

**From the holdings of the Department of Special Collections , Princeton University Library,  
Princeton, NJ**

These documents are either in the public domain, or can only be used for educational and research purposes (Fair use) as per U.S. Copyright law (text below). By accessing this file, all users agree that their use falls in the public domain or within fair use as defined by the copyright law. We do not charge any permission or usage fees for the publication of images of material in our collections, including those provided by Princeton University Library, via our website, catalog, or directly from staff. For more information regarding copyright, credit, and citation please review the information on the Copyright, Credit and Citation Guidelines page of our website.

<https://library.princeton.edu/special-collections/policies/copyright-credit-and-citation-guidelines>. If you have any questions, including how to order higher-quality images from the collection, contact Special Collections Public Services via our Ask Us! Form. <https://library.princeton.edu/special-collections/ask-us>

---

Master's Theses, 2022-2026

Computer Science, 2023

---

2025

# Code It, See It, Touch It: Vision-Powered AR Programming Tool for Children

Zhou, Shuyao

Princeton, NJ : Princeton University

---

<http://arks.princeton.edu/ark:/88435/dsp013t945v20d>

*U.S. Copyright law: The copyright law of the United States (Title 17, §108, United States Code) governs the making of photocopies or other reproductions of copyrighted material. Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be used for any purpose other than private study, scholarship, or research. If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of fair use that user may be liable for copyright infringement. This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve a violation of copyright law. <https://www.copyright.gov/title17/title17.pdf>*

# CODE IT, SEE IT, TOUCH IT: VISION-POWERED AR PROGRAMMING TOOL FOR CHILDREN

BY

Shuyao Zhou

A THESIS

PRESENTED TO THE FACULTY OF PRINCETON UNIVERSITY

IN CANDIDACY FOR THE DEGREE

MASTER OF SCIENCE IN ENGINEERING

RECOMMENDED FOR ACCEPTANCE

BY THE DEPARTMENT OF COMPUTER SCIENCE

ADVISER: ANDRÉS MONROY-HERNÁNDEZ

MAY 2025



# Abstract

Children increasingly interact with intelligent systems through Augmented Reality (AR), but few authoring tools enable them to create expressive experiences grounded in the physical world. We present an AR authoring tool that enables children to create interactive experiences by linking visual code blocks to physical objects recognized by vision-based AI models. The system is built on a block-based programming interface and leverages YOLOv11 to identify physical objects through the device camera. Building on the accessibility of block-based programming, our system encourages creativity and exploration by extending traditional visual authoring tools into the physical environment. Through user studies with 20 children aged 7–16 in the United States and Argentina, we found that children could successfully author interactive AR stories using object detection, creatively extending their environments into hybrid digital-physical play. We discuss the system design, implementation, and implications for educational AI tools that blend embodied programming with computer vision. Our findings suggest that the system supports playful, explorative learning, while also highlighting challenges in real-time object tracking accuracy and responsiveness.

# Acknowledgements

I have been fortunate to be surrounded by wonderful people during my time here—always, and in all ways.

I want to thank my wonderful advisor, Professor Andrés Monroy-Hernández. His support has been central to both the development of this thesis and my personal growth. I am also thankful to Professor Parastoo Abtahi for her valuable feedback and for serving as a reader on my committee.

I want to extend special thanks to the amazing collaborators: Lei Zhang, Amna Liaqat, and Brian Berengard, who all offered critical support throughout. I also closely worked with an undergraduate student, Tinney Mak. I am also grateful to the entire Princeton HCI group for being such an inspiring and welcoming community, with special thanks to Yuhan Liu for her passion, thoughtful feedback, and friendship.

To all my friends, I will forever cherish the moments we shared—watching Broadway shows and concerts in NYC, traveling and snowboarding through new places, countless meals that became heart-to-hearts. I truly appreciate the friendships we built and hope our paths cross again somewhere, someday, out in the world.

To my family, you were always the first to offer support, and that gave me the confidence to pursue my path. To my grandpa, whose memory continues to guide me, and I hope he would be proud of how far I’ve come.

While it is impossible to name everyone individually, their influence remains deeply appreciated. This journey has been filled with wonderful memories, and I step forward with a heart full of gratitude.

# Contents

|                                                                                                                                                 |            |
|-------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>Abstract</b>                                                                                                                                 | <b>iii</b> |
| <b>Acknowledgements</b>                                                                                                                         | <b>iv</b>  |
| <b>1 Introduction</b>                                                                                                                           | <b>1</b>   |
| <b>2 Related Work</b>                                                                                                                           | <b>3</b>   |
| 2.1 Authoring Tools for Immersive Experiences . . . . .                                                                                         | 3          |
| 2.2 Visual Programming Languages . . . . .                                                                                                      | 4          |
| 2.3 Child–AI Interaction . . . . .                                                                                                              | 5          |
| <b>3 System</b>                                                                                                                                 | <b>6</b>   |
| 3.1 Block-based Programming Environments . . . . .                                                                                              | 7          |
| 3.2 Bridging the Virtual and Physical World via Vision-based AI Models . . . . .                                                                | 9          |
| 3.3 Implementation Details . . . . .                                                                                                            | 10         |
| <b>4 Evaluation</b>                                                                                                                             | <b>12</b>  |
| 4.1 Participants . . . . .                                                                                                                      | 12         |
| 4.2 Procedure . . . . .                                                                                                                         | 13         |
| 4.3 Measures and Analysis . . . . .                                                                                                             | 14         |
| <b>5 Results</b>                                                                                                                                | <b>16</b>  |
| 5.1 Vision Feature Enables AR Interactions with the Physical World; However,<br>Tracking Inaccuracy and Latency Remain Key Challenges . . . . . | 16         |
| 5.2 Vision Feature Facilitates Social Interactions . . . . .                                                                                    | 17         |

|          |                                                                                                                                                              |           |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 5.3      | Vision feature unlocks new creative possibilities; however, participants desire<br>3D model creation from real objects and greater spatial control . . . . . | 18        |
| <b>6</b> | <b>Discussion</b>                                                                                                                                            | <b>21</b> |
| 6.1      | Programming intricate interactive behaviors between the virtual and physical<br>worlds . . . . .                                                             | 21        |
| 6.2      | Limitations . . . . .                                                                                                                                        | 22        |
| 6.2.1    | System limitations . . . . .                                                                                                                                 | 22        |
| 6.2.2    | Evaluation limitations . . . . .                                                                                                                             | 23        |
| <b>7</b> | <b>Conclusion</b>                                                                                                                                            | <b>24</b> |
|          | <b>References</b>                                                                                                                                            | <b>25</b> |

# Chapter 1

## Introduction

Prior work, such as Scratch [39], has demonstrated the transformative potential of empowering children as creators of digital experiences. However, integrating artificial intelligence (AI) into child-authored augmented reality (AR) experiences remains rare, primarily due to steep technical barriers in 3D modeling and the complexity of building meaningful interactions between virtual characters and real-world objects. Designing such interactions typically demands proficiency in programming and a deep understanding of AI models—skills that are often inaccessible to young learners.

