Strawbies: Explorations in Tangible Programming

Author 1, Author 2, Author 3
Anonymous
Affiliation
Address

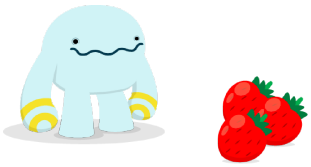
email1, email2, email3

ABSTRACT
In this demo we present Strawbies, a real-time tangible programming game designed for children ages 5 to 10. Strawbies is played by constructing physical programs out of wooden tiles in front of an iPad. This interaction is made possible with the use of an Osmo play system that includes a mirror to reflect images in front of the iPad through the front-facing camera. We combined this system with the TopCodes computer vision library for fast and reliable image recognition. Here we describe a set of principles that guided our iterative design process along with an overview of testing sessions with children that informed our most recent instantiation of Strawbies.

Categories and Subject Descriptors
H.5.m [**Information interfaces and presentation (e.g., HCI)**]: Miscellaneous.

Keywords
Children; tangibles; games; strawberries.

1. INTRODUCTION
In this demo we present Strawbies, a tangible programming game designed for children ages 5 to 10. Strawbies is an iPad app that features Awbie, a character that children guide on a quest for strawberries through a virtual world using wooden programming tiles (Figures 1, 2, 3). Our system combines the TopCode computer vision library [6] with the Osmo play system [12] to allow for real-time recognition of programs that children construct on a flat surface in front of the iPad (Figure 1). This results in an inexpensive, engaging, and portable tangible programming environment. Our design process has involved three major revisions of the game that we tested with children, parents, and teachers (see Figure 4).



Conference'10, Month 1–2, 2010, City, State, Country.

2. DESIGN PRINCIPLES
Our design builds on prior work in tangible interaction and children's programming environments and was guided by the following eight principles:

1. Inviting: Through the use of tangible programming tiles, we hoped to create an inviting experience that would draw children into collaborative play. The use of tangibles increases the visibility of game play, allowing it to move beyond the screen and spill out into the real world. In our testing sessions, children would often notice the game from across the room.

2. Playful and Open-Ended: In the spirit of many children's programming environments [2, 4, 7, 10, 11, 13, 15], we wanted to support children's open-ended exploration. This led us, over time, to an open-world style of game in which Awbie is free to roam around an infinite, randomly generated landscape. While the game is loosely structured around the task of guiding Awbie to eat and grow strawberries, children are free to playfully explore the world in any way they see fit.

3. Simple+Complex: As with prior work on the development of programming environments for informal learning settings [7, 11],

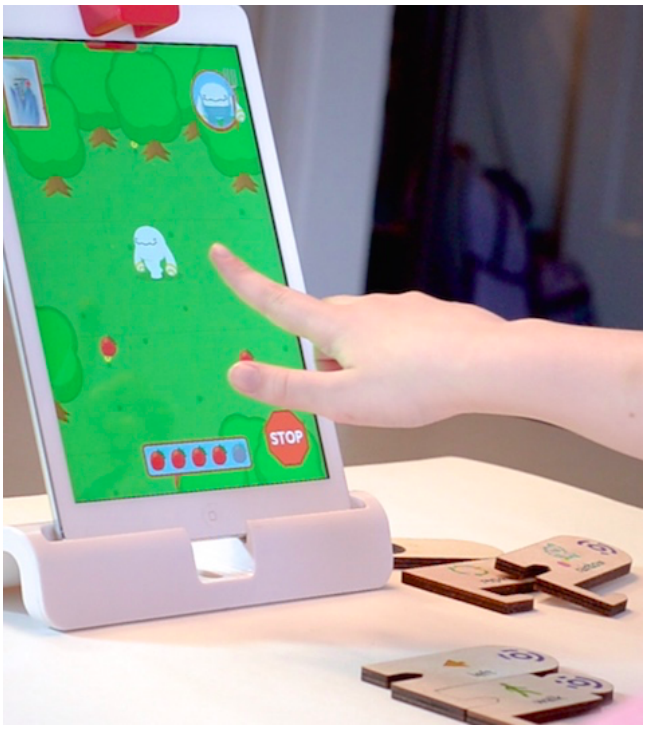


Figure 1. Child creates a program with Strawbies using tangible tiles, an iPad, and the Osmo attachment.

