshop Setup

All workshops took place in a large, open room with two small, square tables at which the students were sitting. A maximum of 5 students were allowed to sign up for a single time slot to reduce the number of people in the room during the study. Participants were asked to sit at tables in groups of 2 or 3 (depending on workshop size). The participant groups are reported in Table 1. Proper social distancing guidelines were enforced and all participants and parents were asked to wear masks during the study. Each workshop participant had a computer, phone, and MicrokARt to use for the study. All surfaces, electronics, and other materials were wiped with cleaning supplies before and after each study. [1]

5.3 Methodology

Our 2-hour study consisted of a pre-workshop survey, a tutorial exercise, four coding challenges adapted from our pilot study (Section 3.1, and individual post-workshop interviews with the participants. The first-author facilitated the workshop, while other researchers wrote observational field notes during the study on an observation form designed by the research team. A copy of the observation form and instructions packet can be found in the Supplementary Materials. As stated in section 3, MicrokARts is built to achieve our four design goals (Open-Ended; Low Floors, Wide Walls, High Ceilings; Increased Engagement; and Quick Iterations). We see potential benefits for the system to support students' problem solving (DG2), engineering (DG4), and project management skills through complex, open-ended projects (DG1, DG3). The evaluation study, described below, was conducted to evaluate to what extent these goals are achieved.

Study #	Participants per Group
1	Group 1: Rodney, Billy
2	Group 2: Brittany, Missy Group 3: Alaina, Charlotte
3	Group 4: Lucy, Matt
4	Group 5: Blaine, Sam, Molly
5	Group 6: Joy, JoAnn Group 7: Harriett, Autumn, Katy
6	Group 8: Ray, Brian Group 9: Payton, Ella
7	Group 10: Jasmine, Erica

Table 1. *Students were separated into groups of 2-3 during the study. Participants are reported here by which participants were in each groups.*

[1] These precautions were taken in order to keep participants and researchers safe as a result of the COVID-19 Pandemic.

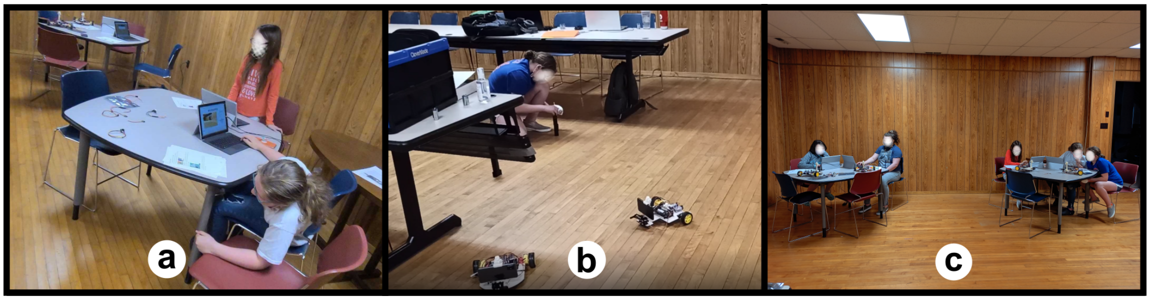

Figure 7: *The setup of our MicrokARts user study. (a) Participants playing with their MicrokARts at the beginning of the study, (b) participants resetting the blocks so their friends can use the claw to move them from one side of the room to the other, and (c) the study space depicting Joy and JoAnn at one table (left) and Katy, Autumn, and Harriett at another table (right).*

At the beginning of the workshop, we uploaded the tutorial code onto each of the MicrokARts. Participants were instructed to navigate to the MicrokARts web app via the Kart Creator website to generate their QR code. Participants then opened the MicrokARts app on the cell phone, scanned the QR code, and streamed the phone's camera feed to the computer. Participants then docked the phone onto the phone mount and were allowed to drive around. During this tutorial, participants were introduced to various elements of the interface: (1) using the arrow keys (or ASWD keys for left-handed users) to navigate the MicrokARt, (2) using the number buttons to activate various programmable functions, (3) changing the speed of the MicrokARt, and (4) pressing the space bar to shoot AR Darts at others.

After about 10 minutes of play, participants were instructed to navigate back to the Kart Creator website and were given a brief tutorial. Participants were shown how to drag and drop blocks into place, how to create functions, how to view the generated Arduino and JavaScript code that they were making with the blocks, and how to use the simulator to test their code. After the tutorial, participants were allowed to begin working on the Coding Challenges (CC). After each student was finished with their CC, the workshop facilitator uploaded the code to each MicrokARt. In order to save time, the code was pre-compiled for each CC before the study. Some CCs were included to introduce them to our two debugging tools: (1) Laboratory Mode, and (2) the "Test My Code" function in the MicrokARts app. The last CC taught students how to design AR-IoT interactions by using the AR Dart block. After finishing CC 4, students were asked to code whatever electronics they wanted to activate when their MicrokARt gets hit with an AR Dart, as well as functions to open and close their claw. Their final code was compiled and uploaded to their MicrokARts and participants were once again instructed to open the MicrokARts web app to generate their unique QR code, scan, and play. For the final activity, the participants were asked to play a game where the workshop facilitator placed small blocks on the ground that the participants were asked to drive around and pick up from around the room. AR Darts were used to stop their opponents from reaching the blocks successfully.

Immediately following the workshop, we conducted semi-structured interviews with the students. During the post-study interview, we asked students about their impressions with MicrokARts and its features, after having used the system, what challenges they ran into while using it, and what they learned as a result of using the system. The full set of interview questions can be found in the Supplementary Materials.

5.4 Data Collection & Analysis

All 22 interviews were recorded and fully transcribed. Interviews were conducted in their groups unless the students specifically asked us to interview separately. Data from both the observation forms and interviews were analyzed by creating affinity diagrams by using a deductive thematic coding approach based on grounded theory to cover the range of experiences and narrow enough to evoke and explore students' specific experience with MicrokARts [52]. Grounded theory provides "sensitizing concepts," such as self-concept, self-identity, communication, and definition of the situation [9, 87], allowing researchers to code and develop more refined and precise concepts.

The affinity diagrams were generated using a Miro board (http://Miro.com) to help consolidate and visualize the observational and interview data into one place. From these affinity diagrams, we held joint data interpretation sessions among the researchers to extract common themes. Additionally, one of the students' interviews were analyzed as a group to help researchers further understand the process. Afterwards, participant data was divvied up amongst the researchers, and each researcher individually extracted the data of the students for which they were responsible. Finally, the researchers met again to analyze the data and come to a consensus on any interpretations.

5.4.1 ***Deductive Thematic Coding:*** Our initial analysis of the qualitative data was to identify a set of codes with which we could derive some common themes. The codes were developed deductively and the analysis team met virtually via Zoom to begin developing the codes. During our discussion, we mutually agreed that our design goals for the system (section 3) were imperative to the development of the codes. Thus, we used a bottom-up qualitative coding approach to examine the interview transcripts and observation forms, while keeping our design goals in mind. Our final set of codes are as follows: (1) Exploratory/Inquisitive/Curious (DG1, DG2), (2) Making Connections to Prior Observations and Experiences (DG3), (3) MicrokARts Fosters Creativity in Some Participants (DG1, DG3), (4) MicrokARts Has Potential to Develop Problem Solving Skills (DG2, DG4), and (5) Participants' Interpretation of Their Own Learning (DG1, DG3). After mutually generating the codes, we began to search the interview and observation data to find excerpts that fit each category. Data that did not fit any of the codes did not fit the scope of this paper and were omitted here, but will be analyzed in future research. The data was divvied up among the researchers to assign to the various codes, then the analysis team reconvened via Zoom. After some discussion about the placement of all the data and mutual agreements regarding the code assignments, the team began to analyze the data. Our data analysis is presented in the next section.

