



Insights from Youth Co-designers on Remote Multimodal Prototyping with Paper Playground

Brett L. Fiedler
brett.fiedler@colorado.edu
University of Colorado at Boulder
Boulder, Colorado, USA

Taliesin L. Smith
taliesin.smith@colorado.edu
University of Colorado at Boulder
Boulder, Colorado, USA

Jesse Greenberg
jesse.greenberg@colorado.edu
University of Colorado at Boulder
Boulder, Colorado, USA

Ann Eisenberg
ann.eisenberg@colorado.edu
University of Colorado at Boulder
Boulder, Colorado, USA

Emily B. Moore
emily.moore@colorado.edu
University of Colorado at Boulder
Boulder, Colorado, USA

ABSTRACT

Paper prototyping presents a low-entry barrier method to engaging youth in interaction design. Purely paper-based designs leave a large gap between ideation and implementation. Paper Playground is a prototyping tool that connects physical and virtual papers with JavaScript programs, enabling the creation of multimodal prototypes in both face-to-face and virtual settings. Paper Playground is being designed and developed through iterative co-design activities including youth and adults. Here we present findings from remote co-design sessions with youth, investigating what affordances the participants requested from a multimodal prototyping tool. We reflect on the co-designers' desires and remarks on paper use for interactive project design, remote collaborative use, and extensibility for physical computing.

CCS CONCEPTS

• **Human-centered computing** → **Systems and tools for interaction design**; *Web-based interaction*; **Collaborative interaction**.

KEYWORDS

co-design, web design, interactive design, web interactives

ACM Reference Format:

Brett L. Fiedler, Taliesin L. Smith, Jesse Greenberg, Ann Eisenberg, and Emily B. Moore. 2024. Insights from Youth Co-designers on Remote Multimodal Prototyping with Paper Playground. In *Interaction Design and Children (IDC '24)*, June 17–20, 2024, Delft, Netherlands. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3628516.3659400>

1 INTRODUCTION

Paper prototyping has revolutionized the way we engage youth in interaction design, offering a low-barrier entry point that fosters creativity and ideation [10, 14]. While beneficial in its simplicity and accessibility, paper prototyping presents a significant limitation: the gap between the paper-based conceptualization of ideas and their

digital implementation. This gap can restrict the creative potential of young designers and undermine the iterative process that is crucial to refining and realizing innovative design solutions [2, 16]. Additionally, solely paper-based approaches cannot fully embrace the multimodal experiences that modern technologies afford, an aspect increasingly recognized as vital in the design and use of augmented reality (AR) technologies [9]. AR's potential to merge virtual and physical realms across all senses—not just sight—promises to enrich interaction design in unprecedented ways, yet this promise remains largely untapped in current prototyping methodologies.

In response to these challenges, we introduce “Paper Playground,” an open-source multimodal prototyping tool in development to bridge the gap between physical and digital worlds. By integrating physical papers with JavaScript programs, Paper Playground seeks to enable the creation of dynamic, multimodal prototypes that can operate within in-person, virtual, and hybrid settings. This tool leverages the vast potential of AR technologies in a manner that engages all senses, offering a richer, more immersive interactive experience [9]. Through iterative co-design activities involving both youth and adults, we seek to understand the specific needs and affordances desired in a multimodal prototyping tool. Our initial findings from remote co-design sessions with school-aged, novice developers shed light on the capabilities that such a tool might possess to harness the creative potential of young designers, paving the way for a new multimodal era of interaction design.

2 BACKGROUND AND RELATED WORK

Our research in inclusive design involves co-design activities with youth, ranging from classroom-based sessions to small group collaborations [3], focused on creating multimodal, inclusive interactive STEM simulations (sims) [11]. These web-based sims are enriched with features, such as auditory descriptions [13], sonifications [15, 17], alternative input [4], among others. In our co-design efforts, we encounter a range of challenges [3], including the need for rapid prototyping tools that support multimodality and remote contexts.