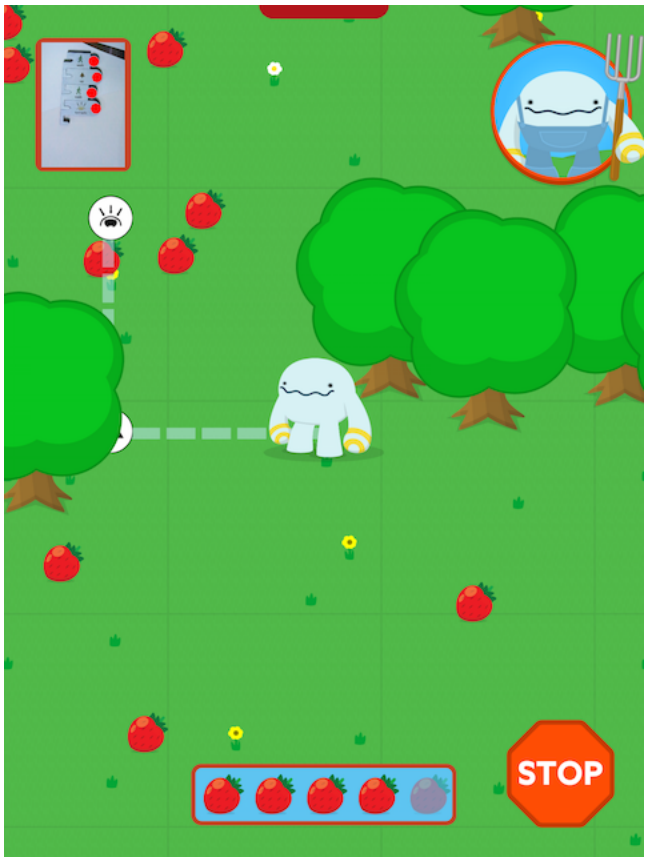


Figure 2. Accommodations to early childhood development include: a preview of Awbie's projected path, matching symbols on both tiles and screen, and a screen-in-screen display of the zone of interactivity.

we wanted to create a system that was simple enough for children as young as five years old to figure out on their own with little or no instruction. However, we had to balance this goal of simplicity against the ability to create relatively sophisticated programs that would lead to complex behavioral outcomes for Awbie.

4. Fluid and Responsive: Tangible programming systems that rely on computer vision have typically had to make tradeoffs between inexpensive materials, portability, and real-time interaction [7]. Achieving continuous recognition of tiles usually requires an overhead camera fixture or an interactive surface with built-in camera hardware. It is possible to instead use a mobile device like an iPad, but this requires a point-and-shoot style of interaction (e.g. [8]). However, with the Osmo's use of a mirror in front of the camera, it is possible to create fluid, real-time interaction with a mobile device and low-cost tangible materials.

5. Developmentally Appropriate: We wanted to ensure that the content of our game was appropriate for our target audience and that the game play was aligned with children's cognitive, perceptual-motor, and social ability. To ensure that we achieved this, we conducted multiple rounds of testing with children in our target age range.

6. Pedagogically Aligned: One of the greatest challenges facing the adoption of developmenally appropriate technology in classrooms is that teachers must feel comfortable and confident with the materials [1]. This includes making sure that technology aligns with the pedagogical philosophy of early childhood educators, a philosophy that emphasizes rich sensory-motor experiences, open-ended exploration, and social interaction. We see the use of tangible technology as an excellent way to introduce computational thinking activities in a way that evokes familiar cultural forms of teaching and learning [5].

7. Social: Along the lines of the previous design principles, we sought to design an activity that was inherently social in nature. The activity should invite multiple children to create programs together through collaborative play. One hallmark of tangibles is that it creates space for multiple hands and bodies and allows children to easily distribute their activity among multiple players.

8. Adaptable: The difference in ability of a five-year-old and a ten-year-old child can be dramatic. In developing the game, we sought to create an engaging childhood activity, that could also be enriched with the proximal assistance of an adult or older child. For example, for pre-literate children, it might be necessary to read text out loud to help children interpret meaning and understand the graphical elements of the tiles. Or, teachers might choose to remove some of the tiles (such as the event triggers) from the activity until a child has mastered the more basic concepts. This is one example of the flexibility that tangibles afford—no customization of the software is necessary to instantly adapt to the abilities of different children.

