Constructionist Gaming Beyond the Screen: Middle School Students' Crafting and Computing of Touchpads, Board Games, and Controllers

Yasmin B. Kafai
University of Pennsylvania
3700 Walnut Street
Philadelphia, PA 19106
001-215-746-3468
kafai@upenn.edu

Veena Vasudevan
University of Pennsylvania
3700 Walnut Street
Philadelphia, PA 19106
001-215-746-3468
veenav@gse.upenn.edu

ABSTRACT
Approaches to constructionist gaming—students making their own games for learning through programming—have mostly focused on screen designs. Hybrid crafting approaches that integrate crafts with digital components can extend game making beyond the screen and provide new opportunities for learning about computational concepts, skills, and perspectives. In this paper, we report on a series of workshops with middle school students (ages 11-14 years) who used Makey Makey, Play Doh, textiles and other materials to craft touchpads, augmented board games, and wearable controllers for their Scratch games. We examined students' approaches to computing and crafting in their onscreen and off screen designs. We discuss in which ways constructionist gaming can benefit from extending their designs into the physical world and what moving constructionist gaming beyond the screen has to offer for K-12 programming instruction.

Categories and Subject Descriptors
K.3.2 [Computers and Education]: Computer and Information Science Education – Computer science education; K.8.0 [Computers and Education]: General – Games.

General Terms
Human factors

Keywords
Game Design, Maker Activities, Controllers, MaKey MaKey, Scratch, Tangible Designs, Wearables

1. INTRODUCTION
With concerted efforts underway across the globe to bring computing into primary and secondary schools the focus is on developing, implementing, and researching novice programming tools and construction kits that can lower the entrance barriers to coding well as comprehensive curricula and effective teaching approaches that can introduce computing across a variety of contexts to students in different grade levels. Among these instructional efforts, programming your own games for learning—or constructionist gaming [11]—has become one of the more popular approaches for introducing students into programming [13]. A recent meta-synthesis identified over 350 articles reporting on children's learning while programming games and identified a range of academic and motivational benefits for learners [4].

For the most part though constructionist gaming efforts have focused on screen designs while commercial gaming has long since moved beyond the screen into the physical world [5] with new genres of controllers such as the Wii remote for Nintendo games, the drum interface for Rock Band, the dance mat for Dance Dance Revolution, and augmented board games such as Monopoly with Electronic Banking. More recent developments even propose to extend controller designs into costumes and wearables. The growing availability of new construction kits for wearable and physical computing (for overviews, see [1; 2]) have made the design of such gaming interfaces accessible to even novice designers. For instance, construction kits such as Makey Makey [21] facilitate hybrid crafting—approaches that integrate coding and crafting with digital components to further learning and creative expression but have not been extensively studied as a context for developing key concepts, practices, and perspectives of computational thinking or participation [12].

In this paper, we report on a series of studies that investigated this new territory of constructionist gaming beyond the screen by having middle school students design interactive touchpads, augmented board games, and wearable controllers. In each of these studies, students not only designed or remixed games in Scratch [20] but then also proceeded to craft and code physical extensions using the Makey Makey construction kit [21] and various other construction materials. In our analyses, we focused on assessing computational concepts and practices used in their tangible game designs, and changes in their perspectives addressing the following questions: (1) What are computational concepts that students can learn in constructionist gaming beyond the screen? (2) In which computational practices do students engage when coding and crafting? and (3) What are their computational perspectives? In the discussion, we examine in which ways introductory programming activities in primary and secondary schools can benefit from extending constructionist gaming beyond the screen into the physical world.

WiPSCE '15, November 09-11, 2015, London, United Kingdom © 2015 ACM. ISBN 978-1-4503-3753-3/15/11…$15.00

DOI: http://dx.doi.org/10.1145/2818314.2818334

2. BACKGROUND

Within computer science education, efforts have focused on helping novice programmers in their initial steps by designing special tools, creating social supports, and offering contexts that can situate the learning of coding within a personally meaningful and relevant context [20]. Digital games assume here a special place because they were and still are one of the most prominent applications through which children have their first and most extensive experiences with digital technologies [15]. The first study that investigated constructionist gaming took place with a class of fourth graders who created computer games that taught younger students in their school about fractions and found significant benefits in learning programming and math when compared to other students in their schools [11]. It is this combination of personal relevance coupled with learning outcomes and motivational benefits that led making games for learning programming—or constructionist gaming—to become a driving approach in K-12 programming [14]. A recent meta-synthesis conducted by Campe and Denner [4] counted more than 350 papers published on the topic of what students in K-12 can learn by programming games. Of the 169 papers that were included for a more focused analysis, they found that constructionist gaming was most popular in middle school grades 6-8, followed secondary grades 9-12, and then primary grades.