As part of our development efforts, we aimed to create a tool to simplify multimodal interactive experience prototyping with non-developer co-designers that 1) highlights program component relationships that drive multimodal input/output for effective STEM-related design iteration and understanding, 2) enables collaborative co-design in co-located and remote settings, and 3)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
IDC '24, June 17–20, 2024, Delft, Netherlands
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0442-0/24/06
<https://doi.org/10.1145/3628516.3659400>

encourages diverse sensory experiences across visual, auditory, and haptic modalities.

Paper Playground draws inspiration from two physical programming projects. Dynamicland is a building-wide computing space that utilizes Lua-based RealTalk scripting for live coding via physical interaction, including paper with dot-encoded patterns [1]. Emphasizing transparency and simplicity, Dynamicland fostered collaborative programming among non-developers. Secondly, Paper Programs is an open-source, browser-based application, that offered a Dynamicland-like experience anywhere [8], utilizing the OpenCV library for computer vision [6] and JavaScript for writing programs. Colored dot sequences on paper are detected by a camera and the associated JavaScript is executed, typically altering the visual display projected onto the papers [5]. Though not actively maintained since 2019, Paper Programs allowed users with sufficient JavaScript knowledge to create interactive experiences. Recently, we began developing a significant extension of Paper Programs into the Paper Playground tool, building upon Paper Programs' existing web infrastructure, and strong foundation for colored-dot pattern detection.

3 PAPER PLAYGROUND CO-DESIGN TOOL

Paper Playground is an open-source co-design tool that enables users to write or build programs to associate with physical or virtual papers [7]. Each paper program is associated with a fixed colored dot sequence, and when this dot sequence is detected by Paper Playground, the program is executed. Paper programs can include visual and auditory (sound and speech) features, and creative use of papers and microprocessors can result in haptic and tangible experiences. Multiple paper programs can be run simultaneously and interacted with in real-time, creating opportunities for interactive paper-based designs with a wide range of complexity and duration. Shared databases of built paper programs allow for asynchronous collaboration, paper program re-use, and re-mixing. Access and full technical details are available from Ref. [12]. Here, we provide an overview of primary features of the tool in use within the context of a card game created through co-design with youth. The premise of the game is similar to Rock Paper Scissors. Players select Spell cards (papers) to place into the (central) play area to Battle. Each Spell card (Fireball, Ice Storm, Tornado) has strengths and weaknesses, and each Battle results in a win, loss, or draw.

Interfaces. Paper Playground has three interfaces: *Camera*, *Creator*, and *Interactive Display* (Figure 1). *Creator* is where users write or build programs to associate with physical or virtual papers. Shown in (Fig. 1.B), co-designers have built paper programs for gameplay and for each Spell card, and the Tornado Spell card is being edited. The *Creator* interface was designed based on abstracting the program building process to utilize the Model-View-Controller (MVC) architecture framework. This approach was chosen to allow novice developers to build programs organized into Models (logic/data), Views (display/interface), and Controllers (input/events) components, to augment the existing JavaScript code environment. Users build components, and can track connected components visually by arrows (Fig. 1.B., indicator b). Pre-made template programs are available to aid users in creating interactive projects without expert coding knowledge.

In *Interactive Display*, users can experience the outcome of their interaction designs, viewing and interacting either onscreen or through projecting this interface onto a physical tabletop. In our example game, *Interactive Display* (Fig 1.C) is shown on an upturned secondary monitor.

In *Camera*, users can calibrate their webcam, view the computer vision field to check for paper detection, and generate dot-coded papers to print. To execute and interact with the programs, co-designers can add or remove papers from view, move or rotate papers, change proximity of papers, or add markers (large paper dots) onto or around papers. In our game example, a webcam is mounted above the secondary monitor; Spell card detection is shown in *Camera* (Fig. 1.A). Programs update synchronously, enabling an active link between the papers' spatial arrangement and the associated code.