3. RELATED WORK

Our project builds on a long and rapidly growing tradition of programming environments for young children [3, 4, 5, 10, 11, 13, 14, 15, 16, 17, 18, 19] (see also [9] for an excellent account of older work dating back to the 1970s). There is also growing momentum around the idea of supporting computational literacy activities throughout K-12 education, starting at the earliest grade levels. Our project contributes to this space by blending the flexibility, portability, and practicality of tangible programming with an open-ended formatting, which enables our game to be highly responsive to the spontaneity in early childhood learning.

4. DESIGN ITERATIONS

Our design process involved several major revisions in which we explored different approaches for the iPad app and the tangible programming tiles. For the programming tiles, we wanted to make sure that they were easy to assemble and disassemble while allowing for connected programs to be dragged around the table without falling apart. We included parameters, loops, and if/then logic. The tiles were split into several categories: verbs, adverbs, and units of measurement. Verbs slide in and out of other verbs conveniently, while units and adverbs attach to verbs like puzzle pieces. Each string of verbs could start with an `Always` tile (for looping behavior) or a `When` tile (for if/then logic). Due to the limited field of vision of the Osmo, we chose to use event-driven logic over sequential logic.

Version 1: Puzzle Based

Our first game iteration consisted of a series of puzzles, much like code.org's Hour of Code series. From testing, we found that our players were easily frustrated by the game—especially the lower end of our target age range. Puzzles had only two or three possible solutions, and failure meant that players had to restart the puzzle. We felt that the combination of the Osmo and tangible

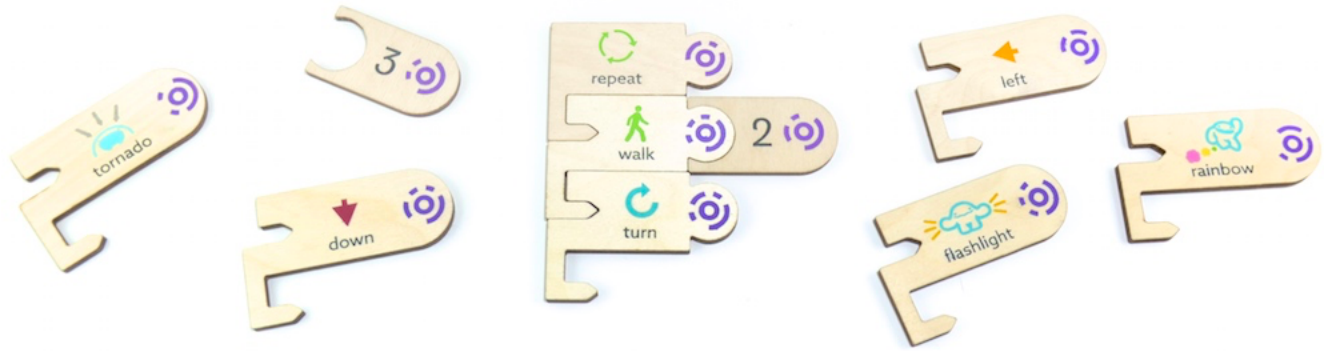


Figure 3. Collection of programming tiles available to guide the character in search of stawberries.

programming pieces could open up new potential for playful and open-ended explorations and discovery.

Our screen layout was also influenced by ScratchJr. and other tablet coding games that do not use tangible pieces. Half of the screen was dedicated to showing what the iPad was reading (Figure 4, left). This was unnecessary, as the physical code tiles in front of the iPad is an extension of the UI.

Version 2: Open World Environment

Realizing that we did not need to dedicate screen real estate to the physical tiles opened new possibilities for design. Responding to user testing, we created an open world game in which the goal was to harvest strawberries. Because we wanted a game that reacted in real time to kids' programs, every sequence of tiles that was recognized by the system would be immediately added to a pool of actions. The avatar performed actions randomly chosen from the pool.

The structure of the game was provided by obstacles: water, trees, and bats. Trees obstruct movement, while contact with water or bats ended the game. Players could add events triggered by obstacles ("When water is in front, turn 90 degrees right and walk"). This version was better received by players, but there were a few issues. Tiles had too many qualifiers, a line of code could require up to three different tiles. The game was too challenging for children and even for adults. Also, collecting strawberries was not a strong enough motivator to engage players in deeper explorations of programming concepts.