Recent advancements in computer vision have significantly expanded how AR systems can perceive and interact with the physical environment [16]. Real-time object detection algorithms, particularly the YOLO (You Only Look Once) family, have emerged as key enablers for environmental sensing. The evolution of YOLO—from its initial release [38] to the latest YOLOv11 [24]—has steadily prioritized improvements in speed, accuracy, and computational efficiency, making real-time applications in AR both feasible and practical [12].

Despite these technological strides, AR authoring tools for children have remained largely constrained to digital-only interactivity, offering limited support for bridging the physical and virtual worlds. Most systems prioritize content consumption or restrict creative agency to pre-defined digital assets, leaving little room for real-world engagement. To address this critical gap, we introduce a child-friendly AR programming environment that places real-time object detection at the core of storytelling and creative exploration.

Our system aims to lower technical barriers and enable children to design AI-enabled AR experiences through an accessible, block-based programming interface. Building on findings from prior research [28], which showed that children are highly motivated to incorporate physical toys into their AR narratives, we explore how real-time vision models can further enhance engagement with the physical world. Through a visual drag-and-drop interface embedded within the AR environment, children can compose programs that link virtual character behaviors to physical object recognition, using off-the-shelf vision-based AI models without needing to manage underlying complexities.

The vision feature was designed to seamlessly integrate into the creative process, abstracting technical details while preserving expressive control. To evaluate its usability and creative potential, we conducted workshops with 20 children (ages 7–16) across the United States and Argentina. Participants created a wide range of experiences—including storytelling, gaming, and educational scenarios—that demonstrated the system’s ability to support imaginative, interactive AR creations. At the same time, we observed challenges inherent to real-time AI integration, such as misalignments between detected objects and user expectations, which point to important future directions.

Our contributions are as follows:

1. We present a child-friendly authoring tool that enables the creation of vision-powered AR experiences.
2. We demonstrate the tool’s expressiveness through a set of novel example applications developed by children.
3. We provide empirical insights from user workshops that reveal both the opportunities and limitations of integrating real-time AI into accessible AR authoring environments.

# Chapter 2

## Related Work

We draw from prior work in AR/VR authoring tools, visual programming languages, and child–AI interaction.

### 2.1 Authoring Tools for Immersive Experiences

Authoring AR/VR experiences has historically posed a high barrier to entry for end-users, especially those without extensive technical expertise. In recent years, many novel AR/VR prototyping tools have sought to lower the “floor” of creating immersive experiences [43, 26, 35, 34, 50, 44, 27, 47, 5, 17, 14, 49]. A core challenge remains enabling the creation of interactive behaviors, which traditionally requires extensive programming knowledge.

Visual programming languages have proven effective for enabling novice users to define interactive behaviors. For example, FlowMatic [50] allows users to program interactions via dataflow diagrams in VR. ProInterAR [49] combines a tablet and AR-HMD to let users construct block-based interactive AR scenes. Teachable Reality [31] enables users to create AR interactions through on-demand tangible and gestural computer vision models. However, these tools primarily target adult designers or novice programmers, rather than young children.

As AR increasingly enters educational contexts [2], recent research has begun to recognize the importance of enabling children (e.g., K–12 students) to author or customize their own AR experiences [41, 51]. Several systems have been proposed to empower children as AR creators [37, 1, 42, 30, 3]. For instance, AR Scratch [37] extends the Scratch platform to support

simple AR experiences, and MovableMaker [1] allows children to design movable AR effects through guided UIs. ExposAR [30] enables collaborative AR creation via image and plane recognition. Other efforts, such as MyPart [46], introduce sensing techniques that encourage exploration of the physical world, while AutomataStage [19] combines visual programming, paper sketches, and physical sensors for multidisciplinary learning.

Despite these advances, a key limitation of existing child-focused AR authoring tools is their relatively low ceiling of expressiveness—particularly for creating AI-enabled AR experiences. As AI-driven AR applications become increasingly prevalent [45, 18], our work advances this line of research by empowering children to leverage vision-based AI models to program their own AR experiences. Specifically, our system raises the ceiling of expressiveness by enabling children to define dynamic, real-time interactions between virtual characters and the physical world, grounded in live AI-driven perception.

## 2.2 Visual Programming Languages

Block-based programming languages have been widely adopted to support children’s learning of computational thinking by lowering syntactic complexity [48]. Over the decades, several environments have been introduced to teach children programming through visual code blocks, including Alice [9], Storytelling Alice [23], Scratch [39], and Blockly [15]. Scratch, in particular, has been influential in empowering children to creatively express computational ideas through easy-to-use block metaphors.

More recently, researchers have proposed extensions to block-based environments to enhance engagement and computational learning (e.g., [11, 4, 6]). For example, Visual StoryCoder[11] incorporates a voice-based agent to scaffold programming without instructor presence. However, most of these environments remain focused on authoring 2D digital content (e.g., 2D assets, animations) rather than interactive, spatial experiences.

Our work contributes to this area by exploring the design of a block-based programming environment within AR, enabling children to manipulate code blocks spatially and author interactions between virtual and physical objects. By embedding programming directly into the

3D environment and linking it with real-world AI perception, we extend block-based learning into a spatial domain.

## 2.3 Child–AI Interaction

AI technologies are increasingly shaping how children learn, create, and socialize. Kosoy et al. [25] found that children generally express positive attitudes toward AI and recognize its potential to support diverse aspects of their daily lives.

Within AR, computer vision models have become key enablers for real-time interaction with the physical environment [8, 16, 33, 40, 22]. For example, Naji et al.[33] incorporated YOLOv4 to support vocabulary learning by overlaying English labels on detected objects, and Blockplay.ai[40] used object recognition to enhance early math learning with physical manipulatives among young children. Kaviyaraj et al. [22] demonstrated improvements in test scores by applying synthetic image generation to boost AR object detection.

However, prior systems primarily relied on inserting pre-created 3D models into scenes rather than enabling dynamic, AI-driven interactions with arbitrary real-world objects. In contrast, our system leverages real-time object recognition directly from the camera feed to support child–AI co-creation grounded in the child’s immediate surroundings. This approach enables virtual characters to dynamically interact with whatever physical objects the child chooses, fostering a more contextually grounded child–AI collaboration.

Motivated by these gaps in prior work, we developed a system that empowers children to create AI-enabled AR experiences through a block-based visual programming environment. Our system leverages real-time vision models to ground interactions in children’s immediate surroundings, enabling dynamic connections between virtual characters and physical objects. In the following sections, we present an overview of our system, including its design rationale, system architecture, and interaction capabilities.