Interaction. Players interact in our game example by adding or removing Spell cards (papers) from the *Camera* view. When Spell cards are placed beside the play area, graphics and speech representing each card is triggered. To Battle, players add their cards to the central play area, triggering the visuals, sounds, and speech for their Spell and the declared result of the Battle. Program logic, game rules, and displays (such as artwork, sounds, etc.) can be changed live in *Creator*. For virtual or hybrid experiences, virtual 'Preview' papers (Fig. 1.A, indicator a) can be added to the *Camera* interface. Virtual paper programs can be executed like physical paper programs and can be moved, rotated, and split like physical paper.

Hardware Requirements. Co-design with a physical tabletop workspace requires Paper Playground on a desktop/laptop device and a webcam for detecting physical papers. The *Interactive Display* may be open on the device screen, however optional overlay of a visual display on the physical papers can be achieved with configurations such as placing papers on a secondary computer monitor (Fig. 2.A) or use of a projector (Fig. 2.B).

4 YOUNG DEVELOPER INSIGHT DURING PAPER PLAYGROUND CO-DESIGN

In this work, we were interested in gaining insight into what users of Paper Playground would want from a tangibly-coupled, multi-modal interaction design tool. Co-located collaboration is readily supported, but remote program creation is possible using a hosted database. We began investigating this question with a pair of youth participants engaged in a fully virtual co-design team with one or two non-developer facilitators. We share preliminary insights from co-design sessions aligned with 1) paper as interactive material, 2) remote collaboration, and 3) extensibility to physical computing.

Context. Two youth participants were recruited for participation in 12 sessions total over two semesters as a remote co-design team, part of an IRB-approved research project. Invitations requested participants to have at least some experience with web-based development, including basic JavaScript knowledge. The 1-1.5 hour duration sessions were focused on the use and design of Paper Playground. One middle-school-aged participant (C1) had experience with block-based programming and was beginning their education in web-centered languages (HTML, CSS, and JavaScript).

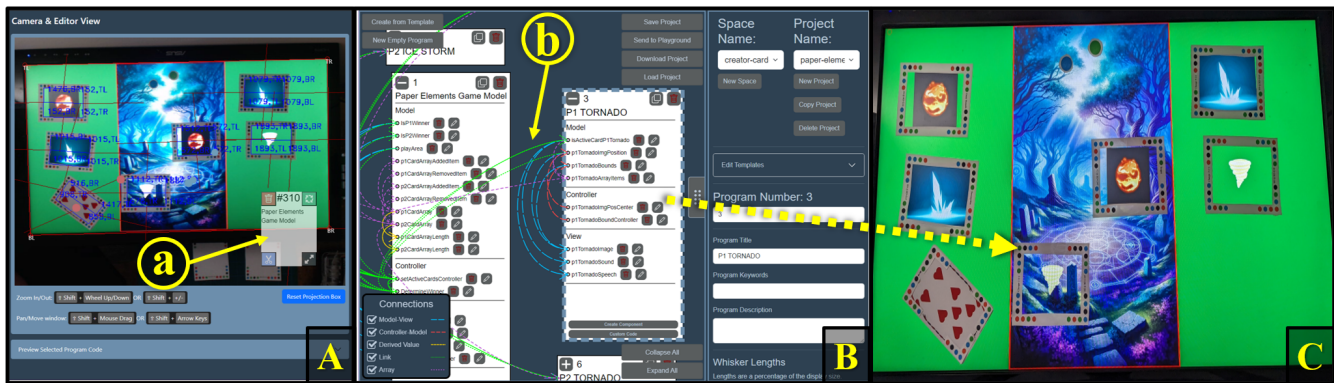


Figure 1: Paper Playground Interfaces. A) *Camera* where computer vision detection is calibrated and (a) virtual ‘Preview’ papers can be manipulated. B) *Creator* where programs are created and edited, and (b) relationships between components can be visualized. C) *Interactive Display*, shown on an upturned monitor, where users can experience their executed program(s). Dashed arrow indicates linked *Creator* program and physical paper.