These preliminary findings visibly demonstrate the widespread and successful use of constructionist gaming in primary and secondary schools over the last two decades. By expanding game design beyond the screen we not only leverage these successes but also connect them to new materials, settings, and displays [5; 7; 10] that have become commonplace in today's gaming culture. Designing wearable controllers, interfaces and board games where novice programmers can manipulate objects not only on the screen but also in the physical world provide compelling new applications in gaming. When designing tangible interfaces, students not only are crafting physical artifacts but they also are writing programs for the digital artifacts to control interactions on and off the screen [18]. These intersections between crafting and computing provide the first steps into the world of physical computing [21] and can introduce students to key concepts, practices, and perspectives in computing [3]. The workshops we conducted afford us the opportunity to examine their perceptions because of the highly relevant and social role games play in many students' lives. Expanding game design beyond the screen also combines computing with crafting, the high and low of technology, and thus could broaden not only participation but also perceptions of computing. Taken together, the focus on computational concepts, practices, and perspectives allows us to examine students' understanding of core CS concepts. It also reveals the generative thinking practices students developed through the process of bringing their hybrid designs to fruition alongside students' perceptions of computing.

3. CONTEXT

3.1 School and Students

The three workshops took place in the same K-8 public school situated in a metropolitan city in US northeastern state over a period of three years. The participants in our workshops were representative of the diversity in the school: 39.2% white, 23.9% African American, 18.4% Asian, 6.7% Latino, less than 1% Pacific Islander and American Indian and 10% other. A total of 28 middle school students (14 boys and 14 girls, ages 11-14) consented to participate in the studies. Students joined the game design workshops as part of their choice elective that met twice a week for 50 minutes each. The total number of hours varied between 8 hours (for touchpads), 12 hours (for augmented board games) and 22 hours (for wearable controllers). The workshops usually had one main instructor (the second author, a graduate student) who designed and facilitated class activities and conducted data collection in partnership with the technology teacher at the school.

3.2 Design of Workshop Activities

Each of the workshops had a different focal theme but they all used Scratch [21], a multimedia programming environment, and the Makey Makey [22] construction kit to connect physical design elements to on-screen design elements in Scratch games. The Makey Makey plugs into the computer using a simple mini-USB to USB cord (in this example, the red upper cable in Figure 1). Another cord (the black left one in Figure 1) is a connection to earth, which helps to ground the circuit (giving the electricity a place to go). Finally, there are the bottom cords that connect to the conductive objects. The MaKey MaKey gives student designers a way to use conductive objects to replace, for instance, the up, down, right, left and spacebar keys, and in the process to learn about conductivity of everyday objects like fruits, Play-Doh and, aluminum foil.