# Chapter 3

## System

The system is an iOS application designed for tablets and phones that enables users to program a digital character, by default a cute capybara, using drag-and-drop code blocks. Our system enables users to define interactive behaviors between the character and their physical surroundings powered by vision-based AI models. We have two primary design goals for the system:

- **D1 - Lowering the barrier of entry to program AI-enabled AR behaviors.** To enable children create AI-enabled AR experiences, we leverage visual programming languages which offer an intuitive medium for children to learn Computational Thinking (CT). Most notably, block-based programming environments such as Scratch [39] have shown to be effective by employing a constructionist approach that allows children to learn programming via building interesting projects. Along this line, in the system, we aim to further offer more possibilities in such authoring tools by enabling children to create experiences that are enabled by AI. We also aim to balance between facilitating users' creative expressions when tasks are automated via machine intelligence, and preserving children's agency of learning CT.
- **D2 - Facilitating interactions between the virtual and physical worlds.** As described by Billingham [2], one of the uniqueness of AR technology is in its support of seamless interaction between real and virtual environments. Recent studies with children have also found that they want more interactions between the virtual and physical worlds [28]. We aim to facilitate seamless interactions between the virtual and physical worlds both dur-

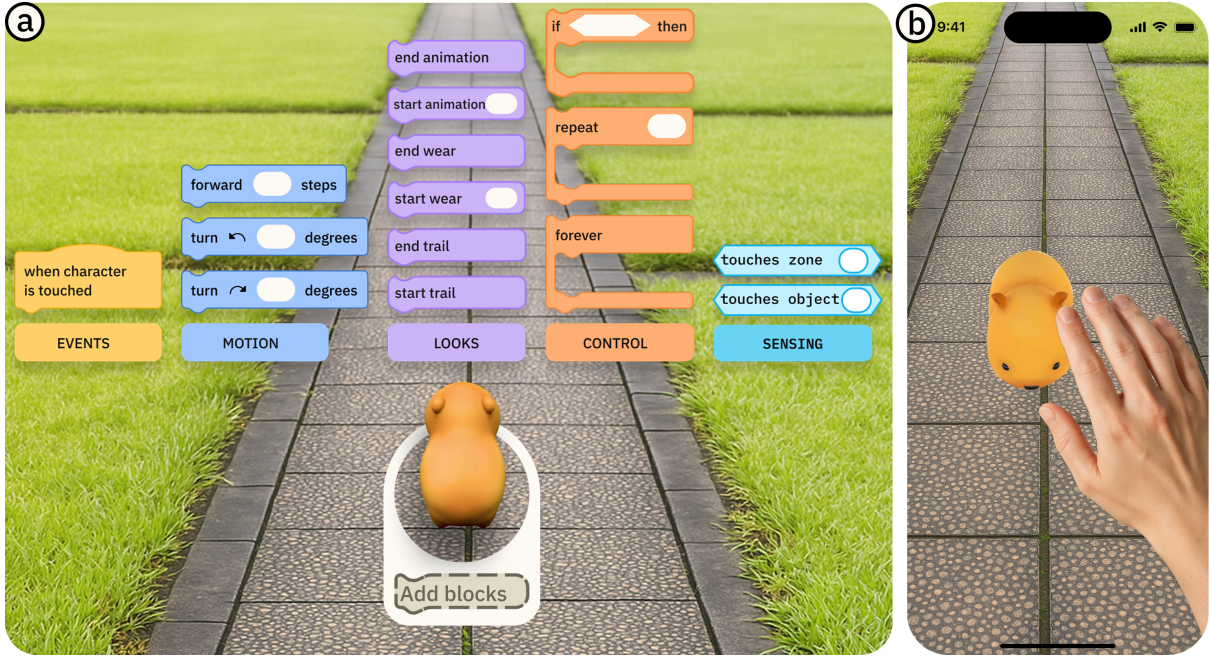


Figure 3.1: The Block-based Programming Environment of the system. (a) In Edit mode, the system visualizes five categories of code blocks and a canvas around the character for program construction in AR. (b) In Run mode, the user executes the program by sticking out their hand to touch the character in AR, mimicking a patting action in the physical world.

ing the authoring process and in the authoring outcomes. Specifically, when authoring, we aim to allow for intuitive interactions and a tangible interface metaphor that users are familiar with in the physical world. In the authoring outcomes, the virtual characters should have meaningful interactions with physical objects, based on the system’s understanding of the physical world.

### 3.1 Block-based Programming Environments

The system is based on a block-based programming environment in AR, as seen in Fig. 3.1. The code blocks feature an interlocking, puzzle-like design that visually guides users in constructing functional programs. These blocks are categorized into four distinct groups, each with a unique color for easy identification:

- (1) **Event Blocks (Yellow):** These blocks act as triggers, initiating the program’s execution. In the system, there is one event block: `When character is touched`. The user activates the program by sticking out their hand in front of the rear camera to touch the

virtual character, mimicking the action of patting the character, as seen in Fig. 3.1.b.

- (2) **Motion Blocks (Dark Blue):** These blocks control the character’s movement, including translation and rotation. Examples include `forward` to make the character step forward and `turn` to turn by a specific number of degrees. The number of steps and degrees can be selected via a slider.
- (3) **Looks Blocks (Purple):** These blocks add visual elements to the character. For instance, `start trail` creates a yellow paint trail as the character moves, allowing users to express 3D drawings in AR. Users can also customize the character’s accessories using the `start wear code` block, and customize the character’s animation using the `start animation code` block.
- (4) **Control Blocks (Orange):** These blocks introduce loops and conditional logic for more advanced programming. The `repeat` block enables the code within to run as many times as indicated in the input box, while the code within the `forever` block will run repeatedly until the user manually stops it by tapping the character. The `if` block enables conditional expressions. All control blocks act as containers which execute the code blocks contained in the middle.
- (5) **Sensing Blocks (Light Blue):** These blocks allow the system to “sense” the physical world. The `touch object` block allows users to select a physical object and detect collision between the virtual character and the object (details in Section 3.2). The `touch zone` block allows users to define a region on a physical plane and detect the collision between the virtual character and the region.

When the application launches, users are instructed to place the character in the physical environment by finding a flat open area. When coding, users can drag and drop code blocks directly from the code inventory onto a canvas surrounding the character in AR. The code blocks inventory hovers vertically above the canvas for easy access when the canvas is open. Users can drag the code blocks away from the canvas to remove them. The design of the system differentiated between on-screen interaction for building the program and gestural interaction

beyond the screen for triggering programs. Programming actions are restricted to on-screen interactions, such as tapping to open the canvas as well as dragging and dropping code blocks. In contrast, triggering the created programs uses playful gestures, such as physically acting out petting the character in the real world to activate “When character is touched”. This clear separation between building and play simplifies the programming process, while the gestures add playfulness to the interaction with the character and the physical world (**D2**).

## 3.2 Bridging the Virtual and Physical World via Vision-based AI Models

A key limitation of existing authoring tools for children is that the outcome experiences remain in the digital world. One of the design goals of the system is to facilitate children to create interactions between the virtual and the physical world (**D2**). We leverage off-the-shelf vision-based AI models to detect objects in the user’s physical surroundings (Section 3.3).