6 RESULTS & DISCUSSION

Overall, participants had positive impressions of the system after completing the study. We collected both quantitative and qualitative data from the participants in the form of survey results, observations, and interviews.

6.1 Coding Analysis & Discussion

In this section, we will discuss the results from our MicrokARts study. We present (1) the codes that were developed by the researchers, (2) any emerging themes that arose as a result, and (3) a set of recommendations for developers of future systems like MicrokARts grounded by our analysis. All participant pseudonyms are reported below as: Pseudonym (Sex, Age), with the exception of those who gave No Response (NR) when asked for their sex. As shown in Figure 8, most of the participants had absolutely no experience with any of our subject areas (indicated in red).

6.1.1 ***Exploratory/Inquisitive/Curious:*** Throughout the studies, participants displayed their curiosity about the system by asking questions and exploring the various features of the system outside of the structured Coding Challenges (CC). Observers noticed participants asking a number of questions, some of which were related to the way that the system worked (Systemic Questions or SQ), while others were related to the underlying concepts that they were asked to use during the Coding Challenges (Conceptual Questions or CQ). An example of an SQ was, *"Where is the function block located?"* while using Kart Creator; whereas, a CQ was, *"Why does the light not turn off?"* when completing CC2.

Observers noticed most participants asked clarifying SQs during the study, which is an expected result of anyone learning to use a new system (especially children). However, there were several participants who wanted to know more about the software as they were using it. Sam (Female, 11) noticed the Unity logo pop up on the screen when she opened the MicrokARts app on her phone and originally thought that Unity had made the app. When we explained to her that

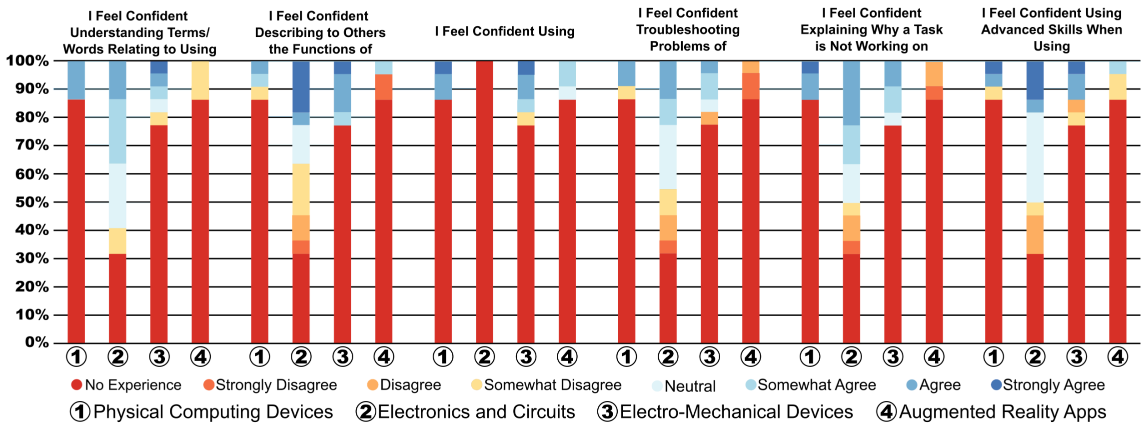

Figure 8: *Pre-Survey Results from our User Study reported by percentage of answers given. All questions are based on a 7-point Likert Scale (1 - strongly disagree, 7 - strongly agree). Most students had no experience in many of the areas (indicated in red). Questions were adapted from Eastin and LaRose's Internet Proficiency Scale. See Appendix A for more information.*

we made the app, she began to probe deeper about how we did it, expressing interest in the process of making the app. Molly (Female, 12), who was in her group and overheard this conversation, noticeably started to play around with the features of the system, such as Laboratory Mode on the kARtridge, the mobile app, and the Kart Creator simulator. At the conclusion of the workshop, Molly asked one of the observers what their job title was at the University and exclaimed to them that she *"wanted to do what [he] is doing when [she] goes to College because this is so cool."*

During the interviews, participants also asked some CQs to the researchers. Billy (Male, 11) was very curious about the system after viewing the Arduino code that he had written for his functions. Noticing that the LED Backpacks are programmed via binary, he inquired: *"Yeah so are those the numbers that like the robot will read to know the code?" [Student is pointing at the computer code on the screen.]* After having the concept of binary explained to him, he began making connection by stating *"Okay, so like one means on and zero means off? So, like in a movie there would be like a lot of 0's and 1's? Oh, okay. Because the computer like doesn't read the words, just 0's and 1's?"* Overall, participants were enthusiastic about exploring the system and showed signs of developing their own mental models (i.e., the movies) of what they were learning (i.e., with MicrokARts) through their actions and interviews.

6.1.2 ***Making Connections to Prior Observations and Experiences:*** Another notable observation from our study was that participants made comments stating that the MicrokARts system was reminiscent of activities that they had participated in both inside the classroom and during after-school activities. Two participants noted that it reminded them of in-school programs like Project Lead the Way [68]. Twenty of our participants indicated that they had live programming experience in some form in their pre-study survey, and commented about their experiences and knowledge during the workshops. However, in their post-study interviews, four students conveyed their thoughts on how easy it was to create their MicrokARts. Blaine (Male, 9) went as far as to say *"Well, I like doing the code and the robots because at school when we do coding...it's kind of like games, but differently, and I kind of don't like the one at school but then when we did [MicrokARts], it was amazing and I liked it a lot."* Quotes like this indicate to us that the children really enjoyed the openness and ease of MicrokARts. We attribute this to the number of components that users can choose from, and their ability to program them without having to formally "learn" anything. This student (and many others) suggested that they did not feel like they were learning, but were still able to program a MicrokARt. Although, in reality, there are underlying concepts (e.g., functions and conditional logic) that they were utilizing throughout the study. The observation on learning suggests that our system is designed well and students are learning while actively engaging in the process [77].

Alaina (NR, 12) mentioned in their post-study interview that they have experience using JavaScript and how they use it to make websites. Notably, they conveyed to us that they felt that seeing the JavaScript code that was responsible for executing the code on the simulator helped them make connections to previous programming experiences they have had. *"I learned a bit more about JavaScript, I like to use it in other situations, And I learned a bit more about it. Sometimes we'll go on - I forget what it's called, but I can use JavaScript there, and program, like a little website kind of thing."* Such connections to prior experiences aid in our analysis of the MicrokARts system because it allows us to further understand how students are exhibiting constructionist behaviors as a result of its use.

6.1.3 ***MicrokARts Fosters Creativity in Some Participants:*** One of our design goals from Section 3 was to make the system open-ended to encourage iterative play and promote a broad range of experiences. We found that students were generally not being very creative with the electronics at their disposal. In fact, when asked to create their own AR-IoT interactions with the AR darts, all participants used the buzzer for the AR Dart interaction, but some decided to use **only** the buzzer (N = 3) and most used the buzzer and only 1 additional electronic device (N = 16). Most of the participants made a very simple program similar to what is shown in Figure 9 where the buzzer sounds and one of the 4 LED devices will display. One exception to this was Ray (Male, 10), who used the Buzzer and Small LED Matrix on his MicrokARt, but decided to make the Buzzer play a short song before turning the LED on. Presumably, we attribute this lack of exploration and creativity to be due to the participants' excitement to play with the cars again before the study was over; however, we acknowledge that there could be a number of reasons for this to occur.