Version 3: Refining the Open World Game

In this version, we kept the open world concept and made additional refinements to improve fun and playability. We started by simplifying the action tiles and removing required qualifiers. A player could instead attach numbers to verbs to increase the number of times an action is performed. We added room for graphical symbols on each tile to increase the accessibility for young children. We also made looping an explicit tile that had to be used to cause actions to be repeated more than once. To make the game less frustrating and more fun, we added three special action tiles: rainbow, tornado and flashlight. The rainbow flies Awbie to a different place in the world, the tornado consumes all the strawberries in a short radius, and the flashlight scares away rats that are trying to steal Awbie's strawberries. To give users an idea of progression and ownership, we added a separate game-within-a-game. As players collect strawberries, they can start to grow fruit bearing plants in a garden. A separate screen shows garden rows that slowly fill with plants that Awbie can harvest.

We paid close attention to player confusion regarding directionality. These issues negatively impacted the player's confidence to solve problems, strategize, and create sequences. This led us to numerous versions of arrow styles and label terminology. New arrow designs lessened player confusion during turning/rotating. Arrow iterations were combined with careful evaluations of the subtle differences in the terminology related to position, direction and turning.

5. EVALUATION

We brought the game for six play sessions at two local schools (PreK-8 and PreK-12). In each school we tested two types of environments: a closed off space with two children at a time, and an open environment with many kids coming and going. Our sessions were divided into 20-30 minute slots. The ability to find, collect, harvest and protect strawberries proved rewarding and well-rounded. The harvesting and scorekeeping farm became a place for lively and quick-paced spontaneity, in a game of thoughtful strategy. Most play testers wanted to play more when told that their time was up.

Player phrases such as "Can you pass me the tornado?" or "Don't do that yet, we're going to the strawberry" revealed their collaborations and social negotiations. Comments such as "We need to bring him down, do you see a down block?" revealed the children were working on common goals. Typical challenges in turn-taking and sharing were also seen such as holding back another's hand or moving an undesired piece outside of the camera range. This is important social work that takes place in childhood and in classroom spaces.

When the play sessions were in open spaces, children would come up and ask what the tiles were. New children joined in, learned from their peers, and then offered strategies of their own. Groups of up to five students eventually came to form around the iPad. Even though there were five students gathered around one iPad, none of the children seemed disengaged. There were always pieces in front of them that they could play with, or slide in and out of the iPad's view. The children's sharing and discussion of

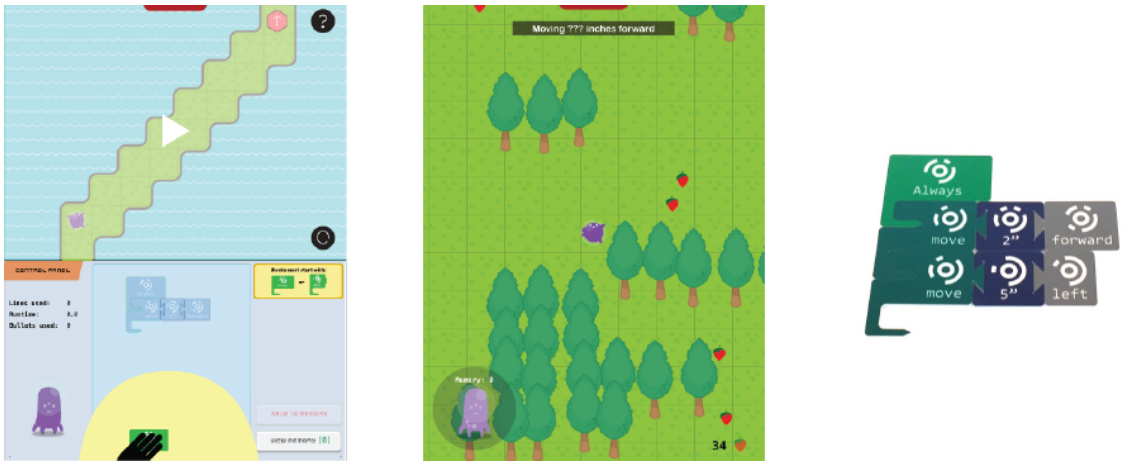


Figure 4. We have developed three major revisions of the game that we have tested with children, parents, teachers.

tiles revealed wide-ranging differences in their strategies that we found encouraging.