Users can program interactions with physical objects using the `touches object` block. When users drag this block onto the canvas, they are directed to the page where they can select objects for detection (see Fig. 3.2.a). We curated a set of 24 recognizable objects, including food items, utensils, furniture, and common workplace tools, from the YOLOv11 detection categories based on COCO dataset [29]. These objects were selected in order to support creative expressions of children’s everyday experiences.

When the user runs the program that contains `touches object`, the system initiates real-time detection and displays a “Searching for object” message on the UI (see Fig.3.2.b). Once the object is found in the scene, the app overlays a flashing bounding box on the object and updates the UI to display “Object found” (see Fig.3.2.c). If a collision between the character and the selected object is detected, the UI displays “Object touched” (see Fig.3.2.d), and the subsequent code blocks are executed accordingly.

To demonstrate the complete authoring workflow of combining the above functionalities and showcase the expressiveness of the system, we offer an example fictional scenarios enabled by the system, which are difficult to create using prior tools (see Fig. 3.3 for the screenshots of

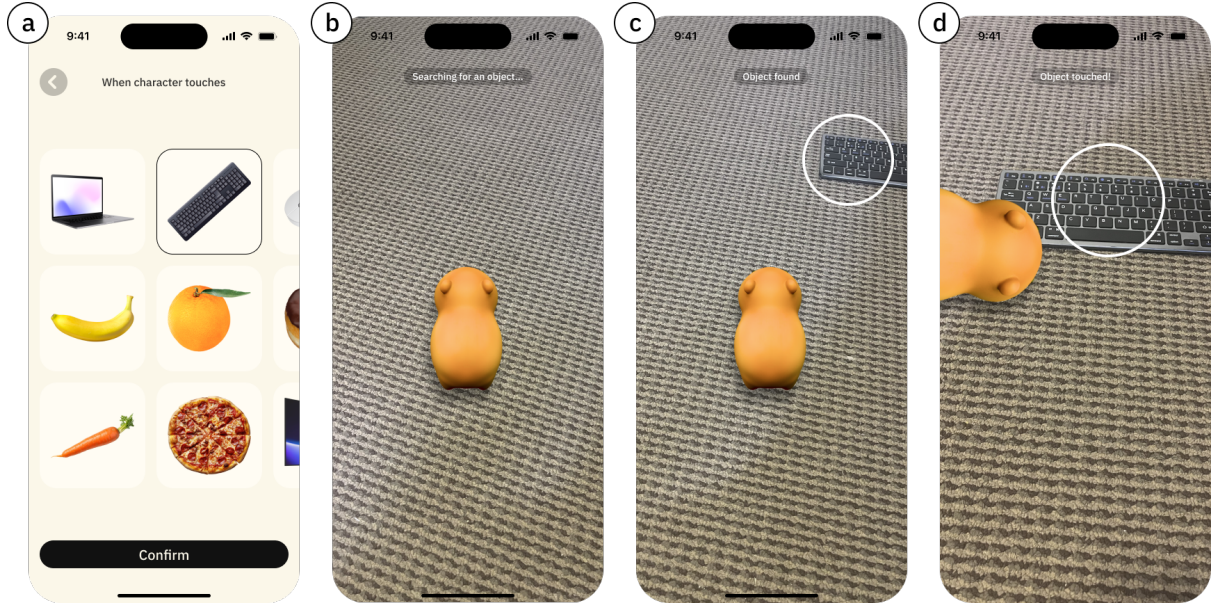


Figure 3.2: User interface for setting up object detection. (a) Object selection. (b) Searching for the selected object in the view. (c) Object found in the view. (d) Character touches the object.

the complete program): This experience features a custom tiger character preparing for the day by putting on a cowboy hat (middle) and walking to greet someone on a physical laptop (right), demonstrating the interactive behavior with the physical laptop powered by object recognition.

### 3.3 Implementation Details

Overall, the system is an iOS app implemented using SwiftUI, ARKit, RealityKit, and Vision. In this section we describe the implementation details of the object detection feature. A high level architecture for character and animation customization is depicted in Figure 3.4.

The system enables programming interactions between the virtual character and the physical surroundings using the codeblock touches object. Our system enables understanding the physical world by employing a deep neural network for real-time object detection, i.e., YOLOv11. We adapted the YOLOv11 model provided by Ultralytics [20] and converted it to CoreML format for deployment. To balance computational efficiency and detection accuracy, inference runs every 10 frames with a confidence threshold of 0.6. The single top-1 detected object is used for comparison with the selected object. Our system detects a “touch” event by sampling points within the selected object’s 2D bounding box and raycasting from those points