On the other hand, three participants used more than those two devices to design their AR-IoT interactions. Brian (Male, 10) is the only participant that utilized any type of sensor in his function by first making the Buzzer beep for 2 seconds, and then turn the LED on if the sensor sensed something red. Rodney (Male, 10) wanted to make a light show happen when his MicrokARt got hit with an AR dart, so he added one of every LED to his MicrokARt except the Neopixel Ring (which he originally added in his code, but had to remove due to only having 3 output pins for those devices). Rodney commented on this in his post-study interview. When asked what challenges he ran into when programming the electronics on the computer, he stated, *"I couldn't put it [the Neopixel Ring] on four or five because they were connected to [other devices]."* Most notably, however, Matt (Male, 9) made one of the most complex interactions with the Buzzer and Large LED matrix by using a for loop to create a song and light combination. When the researchers inspected his code, we asked him if he could explain why he used that block, to which he responded *"I just saw it and thought it was cool and tried it on the computer and it worked."* We also asked him if he had prior experience with using loops beforehand, to which he responded in the negative. Once Matt was finished with his code, his sister

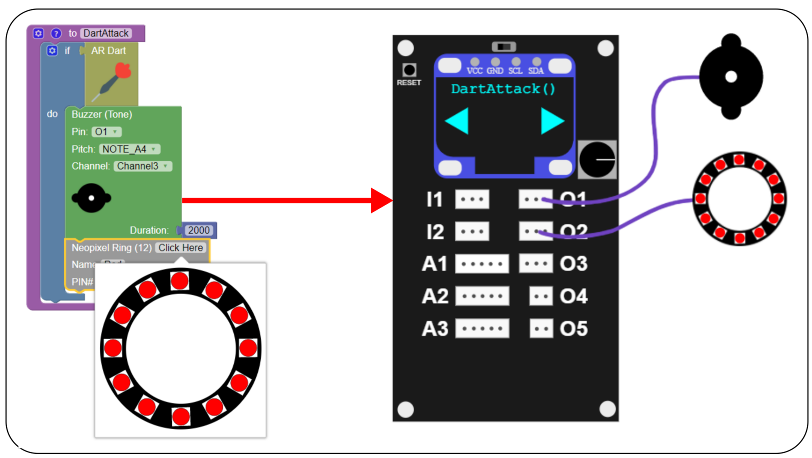

Figure 9: *Example of a lot of our participants' final code for AR-IoT interactions.*

Lucy (Female, 10) saw that he had made his MicrokARt more complex and asked the researchers if she could redo hers like Matt. These behaviors are the type of creative endeavors that we aim to foster with the MicrokARts system.

6.1.4 ***MicrokARts Has Potential to Develop Problem Solving Skills:*** As we mentioned in Section 3 of this paper, it is imperative that toolkits like MicrokARts should foster problem solving skills in children [6, 33, 64]. Several of the children were able to complete all of the Coding Challenges with little-to-no input from the researchers (N = 16). We describe this input as asking only 3 or fewer questions throughout the duration of the study.

A few participants needed help despite having access to the instructions that we gave them for the study. Again, we wrote the instructions such that it would scaffold on top of their learning of the system; the instructions initially introduce the name of the block and where it is located, but then simply asks you to use the block in subsequent Coding Challenges. The students that had the most issues with this either indicated that they had very little experience (N = 5) or no prior experience (N = 2) with block-based live programming, or were not reading the instructions fully (N = 4). We found that, although the majority of the students were able to follow the instructions, complete the Coding Challenges, and be successful at the open-ended project at the end, having the instructions on a separate medium than the code may actually be the root cause of this confusion amongst the students. This was also an observation noted from our youth pilot studies where participants were not fully reading the pseudo-code. Future iterations of MicrokARts will create a tutorial system that integrates the Coding Challenges into the website rather than having them separated.

Several students also had issues with managing the pins that their devices were being plugged into. For instance, Katy (Female, 9), Autumn (Female, 10), and Harriett (Female, 10) were frequently forgetting that they needed to change pin numbers for their devices in Coding Challenge 3, as well as in their AR-IoT interaction. Other students discussed their experiences with problem solving in their post-study interview like Brittany (Female, 13) who stated *"The circuits didn't go together. And then you had to find out how they went together,"* and Alaina (NR, 12) who enjoyed the challenge of using the system: *"I like that it actually made me have to think, because usually I come by things easily. But with this, that wasn't the case... not to where it was frustrating necessarily."* Our design goal of "Low Floors, Wide Walls, High Ceilings" helps attribute to this outcome by allowing users to come up with a unique solution to the problems they encounter.

6.1.5 ***Participants' Interpretation of Their Own Learning:*** Throughout the workshops, we noticed that the students were becoming increasingly better at operating their MicrokARts, programming, and using the debugging tools that we developed for them to use. What was most notable to us was that the participants were able to grasp some foundational concepts of programming, such as the purpose of functions, but were not actually aware of what they were learning. Concepts like modularized code, feedback, control, and connected devices are all a large part of how the MicrokARts system works, but some participants still reported that they don't know what they learned (N = 4), or gave very generic answers like Erica (Female, 11) who said *"You can actually make the cars do stuff ... and make lights and stuff."* These concepts aren't necessarily black boxes as the users all explored these concepts throughout the workshops in the final activity [74]. So why aren't the participants reporting more details when asked what they learned? This is likely influenced by the lack of time that students had to iterate on their MicrokARts combined with the instructions that were given, which focused on helping participants learn the system, rather than the names of the concepts.

6.2 Emerging Themes

After the researchers analyzed the coded data individually, we reconvened via Zoom to extract themes from the data. Based on our survey results, we noted the following as emerging themes:

6.2.1 ***Theme 1 – Interest-Driven Design and Making:*** The MicrokARts system piqued the curiosity of the children because of the electronics toolkit that we provided to them. We found that every child enjoyed using the electronics that invoked one or more of their senses (e.g., the LEDs and sound buzzer). We noted on our observational study forms how many times students were verbally or physically enthusiastic about using the electronics throughout the study. Every single participant exhibited this behavior at least once, while half of the participants (N = 11) showed the behavior five or more times in a single session. As previously mentioned, 16 participants used the buzzer and one additional electronic device, and three participants used more than two electronic devices. Of those 19 total participants with two or more electronics, 15 of them used the claw as one of their electronic devices. We believe that this shows participants' interest in customizing their devices beyond just the basic functionality of a kart (i.e., driving around without any additional functionality). However, seeing as the claw had some utility that the participants really enjoyed, it may be worth studying how students engage with pre-designed fixtures like the claw versus designing their own fixtures. Finally, even parents were enthusiastic about MicrokARts captivating their children. The mother of Rodney and Billy was present for the post-study interview, during which she stated, *"Well I like it cause it sparked an interest in him, I can tell. I love that, and anything to rescue him academically and give him different ideas of interest outside of school. I thought that he would love this [points at Rodney]. And it looks like he is, now that he can actually do it."* The mother also explained to us that they have some electronics kits at home that were similar to Micro:bit, but they have *"more to do with circuitry"* than MicrokARts. Based on the mother's response, Rodney (Male, 9) had not responded to the other kits that he had been exposed to in the same way that he had responded to MicrokARts, especially when using the electronics.