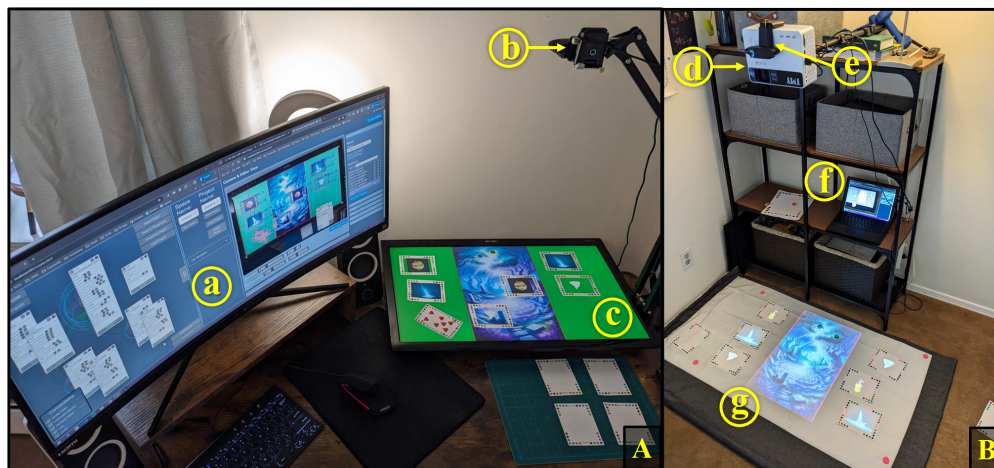


Figure 2: Example equipment setups. A) Tabletop configuration using (a) desktop device displaying *Camera* and *Creator*, with (b) a webcam on a microphone mount above (c) wallet-sized papers on an up-turned secondary monitor displaying *Interactive Display*. B) Floor configuration using (d) a suspended mini projector and (e) webcam. (f) A laptop device displays *Camera* and (g) half-letter-sized papers on a curtain to aid projection visibility.

The other, high-school-aged participant (C2) had been programming with Scratch since early childhood and had experience in languages pertinent to video game mod(ification) development, such as Java, C#, and JavaScript. Participation occurred over web-conferencing software and all participants were located in their residences. Participants were provided a miniature projector and webcam to use for setup, and used their own computer devices. Sessions started with simple tasks to help co-designers familiarize themselves with Paper Playground, such as making an image move with a piece of paper or triggering a sound. The participants alternated sharing their screen and project design responsibility varied across sessions. Based on participant interest in games, we centered most sessions around progressing the Rock-Paper-Scissors (RPS)

game described in Sec. 3, as the foundation for the more advanced game concepts co-designers brainstormed.

Paper Affordances in Project Design. Participants quickly associated paper programs with game cards (e.g., trading card games), generating many project ideas around placing papers in different zones of the detection areas (developed as a “Bounds” model component). They noted the collaborative ability to pass papers between players as a feature. The use of paper markers, similar to game tokens, appealed to the co-designers, with many game ideas featuring marker-initiated mechanics. For example, C1 suggested: “*And like some cards have effects where... they trigger something... You put a special type of marker on fire, which makes it stronger. And if it has a certain amount of markers, it can overpower different types of*

elements." Other project ideas included a Pong game with paper paddles, a speed sorting game, a planet simulation with papers as planets, a rearrangeable puzzle, drawing by sliding markers around, and a game where players manipulate markers in a projected game board (e.g., checkers).

Beyond the shared affordances of moving, rotating, and cutting papers, participants proposed novel ideas such as drawing markers onto papers as a permanent alteration, folding papers in half to reveal others, and applying papers to their bodies to encode their presence in the detection window. C1 was particularly excited about the potential for adding dots to other objects (e.g., "origami Millennium Falcon", paper folded spacecraft) for them to be recognized as programmable objects like the dot-encoded papers.

Participants noted the affordances of having virtual preview papers, as they free up physical play space for more spatially mapped physical papers. Additionally, virtual papers can be overlaid or positioned over physical papers while still executing their scripts, which is impossible with physical papers due to occlusion. As C2 noted, physical papers cannot be "stretched" like virtual papers to encompass entire regions. During remote co-design, virtual papers played a significant role during screen sharing, where seeing the other participants' physical workspace was sometimes difficult.