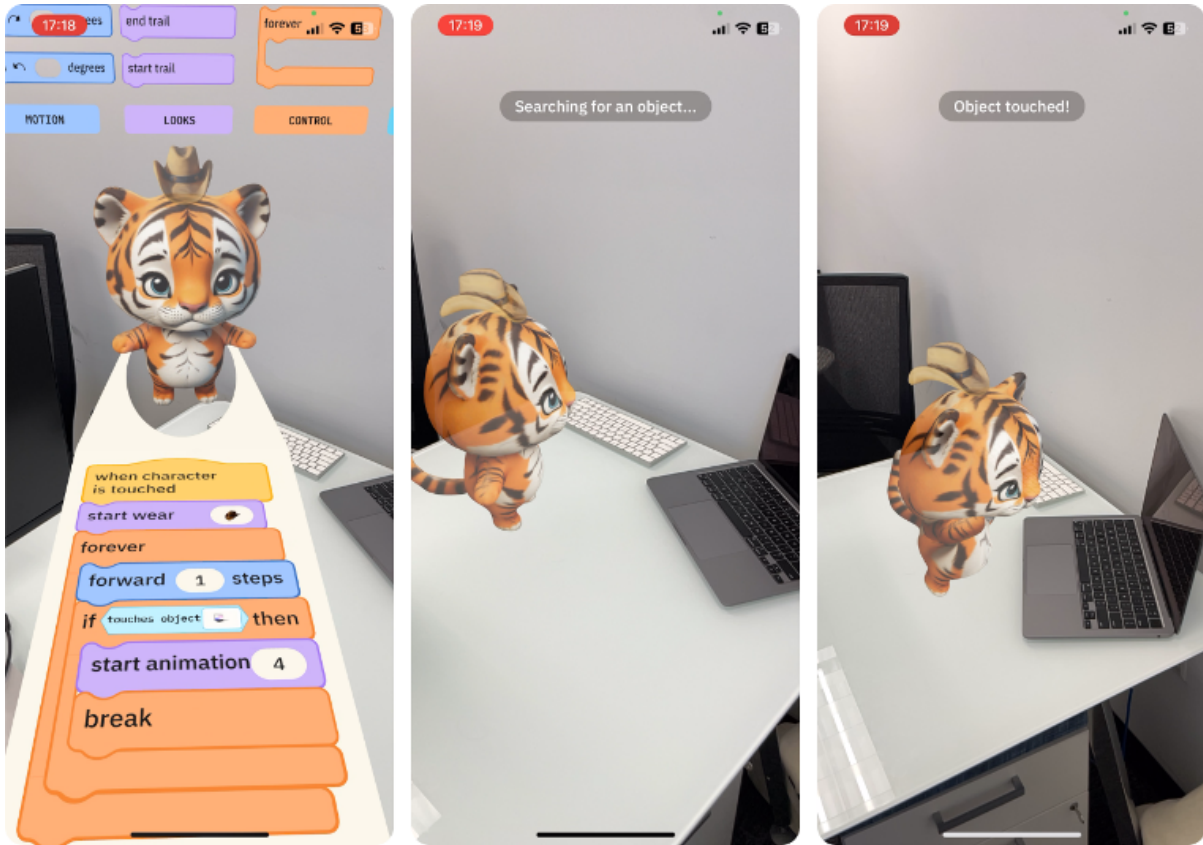


Figure 3.3: Example experience created using the system.

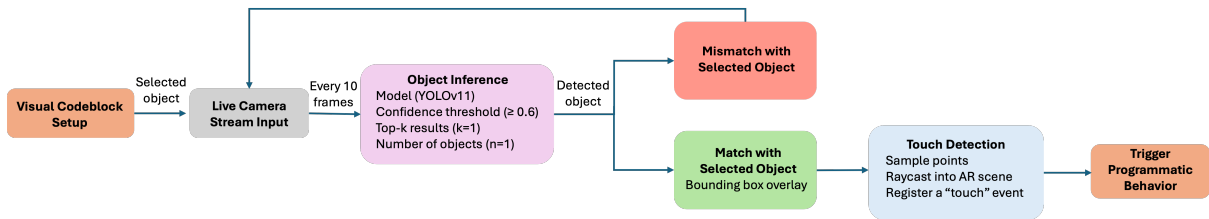


Figure 3.4: Vision-based Interaction Pipeline.

into the AR scene. If any ray intersects the character entity, we consider the object and the AR character to be colliding.

# Chapter 4

## Evaluation

To understand the benefits and challenges of object detection in The system, we conducted a series of user studies with children in United States and Argentina. Because these creations are subjective, we focused on three key research questions:

- **RQ1:** Are participants able to create experiences using the object detection feature?
- **RQ2:** What are the creative uses of this feature?
- **RQ3:** What are the benefits and challenges of using object detection?

### 4.1 Participants

We recruited 20 participants (12 female and 8 male, age 7–16), including nine groups, across the United States and Argentina, to gather a diverse range of feedback. By default, each study session involved two participants, except for US1 and AG1, where a third participant was involved since they are friends to one of the other two participants. We recruited U.S. participants through word of mouth. We recruited participants in Argentina through local schools and a non-profit organization that provides digital literacy classes in Buenos Aires. The participants’ demographic information is shown in Table 4.1. All participants have used Scratch at least once in the studies, but we report the use of coding tools when their self-reported use time is more than one month to indicate their proficiency. For example, P4 reported using Scratch in a one-day workshop and is thus not included in Table 4.1. Over half of the participants (13 out of

Table 4.1: Participants Demographics Information. Sessions are labelled as *country name + session id*, where US stands for United States and AG stands for Argentina.

| ID  | Session | Gender | Age | Coding Tools Use         |
|-----|---------|--------|-----|--------------------------|
| P1  | US1     | Female | 13  | Scratch                  |
| P2  | US1     | Female | 13  | Scratch                  |
| P3  | US1     | Female | 13  | Scratch                  |
| P4  | US2     | Male   | 10  | N/A                      |
| P5  | US2     | Male   | 15  | Scratch                  |
| P6  | US3     | Female | 11  | N/A                      |
| P7  | US3     | Female | 7   | N/A                      |
| P8  | US4     | Female | 10  | Scratch                  |
| P9  | US4     | Female | 10  | Scratch, Tynker          |
| P10 | US5     | Female | 14  | Scratch, Blockly         |
| P11 | US5     | Female | 13  | N/A                      |
| P12 | AG1     | Female | 15  | Scratch                  |
| P13 | AG1     | Male   | 15  | Scratch                  |
| P14 | AG1     | Male   | 15  | App Inventor             |
| P15 | AG2     | Female | 15  | Scratch, Blockly, Tynker |
| P16 | AG2     | Male   | 16  | Scratch                  |
| P17 | AG3     | Male   | 15  | CODE.GAME                |
| P18 | AG3     | Male   | 15  | App Inventor             |
| P19 | AG4     | Male   | 14  | Tynker                   |
| P20 | AG4     | Female | 14  | N/A                      |

20) have experienced Augmented Reality, such as Pokémon GO and AR filters, at least once. Most participants (17 out of 20) have used no-code creation tools, such as Minecraft, at least once. We compensated each U.S. participant with an Amazon gift card worth \$15 USD for 90 minutes of their time. We elected not to provide compensation to Argentinian participants to avoid coercion after consulting with the non-profit we collaborated with.

## 4.2 Procedure

Our studies took place in a room on a university campus in the United States and at a non-profit organization in Argentina. Each study session lasted approximately 90 minutes. At the beginning of the study, participants and their parents signed consent forms and completed a demographics questionnaire. Parents were invited to stay in the room if they wanted. The experimenters then helped participants feel comfortable by talking about school and hobbies before starting the study.

The study session began with an introduction and a walkthrough of the system that lasted

approximately 20 minutes. We ran the system on an iPad Pro (13-inch, M4 Chip, 2024) throughout the sessions. During the walkthrough, participants were shown individual features, including each code block category as described in Section 3.1, to get familiar with the system.

After the walkthrough, we asked participants to complete a warm-up exercise. In this exercise, we asked participants to create the example experience: program a character’s day such that when it encounters a selected object (e.g., a backpack, banana, or chair), it turns and walks. We designed this exercise to help participants understand the novel experience that they can build by integrating the object detection feature and to see whether children can use the system to create such experiences. The researchers encouraged participants to ask any questions about the usage of the system, but they were not allowed to proactively provide instructions on how to accomplish the exercise. The warm-up exercise lasted for 15 minutes.

Then, participants completed a 30-minute open-ended authoring task. In this task, we told participants to create any experience that they imagined. To help facilitate creative ideas of using our features, we provided several physical objects (recognizable in our app), toy animals, and tools like playdough, markers, pen, and paper. Participants were free to ask questions throughout the task. The experimenters were permitted to answer technical questions about the app’s usage but were not allowed to guide what to create.