6.2.2 ***Theme 2 – AR-IoT Interactions Generate Engagement:*** The familiarity of MicrokARts to commercial products that kids use both in-school and in extra-curricular activities is a strength of the system. As stated earlier, most participants had at least some experience with block-based live programming (N = 20), so they were able to leverage some prior knowledge while using Kart Creator. Additionally, less than half the participants (N = 11) stated that they were at least a little familiar with the Mario Kart Live: Home Circuit video game [61], while three participants stated they had experience using AR apps in the past. We believe this discrepancy in reporting is due to the children's lack of understanding of the technology. However, MicrokARts' AR-IoT interactions were new and exciting to the children as the majority (N = 19) cited these interactions as what they liked about the system in their post study interviews. We also observed all the students being excited about programming and using the darts to create their interactions at least once, while nine participants exhibited this behavior five or more times throughout the duration of the study. This yields promise for future versions of MicrokARts to include more AR-IoT interactions through other AR objects like catapults, kart-seeking projectiles, and more.

6.2.3 ***Theme 3 – Collaborations Among the Children:*** Collaboration is a major aspect of the impact that this system can have on children. The power in AR is that it fuses both the digital and physical realms – which should definitely include users and other connected devices. Although we put the participants in groups during the study, very few of the participants were collaborating with each other while completing the first four coding challenges. However, during the final, open-ended challenge, participants began garnering ideas from one another and even helping when they were having issues (an observation also noted during the pilot studies). AR-IoT interactions also played a major role here during play as participants were grabbing and placing blocks. Some students wanted to compete against one another to see who could collect the most blocks before the study was finished, while other groups wanted to collaborate and help each other. The groups helping each other can be seen in the supplementary video placing blocks around the room and pointing to the screen to help navigate and control the MicrokARts. The groups competing against each other enjoyed shooting darts at each other and stopping others from completing their objective, which was noticeably fun and engaging for them. Thus, we believe that shared AR-IoT interactions motivated participants to collaborate in a very engaging way (DG3).

6.2.4 *Theme 4 – Enhanced Play with MicrokARts:*

Participants aren't realizing that they are learning and applying these concepts while using MicrokARts. As stated in Section 2, Michael Schrage discusses the importance of learning through prototyping iterations [77]. Participants were observed performing several iterations of their final programs before playing during the last activity. The three participants in Group 5 each wanted to go back and change their final programs after uploading their code for the final activity because they wanted to improve it by adding more devices. They had this idea with approximately 15 minutes left in the study and each of them were able to iterate, upload, test, and begin playing again before the end. In other words, the participants had garnered enough prowess with MicrokARts during the study to iterate on their code and play with each other even after finishing all the activities. Moreover, we deduce that opportunities for children to iterate quickly can help them hit conceptual milestones and continue being engaged with the technology they are faced with (DG3, DG4).

6.3 Recommendations for Future Systems

Guided by the analysis of our user study with 22 children (ages = 9–15), we offer the following recommendations for developers who aim to design an AR-IoT platform for children:

(1) **More Exposure, Less Disclosure:** Exposing students to Engineering concepts at an early age is important, but so many toolkits today disclose far too many details for young children to get from concept to prototype without becoming frustrated. We propose reducing user development time to make things work (programming, electronics, etc.) by reducing the amount of prior knowledge required to use the system. This idea falls in line with prior work [6, 64], our design goals (DG2 and DG4), and our emerging themes (Theme 1 and Theme 4).

(2) **All Hands On Deck:** A hands-on platform is ideal for future AR-IoT systems – and sharing the experience of using such a system with others helps improve the overall experience of using the system for users of many ages and abilities. Shared AR experiences coupled with physical prototyping can help pique curiosity/inquisitiveness, foster creativity, and enable learning through hands-on creative play, as is demonstrated with MicrokARts (Theme 3).

(3) **Interact & Iterate:** AR-IoT interactions are a popular topic right now, especially for children; however, authoring these interactions is an under-explored area on which MicrokARts aims to shed light (Theme 2 and Theme 3). We propose a block-programming approach to authoring AR-IoT interactions. Doing so will centralize the programming of all IoT functionality, while encouraging more iterations of interactions.

(4) **Connectors Are Key:** Our participants used low fidelity bonding elements like duct tape to attach their electronics to the MicrokARts. In contrast, we support the idea of using simple 3D printed connectors to help attach the electronics to the IoT device in future systems [10] (Theme 4).

6.4 Summary

In this section, we analyzed the data that was collected during our user study. Researchers convened the first time to develop a set of codes and establish an analysis plan. We developed five thematic codes using a deductive coding schema and the Constructionism Learning Theory as our guiding theoretical framework. The work was divvied up amongst the researchers and the qualitative data was analyzed. We discussed this analysis at length in section 5.4.1. Once the data was analyzed, the researchers reconvened to discuss the results and generate themes that emerged from the data. Finally, we offered 4 design recommendations for future AR-IoT systems. In the next section, we describe the limitations that arose while studying the system, as well as some work in progress for future versions of our MicrokARts system.

7 LIMITATIONS & FUTURE WORK

Acknowledging that no technology is perfect, we discuss some of the limitations of our system that were uncovered during the development and study of the MicrokARts system, as well as some work to aim for in future versions.

7.0.1 ***Phones for Cameras:*** One of the limitations of the system is the need for AR-capable smartphones for our karts. From an accessibility standpoint, this is a cost barrier of the system that we would like to improve. There are other standalone cameras with enough processing power to utilize Augmented Reality that we want to explore in the future. From a usability standpoint, the phone can be heavy, take up a lot of space on the karts, and requires extra effort for participants during set up. Compared to the Mario Kart example, the camera is built-in and ready to scan, which we prefer to integrate a similar camera system onto our kARtridge PCB [61].

7.0.2 ***Latency & Internet Bandwidth:*** Video streaming requires minimum internet speed to be fluid and usable. However, access to usable internet is not yet a standard for the average American household [5, 11]. Because our system is collaborative, internet reliability can hinder streaming quality and user experience. During our studies, we found that having more than three video streams caused significant lag in the entire system (video streaming as well as moving the karts), which presents a problem in scalability. Thus, we plan to explore other options for video streaming, as well as different ways to get connected to the internet outside of WiFi. One possibility is accessing newly constructed 5G cellular towers that have speeds much faster than current LTE speeds [51]. Alternatively, the Starlink program, which is providing satellite internet for areas with little to no competition in the internet sector [44, 81].

7.0.3 ***Security Risks:*** The current design of the system tailors better towards controlled environments with secured WiFi networks. When running the system as intended, all connected PCBs show their Local Area Network (LAN) IP addresses so the system knows which device contains what functions. Accessing our compiler is a direct computer-computer connection which presents security concerns when sending (potentially harmful) files back and forth. Alongside this, all connected phones are subject to the Google and Photon privacy policies when using the network capabilities of Augmented Reality, Cloud Anchors, and multiplayer connections respectively [27, 28, 67]. Users are subject to malicious attacks from lower-scale hackers on their WiFi network to large-scale hackers trying to peek into user data stored on the servers of the aforementioned services [50]. This would, however, be an interesting research topic to explore teaching users the risks of using AR-IoT technology in how it can potentially expose identifying data. A workshop may look like showing users the type of information stored during an AR session, and having a different user 'hack' into the network and start affecting the PCBs and/or the Augmented Reality experience.