Using Paper Playground for Remote Collaboration. Using Paper Playground for remote co-design surfaced many challenges to address. We noted that without providing extensive equipment to participants in their homes, horizontal configurations were difficult to assemble or maintain. The intentional decision to limit equipment loans aimed to increase flexibility of tool setup by better understanding what equipment is necessary and what can be accomplished with available household items. It was not feasible to make semi-permanent changes to their environment for 1-hour sessions that typically occurred only once a week. Even temporary setups often could not be left in place and necessitated time-consuming setup each week. The need for troubleshooting at times, for example, addressing computer differences between participants, ambient lighting changes, or hardware such as limiting cable lengths, impacted effectiveness of collaboration.

Participants proposed innovative features to enrich remote collaboration with Paper Playground. As initially designed, the tool uses a hosted database to give all co-designers real-time updates to programs, but they only run locally. C1 suggested a shared or networked play area for multiplayer experiences: *"You could connect like two boards from far away and it would project your opponent's cards onto your board so people could play that weren't like together... Some sort of multiplayer would also make creating easier because two people could work on the same thing, and you could use that multiplayer to have multiplayer programs."* Another suggestion relied on generating unique papers that worked globally across projects that could be traded with others and used in their instance of Paper Playground.

Extensibility for Physical Computing Components. Both participants were particularly interested and engaged when suggesting features that aligned with their hobbies. There was a keen interest in integrating support for physical computing, such as microcontrollers. Integrating these devices poses a novel avenue for enhancing physical interaction with the papers.

They proposed using the devices to mechanically and dynamically manipulate the papers. They recommended connecting microcontrollers to *"move the papers around without moving them yourself"*, *"use a servo motor to flip a paper"*, or *"attach a paper to a robot"* to move it around the workspace. They also envisioned using Paper Playground to control the devices. For instance, C1 recommended supporting microphone input to create interactive experiences that responded to sound input, such as playing an animation in the projected space. Participants also suggested creating papers and markers to define zones for guiding a microcontroller-driven robot or connecting a gamepad to enable hybrid (paper/gamepad) control of the interaction. Additionally, C2 expressed interest in musical manipulation and brought up multiple examples. These included incorporating finer music controls within Paper Playground or directly connecting MIDI devices (e.g., an electronic piano) for more musical control in concert with paper manipulation and graphical projection.

5 DISCUSSION

Co-design sessions with youth participants highlighted several key insights into the desired affordances for a multimodal interaction design prototyping tool like Paper Playground: Participants demonstrated an inherent desire for flexibility and creativity in design tools. They were keen to explore the boundaries of what could be achieved through the combination of tangible and digital elements, manipulating physical objects in a digital environment, and creating zones for interaction. This underscores the need for design tools to support a wide range of interaction modalities and enable users to easily experiment with and iterate on their designs.

Challenges faced during remote collaboration sessions and the suggestion for a shared or networked play area for multiplayer experiences highlight the need for tools that support easy setup and use in a variety of environments, as well as facilitate remote, synchronous collaboration. Future work could enhance remote collaboration capabilities, potentially through cloud-based technologies.

Interest in extending Paper Playground to include physical computing components, reveals potential for more sophisticated interaction possibilities. This suggests that prototyping tools may consider incorporating easy-to-use interfaces for connecting and programming physical computing devices, broadening design and interaction possibilities.

Varied project ideas and the use of different modalities, like sound, movement, and touch, highlight a significant interest in creating designs that engage multiple senses. The enthusiasm for adding novel features, such as microphone input for sound-activated interactions or MIDI device integration for musical manipulation, points to a desire for tools that can support a wide range of sensory inputs and outputs. Design tools should aim to provide a rich set of features that enable creators to engage users through diverse sensory experiences, enriching the interaction design landscape.

6 CONCLUSIONS

We presented insight from young developers engaged in remote co-design sessions for the multimodal interaction design prototyping tool, Paper Playground, where we found a desire for design tools

that are flexible, collaborative, multimodal, and inclusive of physical computing. We aim to leverage these and future insights to empower designers to bring children into direct collaboration during interaction design, leading to more inclusive and effective multimodal experiences. Future enhancements, inspired by this work, will focus on augmenting remote-friendly virtual experiences with additional networking, integrating Bluetooth communication with microcontrollers, and enhancing usability for children via code abstraction, in partnership with youth co-designers and with the open-source community [7].