After finishing all activities, participants filled out a system usability survey. Then, we concluded the study session with an open-ended discussion with participants on what they enjoyed about our system, what they found challenging, and other usability topics, including what else they would like to be able to create with it. During the sessions, the experimenters took notes during observation in the study process and the retrospective interviews. Our institution’s IRB approved our study protocol.

### **4.3 Measures and Analysis**

We measure the usability of the system using a system usability survey adapted to children’s reading skills based on prior work ([36]). Participants rated the questions on a 5-point Likert scale (1=Strongly Disagree, to 5=Strongly Agree). We transcribed the interview data and

conducted semantic analysis on the data. The survey and interview responses collected in Argentina were translated by a bilingual researcher into English and Spanish. One researcher developed a codebook covering participants' experiences of using our system. Once this initial codebook was developed, three researchers coded the transcripts separately and met frequently to discuss and resolve disagreements. Our research team then grouped similar codes to create themes, which we present in the following section.

# Chapter 5

## Results

Participants' experience using the system during the studies suggest that the system empowered them to create engaging and personalized AR experiences enabled by AI. Specifically, all participants were able to complete the given warm-up exercise using the system, addressing **RQ1**.

Additionally, we visualize participants' responses to the system usability scale survey in Figure 5.1. Overall, we found that participants generally felt that the system was easy to use ( $M=3.80$ ,  $SD=1.01$ ). They also thought that they enjoyed playing the system ( $M=4.35$ ,  $SD=0.99$ ) and that they felt confident when playing the system ( $M=4.40$ ,  $SD=0.82$ ). To further investigate users' perceptions of the experiences using the system, we analyzed our interview data and solicited users' qualitative feedback. In the following sections, we provide more detailed insights into the participants' creative usage and perceptions of the system, also the challenges they faced.

### **5.1 Vision Feature Enables AR Interactions with the Physical World; However, Tracking Inaccuracy and Latency Remain Key Challenges**

Our findings indicate that the system's design successfully supported participants in creating interactive behaviors between virtual characters and the physical world.

For instance, participants in US1 noted that object detection allowed virtual characters to respond dynamically to real-world items: “When it saw an object, it would react in a different way.” Similarly, a participant in US2 emphasized the ease of incorporating household items into their stories: “You can just grab something from your house.” This highlights the system’s accessibility and its potential to foster creativity using familiar physical objects. Another participant elaborated on this flexibility:

“That you can add any objects you want. Like this doll, or the ‘mate.’” – AR5

Some participants also imagined more semantically meaningful interactions between the character and the environment, such as household chores. They even related these scenarios to owning a pet:

“Maybe you could get the capybara to do the laundry [...]. It would be fun. If I had a capybara, I’d use it to clean my room.” – US3

Despite the enthusiasm, participants also pointed out limitations in detection robustness. For example, in US1, one participant commented: “If they have the item but it looks different because of the brand or whatever, it might not register as quickly.” In the same session, others experienced difficulty in recognizing a backpack, stating: “You could work on the tracking a little better.” These observations underscore ongoing technical challenges and point toward opportunities for improving tracking accuracy and responsiveness in future iterations.

## **5.2 Vision Feature Facilitates Social Interactions**

In addition to enabling individual creativity, the vision feature also facilitated social interactions among participants. Children leveraged the object detection capability not only to create personalized experiences but also to engage in collaborative and playful activities with others.

For example, in US4, a group of participants proposed a hide-and-seek game that incorporated both physical and digital elements. One participant explained: “Hide and seek. Like hiding them [objects] around the place, and then, you give the tablet to someone else and then they code it to try and go find the item.” This scenario illustrates how object detection was

reimagined as a tool for social play—where one child hides objects in the physical environment, and another programs the virtual character to locate them using the system’s visual recognition capabilities.

Such interactions underscore the potential of vision-based AR systems to support not only individual engagement but also shared experiences that involve turn-taking, role-switching, and cooperative storytelling, expanding the scope of AR applications from solitary exploration to social play.

### **5.3 Vision feature unlocks new creative possibilities; however, participants desire 3D model creation from real objects and greater spatial control**

The vision feature unlocked new creative possibilities by enabling children to incorporate physical objects into their interactive stories. For instance, participants in AG4 created an interaction where the virtual character began a typing animation when touching a physical keyboard (Fig. 5.2). A participant in US2 remarked that vision recognition could “add more variation, and an actual object from your house could be used,” while another participant noted that it could “expand the story.” Similarly, participants in AG3 felt that object detection fostered imagination, allowing characters to “do many things.”

Building on these opportunities, participants expressed a desire for richer and more intricate interactions with the physical world. Several participants wished to create digital versions of real objects that virtual characters could pick up or interact with. For example, children in US5 suggested:

“It would be cool if it would create a digital version of that object and just pick it up sometimes.” – US5

“The panda could, if it touches an orange, an orange appears in its hand.” – US5

They also proposed scanning real-world objects to generate digital representations:

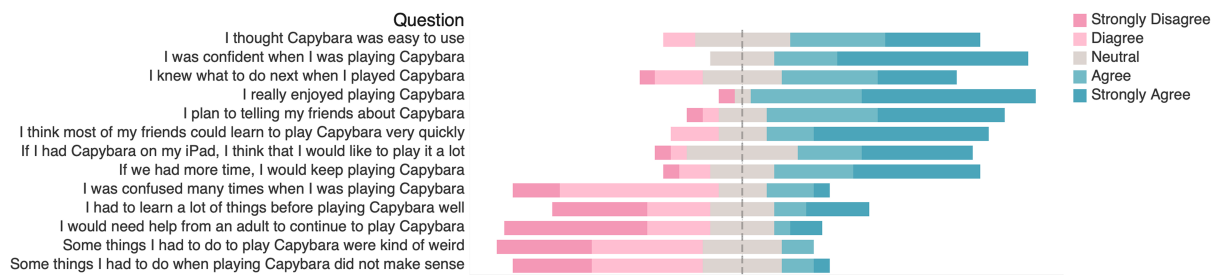


Figure 5.1: System Usability Survey Results.

“Or, you could maybe scan it to try and get an image of it.” – US5

Similarly, participants in US3 envisioned interactions involving virtual counterparts of physical objects, expressing interest in a tighter coupling between virtual and real-world elements.

Spatial control also emerged as a recurring theme; participants desired that virtual characters recognize and respond to physical barriers. For example, a participant in US1 suggested:

“Like, have it detect walls or things around it. Not just zones that you make. There was a ghost and it just passed.” – US1

Beyond object interaction, participants requested a broader selection of accessories to support storytelling. As one participant observed:

“It’s really cool, I just wish there were more options overall because I don’t know if everyone will have a random orange lying around.” – US5

Participants also envisioned the ability to detect multiple objects simultaneously:

“You could have multiple objects or multiple animals.” – US5

Moreover, some participants desired more control over detection timing, suggesting more interactive mechanisms. For instance, a participant in US1 proposed:

“Maybe tap on one person. And then it’ll only interact with the objects around it.”  
– US1

These findings highlight opportunities to further enhance vision-enabled AR storytelling by expanding object digitization, improving spatial awareness, and increasing the interactivity and variability of object-based triggers.