7.0.4 ***Co-location & Remote Capability:*** Another system requirement is for users to be co-located as the Augmented Reality interactions with the karts require the karts to be in the same room. However, it may be inconvenient for users be in a shared space to use MicrokARts. Our current system implementation makes it possible to control the kart via the desktop interface in a remote location, which brings to light that the actual limitation is that the karts themselves need to be co-located, not the participants. This brings up an interesting area to consider, in possibly replicating cloud services like Google Cloud run their purchasable Virtual Machines [30], allowing users to connect to a remote computer located within Google and do all of the tasks you need the machine for. Persistent kARtridges could be located in a set space for users to remotely connect to and control the MicrokARts from their own homes. Although this takes away the physical kart customization aspect, it increases accessibility to people who cannot utilize one of our custom PCBs, but still want to use our system.

7.0.5 ***Broader Educational Impacts:*** Computational thinking is a field that does not have a widely agreed upon definition that is suitable for K-12 education due to the complexity of the field [2]. However, Wing has devised a set of characteristics for computational thinking that researchers have generally agreed upon [98], in which she states that, *"Computational Thinking is a way humans solve problems... we use our cleverness to tackle problems we would not dare take on before the age of computing and build systems with functionality limited only by our imaginations."* Grover and Pea take a similar stance in their review of the state of Computational Thinking in K–12 Education by discussing the inter-connectivity of digital devices and systems and their relation to fostering problem solving skills [33]. What is more, in his book *Serious Play*, Schrage discusses the importance of prototyping and iterating to learn [77]. These findings highlight the significance of a system that has both physical and digital components to have an open-ended, iterative nature which can help children cultivate creative solutions to computational problems. We aim to use MicrokARts to evaluate its potential as a learning environment for children. Future studies can be curated with local computer science educators to establish learning goals.

8 CONCLUSION

In this paper, we presented MicrokARts – an AR-IoT platform that enables co-located play for young children to design, program, and play with AR-enabled karts together. The system was designed with customizability in mind to encourage collaborative play through AR-IoT interactions. We presented a set of design goals based on prior work, existing toolkits, and the pilot studies with our initial prototype. We discussed the block-programming simulation interface called Kart Creator, plug-and-play electronics with our kARtridge PCB, and AR app run on a smart phone, and a web-based interface that enables users to interact with each other in a dynamic AR-IoT environment. We reported a study with 22 youth between 9 and 15 years old that showed that MicrokARts can be used by children with experience in the field, as well as those without any experience to create these dynamic AR-IoT environments. We discussed the opportunities for engagement in this area through our emerging themes (Section 6.2 and recommend that future systems incorporate (1) more exposure to concepts and less disclosure of unnecessary details, (2) all hands on deck during play, (3) opportunities for interactions with one another and iterations on their developed products, and (4) connectors for higher-fidelity prototypes. These recommendations will assist users to generate iterative and creative solutions to the problems that arise from collaborative AR-IoT platforms. Moreover, MicrokARts contributes to continued efforts to support youth in building foundational understandings of interactive and creative play through physical computing and Augmented Reality.

9 APPENDIX

A PROFICIENCY SCALES FOR PRE-SURVEYS

During our pilot and main studies, we adapted the Internet Proficiency Scale developed by Eastin and LaRose [19]. Changes were made to be relevant to the field in which the Expert/Youth participant stated they were proficient, but we were careful not to change the meaning of the statement. An example of our adaptation is below, while the rest of the proficiency scales were modified in a similar manner. Each question was evaluated on a 7-point Likert Scale. These questions can be seen graphically in Figure 8.

- Understanding terms/words relating to developing an AR app,
- Describing functions of AR to others,
- Troubleshooting problems of an AR app,
- Explaining why a task is not working on an AR app,
- Using advanced skills when developing an AR app,

- Using an AR app

B POST SURVEY QUESTIONS FOR EXPERT PILOT

During our expert pilot study, we administered a post-study survey to elicit feedback from the participants. With this survey, we aimed to gather some Likert scale information and open-ended responses regarding their feelings toward the initial system and to understand their interpretation of its prowess in helping youth/novice users with creating complex electro-mechanical devices. Some questions were also matched to their interview answers, and is described in section 3.1. We asked our expert participants the following post-survey questions:

- Did the actions required to make the simulator function the way you expect meet your expectations? (Likert)
- In your opinion, what aspects of using the system might young students (age = 14-18) struggle with when using the system? (Open-Ended)
- Do you have any suggestions to make the system more user friendly for this demographic? (Open-Ended)
- Could this system be attractive/usable by both novices and experts? (Likert)
- What features of the system do you believe are missing? (Open-Ended)
- What existing features of the system, if any, do you believe should be omitted? (Open-Ended)
- As an expert, how would you use this system? (Open-Ended)
- In what ways do you believe this system can help in the design of electro-mechanical devices? (Open-Ended)
- Do you think our system does a good job of helping to design electro-mechanical devices? (Likert)

C POST SURVEY QUESTIONS FOR YOUTH PILOT

During our youth pilot study, we administered a post-study survey to elicit feedback from the participants. With this survey, we aimed to gather mostly Likert scale and some open-ended information regarding their feelings toward the initial system. Some questions were also matched to their interview answers, and is described in section 3.1. We asked our youth participants the following post-survey questions:

- To what extent do you agree that...
  - It was easy to connect your motors/sensors to your circuit board
  - It didn't require a lot of training to use this circuit board
  - I believe that the easily connecting electronics helped me quickly bring my physical structures to life
  - I'm glad that I don't have to use a lot of knowledge about electronics to operate them.
  - I liked watching my code running on the computer screen
  - Watching my code on the screen helped me visualize what would happen when I uploaded the code to my board.
  - I think I can use Kart Creator to make something even more complex
  - This system is attractive to me
  - This system is usable by me
  - Using the blocks to program my devices was easy.
  - I feel that I don't have to use knowledge of computer programming to use Kart Creator.
  - The pseudo-code and instructions that were given to me was clear and easy to follow step-by-step.

  - The pseudocode helped me understand the blocks more.
- In your own words, what is different about Kart Creator vs. other programming software that you may have used in the past (if you haven't use any programming software, then please put "N/A").
- In what ways do you think live programming helps with the design of an electro-mechanical device?

D PILOT STUDY INTERVIEWS

- **Grand Tour Question**: Considering everything we did with Kart Creator today, do you like the system?
  - If yes, → What did you like about Kart Creator?
  - If no, → What do you think should be added to make it more usable to you?
- If you were using the system in the future, in what ways would you use it?
- When you programmed your electronics using Kart Creator, what challenges did you run into? How did you overcome them?
- Do you know how the activities we did today could be used practically?

E MAIN STUDY INTERVIEWS

- **Grand Tour Question**: What did you think of the MicrokARts system that you used today?
- Considering everything we did with MicrokARts today, do you like the system?
  - If yes, → What did you like about MicrokARts?
  - If no, → What do you think should be added or changed to make it better to you?
- If you were using the system in the future, in what ways would you use it?
- When you programmed your electronics using Kart Creator, what challenges did you run into? How did you overcome them?
- When you were building your MicrokARts, what challenges did you run into? How did you overcome them?
- When you were playing with your MicrokARts, what challenges did you run into? How did you overcome them?
- What did you learn from using the MicrokARts system today?

REFERENCES

[1] Apple. 2021. Dive into the world of augmented reality. | Apple Developers. https://developer.apple.com/augmented-reality/

[2] 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.