ACKNOWLEDGMENTS

We thank all co-designers for their time and perspectives, and JP Posma for his encouragement to build upon Paper Programs. This material is based on work supported by the National Science Foundation under IIS-#2119303.

REFERENCES

- [1] The Dynamic Medium Group. 2014. Dynamicland. <https://dynamicland.org/>
- [2] Felix Lauber, Claudius Böttcher, and Andreas Butz. 2014. PapAR. *Adjunct Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (2014), 1–6. <https://doi.org/10.1145/2667239.2667271>
- [3] Emily B Moore, Brett L Fiedler, and Taliesin L Smith. 2022. Co-Design: Success, Challenge, and an Inclusive Design Future for PhET Interactive Simulations. *Workshop Paper for the Co-Designing with Mixed-Ability Groups of Children to Promote Inclusive Education Workshop* (2022). <https://www.inclusiveeducation.tech/wp-content/uploads/2022/06/Moore-Workshop-Submission-FINAL.pdf>
- [4] Emily B Moore, Taliesin L Smith, and Jesse Greenberg. 2018. Keyboard and Screen Reader Accessibility in Complex Interactive Science Simulations: Design Challenges and Elegant Solutions. *Lecture Notes in Computer Science* (2018), 385–400. https://doi.org/10.1007/978-3-319-92049-8_28
- [5] Mozilla. 2023. Canvas API. https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API
- [6] OpenCV. 2024. Open Source Computer Vision Library. <https://opencv.org/>
- [7] Paper Playground. 2023. Paper Playground Documentation. <https://phetsims.github.io/paper-land/>
- [8] Jan Paul Posma. 2017. Paper Programs Github Repository. <https://github.com/janpaul123/paperprograms>
- [9] Hanna Schraffenberger and Edwin van der Heide. 2016. Multimodal augmented reality: the norm rather than the exception. *Proceedings of the 2016 workshop on Multimodal Virtual and Augmented Reality* (2016). <https://doi.org/10.1145/3001959.3001960>
- [10] Reinhard Sefelin, Manfred Tscheligi, and Verena Giller. 2003. Paper prototyping - what is it good for? *Extended Abstracts on Human Factors in Computing Systems - CHI '03* (2003). <https://doi.org/10.1145/765891.765986>
- [11] PhET Interactive Simulations. 2023. PhET Inclusive Design. <https://phet.colorado.edu/en/inclusive-design>
- [12] PhET Interactive Simulations. 2024. Paper Playground GitHub Repository. <https://github.com/phetsims/paper-land>
- [13] Taliesin L Smith and Emily B Moore. 2020. Storytelling to sensemaking: A systematic framework for designing auditory description display for interactives. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)* (2020), 1–12. <https://doi.org/10.1145/3313831.3376460>
- [14] Carolyn Snyder. 2003. *Paper Prototyping : the fast and easy way to design and refine user interfaces*. Morgan Kaufmann.
- [15] Brianna J Tomlinson, Bruce N Walker, and Emily B Moore. 2021. Identifying and evaluating conceptual representations for auditory-enhanced interactive physics simulations. *Journal on Multimodal User Interfaces* 15 (2021), 323–334. <https://doi.org/10.1007/s12193-021-00365-z>
- [16] Alexander Wiethoff, Hanna Schneider, Julia Küfner, Michael Rohs, Andreas Butz, and Saul Greenberg. 2013. Paperbox. *Proceedings of the 9th ACM Conference on Creativity & Cognition* (2013), 64–73. <https://doi.org/10.1145/2466627.2466635>
- [17] R. Michael Winters, E L Harden, and Emily B Moore. 2020. Co-designing accessible science education simulations with blind and visually-impaired teens. *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility* (2020), 1–4. <https://doi.org/10.1145/3373625.3418025>