Figure 5.2: An usage case in the user studies where participants from AG4 created an interaction where the character started a typing animation when touching the physical keyboard (view from the back).

# Chapter 6

## Discussion

Our results demonstrate that the system design enables children to create AI-enabled AR experiences by programming interactive behaviors with the physical world. However, embedding AI into AR applications remains challenging. In particular, there is still significant room for improving tracking accuracy and reducing latency. Based on our findings, we highlight the opportunities and challenges revealed by the system design and discuss the broader implications of our research.

### **6.1 Programming intricate interactive behaviors between the virtual and physical worlds**

Our findings suggest that the system effectively facilitates children’s programming of interactive behaviors between the virtual and physical worlds. This capability is primarily supported by the system’s understanding of the physical world through vision-based models and collision detection between virtual and real-world objects. Participants expressed a desire for more intricate behaviors, such as virtual objects bouncing off walls or jumping onto physical furniture like chairs and tables.

Future research could explore recognizing a broader range of physical objects across different contexts, such as education. For example, children could create AR experiences involving interactions with physical textbooks [16]. Additionally, future systems could enrich collabora-

tive learning by enabling children to share, remix, and build upon each other’s creations [10].

Participants also envisioned digitizing elements of the physical world, such as virtually picking up a physical orange. We see exciting opportunities to incorporate recent advancements in 3D reconstruction technologies, such as 3D Gaussian Splatting [13], which could enable duplicating physical environments into virtual spaces. More specifically, Apple Object Capture also provides photogrammetry active sensing to create 3D object from a series of photographs directly. Allowing children to program virtual characters to interact with or modify the physical environment, such as picking up physical objects, could also be explored through techniques in Diminished Reality [32, 7, 21].

Beyond physical objects and environments, it would be valuable to empower children to author AR experiences that understand human-physical interactions. For example, a child could program a virtual companion that reads alongside them while studying, offers comfort when detecting a sad facial expression, or cheers when recognizing a smile. By integrating physical interaction understanding, children could create more engaging, intelligent, and personalized AR experiences.

## **6.2 Limitations**

Our paper explores empowering children to create vision-based AR experiences using AI models. However, several limitations remain.

### **6.2.1 System limitations**

Our system has several technical limitations. It relies on real-time object recognition, which is susceptible to misclassification, missed detections, and latency—particularly in uncontrolled environments or when objects are occluded or ambiguous. The YOLOv11 model used is pre-trained on the COCO dataset and supports only a fixed set of 80 object categories, limiting the diversity of physical interactions children can meaningfully program. Moreover, although visual block-based programming is accessible, it can become unwieldy as the complexity of programmed behaviors increases, imposing a ceiling on expressiveness without more advanced

programming constructs.

### **6.2.2 Evaluation limitations**

Our user evaluation also has important limitations. First, we conducted studies with a relatively small sample of participants ( $N = 20$ ). While the workshops provided valuable insights, the limited sample size restricts the generalizability of our findings. Additionally, we did not deeply explore how cultural or contextual differences between participants in Argentina and the United States might influence system perceptions or use patterns.

Given the short-term nature of the workshops, it remains unclear how sustained use of the system would impact children’s learning, creativity, or understanding of AI and AR concepts over time. Finally, our participant pool included a range of prior coding experience, from novices familiar with Scratch to an advanced participant who had already learned programming languages such as Python and Java by the age of 13. As a result, our findings may not fully generalize to all future users.

# Chapter 7

## Conclusion

In this paper, we introduced an authoring tool that empowers children to program interactions between the virtual and physical worlds using vision-based AI models. Our system enables a set of novel experiences by linking virtual behaviors to real-world objects, lowering technical barriers to AR authoring. Through a series of user studies conducted across two countries, we investigated children’s perceptions and engagement with the system.

Our findings suggest that vision-based AI models can enable children to create interactive AR experiences that seamlessly connect the physical and virtual worlds. The programming environment supported participants in designing dynamic interactions, while also highlighting challenges such as tracking inaccuracies and latency that could impact user expectations. We offer insights and recommendations that provide a foundation for future research and practice in leveraging vision-based AI to foster children’s creativity and learning in AR environments.

Looking ahead, future work could explore cross-platform integration, such as extending support to devices like Apple Vision Pro. Incorporating depth estimation to enable 3D reconstruction, expanding beyond object detection to include surface sensing and spatial behaviors, and conducting large-scale field deployments are promising directions to further enhance the system’s capabilities and understand its long-term impact.

# References

- [1] Michelle Annett et al. “MoveableMaker: facilitating the design, generation, and assembly of moveable papercraft”. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 565–574.
- [2] Mark Billinghurst et al. “Augmented reality in education”. In: *New horizons for learning* 12.5 (2002), pp. 1–5.
- [3] John Chen et al. ““Oh My God! It’s Recreating Our Room!” Understanding Children’s Experiences with A Room-Scale Augmented Reality Authoring Toolkit”. In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 2024, pp. 1–17.
- [4] Liuqing Chen et al. “ChatScratch: An AI-Augmented System Toward Autonomous Visual Programming Learning for Children Aged 6-12”. In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 2024, pp. 1–19.
- [5] Mengyu Chen, Andrés Monroy-Hernández, and Misha Sra. “SceneAR: Scene-based micro narratives for sharing and remixing in augmented reality”. In: *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2021, pp. 294–303.
- [6] Yunnong Chen et al. “CoRemix: Supporting Informal Learning in Scratch Community With Visual Graph and Generative AI”. In: *arXiv preprint arXiv:2412.05559* (2024).
- [7] Yi Fei Cheng et al. “Towards understanding diminished reality”. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–16.

- [8] Neil Chulpongsatorn et al. “Augmented Math: Authoring AR-Based Explorable Explanations by Augmenting Static Math Textbooks”. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 2023, pp. 1–16.
- [9] Matthew Conway et al. “Alice: lessons learned from building a 3D system for novices”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2000, pp. 486–493.
- [10] Sayamindu Dasgupta et al. “Remixing as a pathway to computational thinking”. In: *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*. 2016, pp. 1438–1449.
- [11] Griffin Dietz et al. “Visual StoryCoder: A Multimodal Programming Environment for Children’s Creation of Stories”. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023, pp. 1–16.
- [12] Tausif Diwan, G Anirudh, and Jitendra V Tembhurne. “Object detection using YOLO: Challenges, architectural successors, datasets and applications”. In: *multimedia Tools and Applications* 82.6 (2023), pp. 9243–9275.
- [13] Ben Fei et al. “3d gaussian splatting as new era: A survey”. In: *IEEE Transactions on Visualization and Computer Graphics* (2024).
- [14] Vittoria Frau et al. “Xrspotlight: Example-based programming of XR interactions using a rule-based approach”. In: *Proceedings of the ACM on Human-Computer Interaction* 7.EICS (2023), pp. 1–28.
- [15] Google. *Blockly*. 2024. URL: <https://developers.google.com/blockly>.
- [16] Aditya Gunturu et al. “Augmented Physics: Creating Interactive and Embedded Physics Simulations from Static Textbook Diagrams”. In: *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 2024, pp. 1–12.
- [17] Martin Hedlund et al. “Blocklyvr: Exploring block-based programming in virtual reality”. In: *Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia*. 2023, pp. 257–269.