[3] British Broadcasting Corporation (BBC). 2020. Micro:bit. https://microbit.org/.

[4] Ayah Bdeir. 2009. Electronics as Material: LittleBits. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction* (Cambridge, United Kingdom) *(TEI '09)*. Association for Computing Machinery, New York, NY, USA, 397–400. https://doi.org/10.1145/1517664.1517743

[5] Natalie C. Benda, Tiffany C. Veinot, Cynthia J. Sieck, and Jessica S. Ancker. 2020. Broadband Internet Access Is a Social Determinant of Health! *American Journal of Public Health* 110, 8 (2020), 1123–1125. https://doi.org/10.2105/AJPH.2020.305784

[6] Paulo Blikstein. 2013. Gears of Our Childhood: Constructionist Toolkits, Robotics, and Physical Computing, Past and Future. In *Proceedings of the 12th International Conference on Interaction Design and Children* (New York, New York, USA) *(IDC '13)*. Association for Computing Machinery, New York, NY, USA, 173–182. https://doi.org/10.1145/2485760.2485786

[7] C. Cabo. 2018. Effectiveness of Flowcharting as a Scaffolding Tool to Learn Python. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, San Jose, CA, 1–7.

[8] Lautaro Cabrera, John H. Maloney, and David Weintrop. 2019. Programs in the Palm of your Hand: How Live Programming Shapes Children's Interactions with Physical Computing Devices. In *Proceedings of the Interaction Design and Children on ZZZ - IDC '19*. ACM Press, Boise, ID, USA, 227–236. https://doi.org/10.1145/3311927.3323138

[9] Kathy Charmaz. 2014. *Constructing grounded theory*. sage, California.

[10] Subramanian Chidambaram, Yunbo Zhang, Venkatraghavan Sundararajan, Niklas Elmqvist, and Karthik Ramani. 2019. Shape Structuralizer: Design, Fabrication, and User-driven Iterative Refinement of 3D Mesh Models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. ACM Press, Glasgow, Scotland Uk, 1–12. https://doi.org/10.1145/3290605.3300893

[11] Lesley Chiou and Catherine Tucker. 2020. *Social Distancing, Internet Access and Inequality*. Working Paper 26982. National Bureau of Economic Research. https://doi.org/10.3386/w26982

[12] Sharon Lynn Chu, Francis Quek, Sourabh Bhangaonkar, Amy Boettcher Ging, and Kumar Sridharamurthy. 2015. Making the Maker: A Means-to-an-Ends approach to nurturing the Maker mindset in elementary-aged children. *International Journal of Child-Computer Interaction* 5 (2015), 11–19. https://doi.org/10.1016/j.ijcci.2015.08.002 Digital Fabrication in Education.

[13] Nick Collins, Alex McLEAN, Julian Rohrhuber, and Adrian Ward. 2003-12. Live coding in laptop performance. *Organised Sound* 8, 3 (2003-12), 321–330. https://doi.org/10.1017/S135577180300030X

[14] Stanley Coren and Joan S Girgus. 1980. Principles of perceptual organization and spatial distortion: the gestalt illusions. *Journal of Experimental Psychology: Human Perception and Performance* 6, 3 (1980), 404.

[15] Sayamindu Dasgupta and Benjamin Mako Hill. 2018. How "Wide Walls" Can Increase Engagement: Evidence From a Natural Experiment in Scratch. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3173935

[16] Dragoş Datcu, Stephan G. Lukosch, and Heide K. Lukosch. 2016. Handheld Augmented Reality for Distributed Collaborative Crime Scene Investigation. In *Proceedings of the 19th International Conference on Supporting Group Work* (Sanibel Island, Florida, USA) *(GROUP '16)*. Association for Computing Machinery, New York, NY, USA, 267–276. https://doi.org/10.1145/2957276.2957302

[17] Jill Denner and Linda Werner. 2007. Computer Programming in Middle School: How Pairs Respond to Challenges. *Journal of Educational Computing Research* 37, 2 (2007), 131–150. https://doi.org/10.2190/12T6-41L2-6765-G3T2 arXiv:https://doi.org/10.2190/12T6-41L2-6765-G3T2

[18] Volodymyr Dziubak, Ben Lafreniere, Tovi Grossman, Andrea Bunt, and George Fitzmaurice. 2018. Maestro: Designing a System for Real-Time Orchestration of 3D Modeling Workshops. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 287–298. https://doi.org/10.1145/3242587.3242606

[19] Matthew S Eastin and Robert LaRose. 2000. Internet self-efficacy and the psychology of the digital divide. *Journal of computer-mediated communication* 6, 1 (2000), JCMC611.

[20] Keith Elliot. 2016. Swift Playground–Interactive Awesomeness. https://medium.com/swift-programming/swift-playgrounds-interactive-awesomeness-2a74143c233.

[21] Barrett Ens, Fraser Anderson, Tovi Grossman, Michelle Annett, Pourang Irani, and George Fitzmaurice. 2017. Ivy: Exploring spatially situated visual programming for authoring and understanding intelligent environments. In *Proceedings of the 43rd Graphics Interface Conference*. Association for Computing Machinery, Waterloo, Canada, 156–162.

[22] Espressif. 2020. ESP32 Overview | Espressif Systems. https://www.espressif.com/en/products/socs/esp32/overview

[23] Espressif. 2020. ESP8266 Overview | Espressif Systems. https://www.espressif.com/en/products/socs/esp8266/overview

[24] Sally Fincher and Marian Petre. 2004. *Computer science education research*. CRC Press, London, UK.

[25] Daniel Fitton, Janet C. Read, and John Dempsey. 2015. Exploring Children's Designs for Maker Technologies. In *Proceedings of the 14th International Conference on Interaction Design and Children* (Boston, Massachusetts) *(IDC '15)*. Association for Computing Machinery, New York, NY, USA, 379–382. https://doi.org/10.1145/2771839.2771921

[26] Terrell Glenn, Ananya Ipsita, Caleb Carithers, Kylie Peppler, and Karthik Ramani. 2020. StoryMakAR: Bringing Stories to Life With An Augmented Reality & Physical Prototyping Toolkit for Youth. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376790

[27] Google. 2020. How visual data powers shared AR experiences | Google Developers. https://developers.google.com/ar/cloud-anchors-privacy.

[28] Google. 2020. Privacy Policy – Privacy & Terms. https://policies.google.com/privacy.

[29] Google. 2020. Share AR experiences with Cloud Anchors | ARCore | Google Developers. https://developers.google.com/ar/develop/unity-arf/cloud-anchors/overview

[30] Google. 2021. Compute Engine: Virtual Machines (VMs) | Google Cloud. https://cloud.google.com/compute/

[31] Saul Greenberg. 2007. Toolkits and interface creativity. *Multimedia Tools and Applications* 32, 2 (2007), 139–159.

[32] The Samsung Group. 2020. What is AR Zone on the Galaxy S20? https://www.samsung.com/au/support/mobile-devices/what-is-ar-zone-on-the-galaxy-s20/.

[33] Shuchi Grover and Roy Pea. 2013. Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher* 42, 1 (2013), 38–43. https://doi.org/10.3102/0013189X12463051 arXiv:https://doi.org/10.3102/0013189X12463051

[34] Idit Ed Harel and Seymour Ed Papert. 1991. *Constructionism*. Ablex Publishing, Boston, MA.

[35] Valentin Heun, James Hobin, and Pattie Maes. 2013. Reality editor: programming smarter objects. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. Association for Computing Machinery, Zurich, Switzerland, 307–310.