Different children interacted with the tiles in different ways. One boy in Grade 3 turned the tile to face the iPad screen, so that the word "Walk" was facing Awbie on screen, and not himself. Another child thought that tapping on the tile's TopCode would send a signal to the iPad. However, with some hints (which could be replaced by tutorials), the children understood how the tangible tiles function as sequential commands. Young children (ages 4-5) were also able to remember the names of tiles such as rainbow and walk, despite not being able to read them initially.

Many children used only one or two tiles at a time, although we sensed that they had more elaborate chains of actions in mind. Placing a "stop sign" button on the screen allowed time for kids to plan out their strategies and search for desired tiles. We are experimenting with additional ways to encourage players to construct longer chains of tiles.

6. ACKNOWLEDGEMENTS

Omitted for blind review.

7. REFERENCES

[1] Cuban, L. (2009). *Oversold and underused: Computers in the classroom*. Harvard University Press.

[2] Chawla, K., Chiou, M., Sandes, A., & Blikstein, P. (2013). Dr. Wagon: a 'stretchable' toolkit for tangible computer programming. In *Proc. Interaction Design and Children* (IDC'13), 561-564.

[3] diSessa, A. (2000). *Changing Minds: Computers, Learning, and Literacy.* MIT Press.

[4] Flannery, L.P., Silverman, B., Kazakoff, E.R., Bers, M.U., Bontá, P., & Resnick, M. (2013). Designing ScratchJr: Support for early childhood learning through computer programming. In *Proc. Interaction Design and Children* (IDC'13), 1-10.

[5] Horn, M.S. (2013). The role of cultural forms in tangible interaction design. In *Proc. Tangible, Embedded, and Embodied Interaction* (TEI'13).

[6] Horn, M. TopCode: Tangible Object Placement Codes. http://users.eecs.northwestern.edu/~mhorn/topcodes/

[6] Horn, M.S., Crouser, R.J., & Bers, M.U. (2012). Tangible interaction and learning: The case for a hybrid approach. *Pers. and Ubiq. Computing, 16*(4), 379-389.

[8] Horn, M.S., AlSulaiman, S., Koh, J. (2013). Translating Roberto to Omar: Computational literacy, stickerbooks, and cultural forms. In *Proc. IDC'13*, 120-127.

[9] McNerney, T. (2004). From turtles to tangible programming bricks: explorations in physical language design. *Pers. and Ubiq. Computing, 8*(5), 326-337.

[10] Montemayor, J., Druin, A., Chipman, G., Farber, A., & Guha, M.L. (2004). Tools for children to create physical interactive StoryRooms. *Computers in Entertainment: Educating children through entertainment Part II, 2*(1).

[11] Oh, H., Deshmane, A., Li, F., Han, J. Y., Stewart, M., Tsai, M., & Oakley, I. (2013). The digital dream lab: tabletop puzzle blocks for exploring programmatic concepts. In *Proc. Tangible, Embedded and Embodied Interaction* (TEI'13)

[12] Osmo. https://www.playosmo.com

[13] Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.

[14] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Sliverman, B. & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM, 52*(11), 60-67.

[15] Schweikardt, E., & Gross, M.D. (2008). The robot is the program: interacting with roBlocks. In *Proc. Tangible and Embedded Interaction* (TEI'08), 167-168.

[16] Sapounidis, T., & Demetriadis, S. (2013). Tangible versus graphical user interfaces for robot programming: exploring cross-age children's preferences. *Personal and ubiquitous computing, 17*(8), 1775-1786.

[17] Sipitakiat, A., & Nusen, N. (2012). Robo-Blocks: designing debugging abilities in a tangible programming system for early primary school children. In *Proc. Interaction Design and Children* (IDC'12), 98-105.

[18] Weintrop, D., & Wilensky, U. (2013). RoboBuilder: A Computational Thinking Game. In *Proc. ACM Technical Symposium on Computer Science Education*, 736-736.

[19] Wyeth, P. (2008). How young children learn to program with sensor, action, and logic blocks. *Journal of the Learning Sciences, 17*(4), 517-550.