- [18] Teresa Hirzle et al. “When xr and ai meet-a scoping review on extended reality and artificial intelligence”. In: *Proceedings of the 2023 CHI conference on human factors in computing systems*. 2023, pp. 1–45.
- [19] Yunwoo Jeong et al. “AutomataStage: An AR-mediated creativity support tool for hands-on multidisciplinary learning”. In: *Proceedings of the 2023 CHI conference on human factors in computing systems*. 2023, pp. 1–16.
- [20] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. *Ultralytics YOLO*. Version 8.0.0. Jan. 2023. URL: <https://github.com/ultralytics/ultralytics>.
- [21] Mohamed Kari, Reinhard Schütte, and Raj Sodhi. “Scene responsiveness for visuotactile illusions in mixed reality”. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 2023, pp. 1–15.
- [22] R Kaviyaraj and M Uma. “An Augmented Reality Framework for Education: Deep Learning Integration and Impact Evaluation”. In: *IEEE Access* (2025).
- [23] Caitlin Kelleher, Randy Pausch, and Sara Kiesler. “Storytelling alice motivates middle school girls to learn computer programming”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2007, pp. 1455–1464.
- [24] Rahima Khanam and Muhammad Hussain. “Yolov11: An overview of the key architectural enhancements”. In: *arXiv preprint arXiv:2410.17725* (2024).
- [25] Eliza Kosoy et al. *Children’s Mental Models of Generative Visual and Text Based AI Models*. 2024. arXiv: 2405.13081 [cs.HC]. URL: <https://arxiv.org/abs/2405.13081>.
- [26] Gun A Lee et al. “Immersive authoring of tangible augmented reality applications”. In: *Third IEEE and ACM international symposium on mixed and augmented reality*. IEEE. 2004, pp. 172–181.
- [27] Germán Leiva et al. “Rapido: Prototyping interactive ar experiences through programming by demonstration”. In: *The 34th Annual ACM Symposium on User Interface Software and Technology*. 2021, pp. 626–637.

- [28] Amna Liaquat et al. “Understanding Young People’s Creative Goals with Augmented Reality”. In: *arXiv preprint arXiv:2411.02601* (2024).
- [29] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV]. URL: <https://arxiv.org/abs/1405.0312>.
- [30] Mille Skovhus Lunding et al. “Exposar: Bringing augmented reality to the computational thinking agenda through a collaborative authoring tool”. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–14.
- [31] Kyzyl Monteiro et al. “Teachable reality: Prototyping tangible augmented reality with everyday objects by leveraging interactive machine teaching”. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023, pp. 1–15.
- [32] Shohei Mori, Sei Ikeda, and Hideo Saito. “A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects”. In: *IPSJ Transactions on Computer Vision and Applications* 9 (2017), pp. 1–14.
- [33] Hayder Faris Naji, Pinar Kullu, and Sahin Emrah Amrahov. “An augmented reality-based system with sound effects for teaching english in primary school”. In: *Education and Information Technologies* 29.10 (2024), pp. 12023–12045.
- [34] Michael Nebeling and Katy Madier. “360proto: Making interactive virtual reality & augmented reality prototypes from paper”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–13.
- [35] Michael Nebeling et al. “Protoar: Rapid physical-digital prototyping of mobile augmented reality applications”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–12.
- [36] Cynthia Putnam et al. “Adaptation of the system usability scale for user testing with children”. In: *Extended abstracts of the 2020 CHI conference on human factors in computing systems*. 2020, pp. 1–7.
- [37] Iulian Radu and Blair MacIntyre. “Augmented-reality scratch: a children’s authoring environment for augmented-reality experiences”. In: *Proceedings of the 8th International Conference on Interaction Design and Children*. 2009, pp. 210–213.

- [38] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.
- [39] Mitchel Resnick et al. “Scratch: programming for all”. In: *Communications of the ACM* 52.11 (2009), pp. 60–67.
- [40] Michael Rumbelow and Alf Coles. “The Promise of AI Object-Recognition in Learning Mathematics: An Explorative Study of 6-Year-Old Children’s Interactions with Cuise-naire Rods and the Blockplay. ai App”. In: *Education Sciences* 14.6 (2024), p. 591.
- [41] Manoela Silva et al. “Development of Design Principles for AR Authoring Tools for Education Based on Teacher’s Perspectives”. In: *IEEE Transactions on Learning Technologies* 17 (2023), pp. 677–690.
- [42] Abbey Singh et al. “Story creator: a toolkit for spatially-adaptive augmented reality storytelling”. In: *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2021, pp. 713–722.
- [43] Anthony Steed and Mel Slater. “A dataflow representation for defining behaviours within virtual environments”. In: *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*. IEEE. 1996, pp. 163–167.
- [44] Ryo Suzuki et al. “Realitysketch: Embedding responsive graphics and visualizations in AR through dynamic sketching”. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 2020, pp. 166–181.
- [45] Ryo Suzuki et al. “Xr and ai: Ai-enabled virtual, augmented, and mixed reality”. In: *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 2023, pp. 1–3.
- [46] Rundong Tian et al. “Mypart: Personal, portable, accurate, airborne particle counting”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 1338–1348.

- [47] Tianyi Wang et al. “GesturAR: An Authoring System for Creating Freehand Interactive Augmented Reality Applications”. In: *The 34th Annual ACM Symposium on User Interface Software and Technology*. UIST ’21. Virtual Event, USA: Association for Computing Machinery, 2021, 552–567. ISBN: 9781450386357. DOI: [10 . 1145 / 3472749 . 3474769](https://doi.org/10.1145/3472749.3474769). URL: <https://doi.org/10.1145/3472749.3474769>.
- [48] David Weintrop. “Block-based programming in computer science education”. In: *Communications of the ACM* 62.8 (2019), pp. 22–25.
- [49] Hui Ye et al. “ProInterAR: A Visual Programming Platform for Creating Immersive AR Interactions”. In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 2024, pp. 1–15.
- [50] Lei Zhang and Steve Oney. “Flowmatic: An immersive authoring tool for creating interactive scenes in virtual reality”. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 2020, pp. 342–353.
- [51] Nandi Zhang, Yukang Yan, and Ryo Suzuki. “From Following to Understanding: Investigating the Role of Reflective Prompts in AR-Guided Tasks to Promote Task Understanding”. In: *arXiv preprint arXiv:2501.13258* (2025).