[36] Donell Holloway and Lelia Green. 2016. The Internet of toys. *Communication Research and Practice* 2, 4 (2016), 506–519. https://doi.org/10.1080/22041451.2016.1266124

[37] Nathaniel Hudson, Benjamin Lafreniere, Parmit K. Chilana, and Tovi Grossman. 2018. Investigating How Online Help and Learning Resources Support Children's Use of 3D Design Software. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173831

[38] Ke Huo, Tianyi Wang, Luis Paredes, Ana M. Villanueva, Yuanzhi Cao, and Karthik Ramani. 2018. SynchronizAR: Instant Synchronization for Spontaneous and Spatial Collaborations in Augmented Reality. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 19–30. https://doi.org/10.1145/3242587.3242595

[39] PIRITA Ihamäki and K Heljakka. 2018. Smart, skilled and connected in the 21st century: Educational promises of the Internet of Toys (IoToys). In *Proceedings of 2018 Hawaii University International Conference, Arts, Humanities, Social Sciences & Education 3.-6.1. 2018*. Hawaii University International Conferences, Honolulu, Hawaii, 19 pages.

[40] Niantic Inc. and The Pokemon Company. 2020. Catch Pokémon in the Real World with Pokémon GO! https://www.pokemongo.com/en-us/

[41] Inc. Innovation First International. 2021. Welcome to VEX Robotics. https://www.vexrobotics.com/

[42] Spatial io. 2020. Spatial. https://spatial.io/

[43] 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* (Trondheim, Norway) *(IDC '18)*. Association for Computing Machinery, New York, NY, USA, 611–616. https://doi.org/10.1145/3202185.3210784

[44] Wall Street Journal. 2021. Testing Elon Musk's STARLINK: Is it really a rural internet game changer? https://www.wsj.com/video/series/in-depth-features/testing-elon-musk-starlink-is-it-really-a-rural-internet-game-changer/92E2D423-50F2-4873-851A-1D6FD54B657C

[45] Majeed Kazemitabaar, Jason McPeak, Alexander Jiao, Liang He, Thomas Outing, and Jon E. Froehlich. 2017. MakerWear: A Tangible Approach to Interactive Wearable Creation for Children. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 133–145. https://doi.org/10.1145/3025453.3025887

[46] Juraj Kubelka, Romain Robbes, and Alexandre Bergel. 2018-05-27. The road to live programming: insights from the practice. In *Proceedings of the 40th International Conference on Software Engineering*. ACM, Gothenburg Sweden, 1090–1101. https://doi.org/10.1145/3180155.3180200

[47] Thomas Kubitza and Albrecht Schmidt. 2017. meSchup: A platform for programming interconnected smart things. *Computer* 50, 11 (2017), 38–49.

[48] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. *Evaluation Strategies for HCI Toolkit Research*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3173574.3173610

[49] Google LLC. 2019. Blockly. https://developers.google.com/blockly/

[50] Yang Lu and Li Da Xu. 2019. Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics. *IEEE Internet of Things Journal* 6, 2 (2019), 2103–2115. https://doi.org/10.1109/JIOT.2018.2869847

[51] Caitlin McGarry. 2021. 5G speed: 5g VS 4G performance compared. https://www.tomsguide.com/features/5g-vs-4g

[52] Sharan B Merriam and Elizabeth J Tisdell. 2015. *Qualitative research: A guide to design and implementation*. John Wiley & Sons, San Francisco, CA, USA.

[53] Amon Millner and Edward Baafi. 2011. Modkit: Blending and Extending Approachable Platforms for Creating Computer Programs and Interactive Objects. In *Proceedings of the 10th International Conference on Interaction Design and Children* (Ann Arbor, Michigan) *(IDC '11)*. Association for Computing Machinery, New York, NY, USA, 250–253. https://doi.org/10.1145/1999030.1999074

[54] S. N. H. Mohamad, A. Patel, R. Latih, Q. Qassim, Liu Na, and Y. Tew. 2011. Block-based programming approach: challenges and benefits. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. IEEE, Bandung, Indonesia, 1–5.

[55] F. Mondada, M. Bonani, F. Riedo, M. Briod, L. Pereyre, P. Retornaz, and S. Magnenat. 2017. Bringing Robotics to Formal Education: The Thymio Open-Source Hardware Robot. *IEEE Robotics Automation Magazine* 24, 1 (2017), 77–85. https://doi.org/10.1109/MRA.2016.2636372

[56] Brad Myers, Scott E Hudson, and Randy Pausch. 2000. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7, 1 (2000), 3–28.

[57] Koya Narumi, Steve Hodges, and Yoshihiro Kawahara. 2015. ConductAR: an augmented reality based tool for iterative design of conductive ink circuits. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15*. ACM Press, Osaka, Japan, 791–800. https://doi.org/10.1145/2750858.2804267

[58] Michael Nebeling, Janet Nebeling, Ao Yu, and Rob Rumble. 2018. ProtoAR: Rapid Physical-Digital Prototyping of Mobile Augmented Reality Applications. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press, Montreal QC, Canada, 1–12. https://doi.org/10.1145/3173574.3173927

[59] Vinh T Nguyen, Kwanghee Jung, and Tommy Dang. 2020. BlocklyAR: A Visual Programming Interface for Creating Augmented Reality Experiences. *Electronics* 9, 8 (2020), 1205.

[60] Click Nilson. 2007. Live coding practice. In *Proceedings of the 7th international conference on New interfaces for musical expression - NIME '07*. ACM Press, New York, New York, 112. https://doi.org/10.1145/1279740.1279760

[61] Nintendo. 2020. Mario Kart Live: Home Circuit – Official Site. https://mklive.nintendo.com/

[62] A. M. Olney and S. D. Fleming. 2019. A Cognitive Load Perspective on the Design of Blocks Languages for Data Science. In *2019 IEEE Blocks and Beyond Workshop (B B)*. IEEE, Memphis, TN, 95–97. https://doi.org/10.1109/BB48857.2019.8941224

[63] Dan R. Olsen. 2007. Evaluating User Interface Systems Research. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (Newport, Rhode Island, USA) *(UIST '07)*. Association for Computing Machinery, New York, NY, USA, 251–258. https://doi.org/10.1145/1294211.1294256

[64] Seymour A Papert. 2020. *Mindstorms: Children, computers, and powerful ideas.* Basic books, New York, NY.

[65] Thomas H. Park, Rachel M. Magee, Susan Wiedenbeck, and Andrea Forte. 2013. Children as Webmakers: Designing a Web Editor for Beginners. In *Proceedings of the 12th International Conference on Interaction Design and Children* (New York, New York, USA) *(IDC '13)*. Association for Computing Machinery, New York, NY, USA, 419–422. https://doi.org/10.1145/2485760.2485845

[66] E. Pasternak, R. Fenichel, and A. N. Marshall. 2017. Tips for creating a block language with blockly. In *2017 IEEE Blocks and Beyond Workshop (B B)*. IEEE, Raleigh, NC, USA, 21–24.

[67] Photon. 2020. PUN. https://www.photonengine.com/pun

[68] PLTW. 2021. Homepage. https://www.pltw.org

[69] PTC PTC. 2020. Communicate. Collaborate. Get it done. https://chalk.vuforia.com/

[70] Kanjun Qiu, Leah Buechley, Edward Baafi, and Wendy Dubow. 2013. A Curriculum for Teaching Computer Science through Computational Textiles. In *Proceedings of the 12th International Conference on Interaction Design and Children* (New York, New York, USA) *(IDC '13)*. Association for Computing Machinery, New York, NY, USA, 20–27. https://doi.org/10.1145/2485760.2485787

[71] Iulian Radu and Blair MacIntyre. 2009. Augmented-reality scratch: a children's authoring environment for augmented-reality experiences. In *Proceedings of the 8th International Conference on Interaction Design and Children.* Association for Computing Machinery, Como, Italy, 210–213.

[72] Jun Rekimoto. 1996. TRANSVISION: A HAND-HELD AUGMENTED REALITY SYSTEM FOR COLLABORATIVE DESIGN. *Proceeding of Virtual Systems and Multimedia* 96 (1996), 6.

[73] Mitchel Resnick, Fred Martin, Randy Sargent, and Brian Silverman. 1996. Programmable bricks: Toys to think with. *IBM Systems journal* 35, 3.4 (1996), 443–452.

[74] Mitchel Resnick and Brian Silverman. 2005. Some reflections on designing construction kits for kids. In *Proceeding of the 2005 conference on Interaction design and children - IDC '05.* ACM Press, Boulder, Colorado, 117–122. https://doi.org/10.1145/1109540.1109556

[75] R. A. Roberto, D. Q. d. Freitas, F. P. M. Simões, and V. Teichrieb. 2013. A Dynamic Blocks Platform Based on Projective Augmented Reality and Tangible Interfaces for Educational Activities. In *2013 XV Symposium on Virtual and Augmented Reality.* IEEE, Cuiaba - Mato Grosso, Brazil, 1–9. https://doi.org/10.1109/SVR.2013.11

[76] Damien Constantine Rompapas, Christian Sandor, Alexander Plopski, Daniel Saakes, Joongi Shin, Takafumi Taketomi, and Hirokazu Kato. 2019. Towards large scale high fidelity collaborative augmented reality. *Computers & Graphics* 84 (2019), 24 – 41. https://doi.org/10.1016/j.cag.2019.08.007

[77] Michael Schrage. 2000. Serious play: The future of prototyping and prototyping the future. *Design Management Journal (Former Series)* 11, 3 (2000), 50–57.

[78] Scratch. 2021. Scratch Link for Micro:bit. https://scratch.mit.edu/microbit

[79] Snaappy. 2020. Snaappy - Augmented Reality social network. https://snaappy.com/en/ Library Catalog: snaappy.com.

[80] Inc. Snapchat. 2020. The fastest way to share a moment! https://www.snapchat.com/

[81] SpaceX. 2021. Starlink. https://www.starlink.com/

[82] Arduino Team. 2020. Get ready to Explore IoT with Arduino Education. https://blog.arduino.cc/2020/09/15/get-ready-to-explore-iot-with-arduino-education/

[83] The React Development Team. 2017. React: A Javascript Livrary for Building User interfaces. https://facebook.github.io/react/.

[84] Bird Brain Technologies. 2020. Bird brain Technologies: Hummingbird Robotics Kit. https://www.birdbraintechnologies.com/hummingbirdbit/.

[85] Balasaravanan Thoravi Kumaravel, Fraser Anderson, George Fitzmaurice, Bjoern Hartmann, and Tovi Grossman. 2019. Loki: Facilitating Remote Instruction of Physical Tasks Using Bi-Directional Mixed-Reality Telepresence. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 161–174. https://doi.org/10.1145/3332165.3347872

[86] Unity. 2020. Unity's AR Foundation Framework. https://unity.com/unity/features/arfoundation

[87] Will C Van den Hoonaard. 1997. *Working with sensitizing concepts: Analytical field research.* Thousand Oaks (Calif.): Sage, 1997., California.

[88] DWF Van Krevelen and Ronald Poelman. 2010. A survey of augmented reality technologies, applications and limitations. *International journal of virtual reality* 9, 2 (2010), 1–20.

[89] C. Vidal-Silva, J. Serrano-Malebran, and F. Pereira. 2019. Scratch and Arduino for Effectively Developing Programming and Computing-Electronic Competences in Primary School Children. In *2019 38th International Conference of the Chilean Computer Science Society (SCCC).* SCCC, Concepcion, Chile, 1–7. https://doi.org/10.1109/SCCC49216.2019.8966401

[90] Ana Villanueva, Zhengzhe Zhu, Ziyi Liu, Kylie Peppler, Thomas Redick, and Karthik Ramani. 2020. Meta-AR-App: An Authoring Platform for Collaborative Augmented Reality in STEM Classrooms. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376146

[91] Nicolas Villar, Haiyan Zhang, Daniel Cletheroe, Greg Saul, Christian Holz, Tim Regan, Oscar Salandin, Misha Sra, Hui-Shyong Yeo, and William Field. 2018. Project Zanzibar: A Portable and Flexible Tangible Interaction Platform. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18.* ACM Press, Montreal QC, Canada, 1–13. https://doi.org/10.1145/3173574.3174089

[92] WallaMe. 2020. WallaMe Augmented Reality | Hide messages in the real world. http://walla.me/

[93] Adrian Ward, Julian Rohrhuber, Fredrik Olofsson, Alex McLean, Dave Griffiths, Nick Collins, and Amy Alexander. 2004. Live algorithm programming and a temporary organisation for its promotion. In *Proceedings of the README Software Art Conference*, Vol. 289. README Software Art Conference, Aarhus, Denmark, 290.

[94] David Weintrop and Uri Wilensky. 2017. Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms. *ACM Trans. Comput. Educ.* 18, 1, Article 3 (oct 2017), 25 pages. https://doi.org/10.1145/3089799

[95] Thomas Wells and Steven Houben. 2020. CollabAR: Investigating the Mediating Role of Mobile AR Interfaces on Co-Located Group Collaboration. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376541

[96] Karl D.D. Willis, Ivan Poupyrev, and Takaaki Shiratori. 2011. Motionbeam: a metaphor for character interaction with handheld projectors. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. ACM Press, Vancouver, BC, Canada, 1031. https://doi.org/10.1145/1978942.1979096

[97] Karl D. D. Willis, Takaaki Shiratori, and Moshe Mahler. 2013. HideOut: mobile projector interaction with tangible objects and surfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction - TEI '13*. ACM Press, Barcelona, Spain, 331. https://doi.org/10.1145/2460625.2460682

[98] Jeannette M Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.

[99] Te-Yen Wu, Jun Gong, Teddy Seyed, and Xing-Dong Yang. 2019. Proxino: Enabling Prototyping of Virtual Circuits with Physical Proxies. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 121–132. https://doi.org/10.1145/3332165.3347938

[100] Bieke Zaman, Maarten Van Mechelen, and Lizzy Bleumers. 2018. When Toys Come to Life: Considering the Internet of Toys from an Animistic Design Perspective. In *Proceedings of the 17th ACM Conference on Interaction Design and Children* (Trondheim, Norway) *(IDC '18)*. Association for Computing Machinery, New York, NY, USA, 170–180. https://doi.org/10.1145/3202185.3202745

[101] Junling Zhang and Jing Liu. 2018. Construction of Scaffolding Instruction Mode for Mblock for Arduino Maker Course Based on Design Thinking. In *Proceedings of the 2nd International Conference on Computer Science and Application Engineering* (Hohhot, China) *(CSAE '18)*. Association for Computing Machinery, New York, NY, USA, Article 147, 6 pages. https://doi.org/10.1145/3207677.3278031