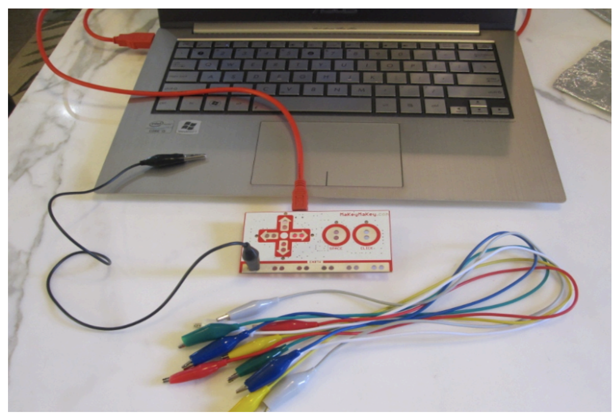

Figure 1: **Setup of Makey Makey and Computer.**

The first workshop focused on designing touchpads for Scratch video games. Students in this workshop remixed or designed their own games in Scratch. The remixes built on existing starter projects available in Scratch. The second workshop focused on students designing augmented board games. Finally, the third workshop focused on students remixing Scratch games and designing wearable controllers with Makey Makey and various conductive materials. Given that each workshop incorporated tangible designs in a unique way the curriculum varied in terms of order however, the activities always included: orienting students to Scratch, having them prototype with MaKey MaKey and using a craft and conductive materials to create tangible designs. To acclimate students to the Scratch environment and key concepts we always devoted a portion of workshops to creating simple designs. We also incorporated opportunities for informal in-class feedback and in the tangible controller and wearable controller workshop, we also ended with an arcade. As mentioned above the three workshops touchpad controllers, augmented board games and wearable controllers had 10, 8 and 20 class sessions respectively. The available time had an impact on the complexity of projects that could be designed.

3.3 Data Collection and Analysis

Across our three workshops we documented students' game design processes and interactions over time in field notes and video recordings. In addition we collected each student's final physical artifacts (e.g., touchpads, wearable controllers, and board games) and their final Scratch code. Finally, we interviewed each student who participated in the workshops. For the physical artifacts we first used a descriptive analysis to investigate the aesthetic choices youth made as well as the relationship of physical artifacts to the actual Scratch code. We also used Brennan and Resnick's [3] framework for computational thinking as a way to analyze the concepts and practices youth implicitly or explicitly experienced in the game making activities across all three projects. In analyzing youth's video games we identified the computational concepts utilized in students' remixed code. In addition we determined which computational practices students experienced as part of the overall game and controller design process. Finally we used two-step open coding to analyze students' interviews.

4. FINDINGS

4.1 Crafting Designs

In the first workshop, we asked students to make interactive touchpads for their Scratch games. In general, controllers fell into two groups, those that incorporated both aesthetic and functional elements, and those that seemed mainly functional. The interfaces in the aesthetic group included detailed components matched to the sprites (characters) in their video games. The interactive touchpads in the functional group were mostly built from solid colors of Play-Doh that represented the directions that the sprites in the games could move (e.g., up, right, and left) but did not aesthetically match their accompanying Scratch game designs. An example of a student from the aesthetic group was Amani, a sixth grade girl, remixed a Scratch game where the objective is for the player to get the zombie to consume brains. To remix the original project, Amani found a zombie graphic, used Michael Jackson's *Thriller* as her background music and downloaded an image of brains. She made the game complex by adding both good (pink) and poisonous (green) brains that added or removed points respectively. She also correlated the size of each of the brains to a point value so the ones with higher point values were larger and those with lower point values, smaller in size and provided a high score so there was a way to win the game (see top, Figure 2).

Amani also iterated on her touchpad controller (see bottom, Figure 2) because in her initial design the moisture from the Play-Doh began to seep into the paper, thus causing short circuits and stopping her controller from working. In her final version she tweaked her design to include three separate buttons (right, left and space), which resolved the short-circuit issues. When later asked about her design process, Amani mentioned that her rationale for pink buttons was to match the color of the brains in her Scratch game and the arrows were incorporated for usability; the arrow keys and rectangular buttons she constructed out of Play-Doh mirrored a keyboard design. When we analyzed Amani's remix to the original starter code the features she added like the good/bad brains and a high score demonstrate how youth engage with computational concepts like *parallelism, event handling* and *data* (e.g., keeping score) as well as practices like *being iterative and incremental* and *remixing.*

In our second workshop students designed augmented board games. Augmented board games are traditional board games that have integrated digital components like digital dice, playing cards or other features. To augment games we gave youth Scratch starter code for digital dice and other features (e.g. playing cards). Once they designed their initial board games they selected digital components to integrate into their existing games. Five groups (comprised of 2-3 students) designed unique board games. We categorized the five board games into two groups: start-to-finish games (n=3) and Monopoly games (n=2). Start-to-finish games are won by a player that gets to the end first, whereas Monopoly-style games are defined as those that do not have a definitive end, but rather require players to survive or acquire the most resources (e.g., money) to win. The five games were distinct in terms of theme: *Road Trip,* a game where players travel across the country, *The Farm Game* is a trivia game with a country aesthetic, *That Spot*, is named for the red and green spots that indicate good or bad luck in the game, *School Boy 2* allows "gangsters" and "good guys" to battle for money and, *Philadelphia Gangsters,* a city-specific themed gangster game similar to Monopoly.

Figure 2: Amani's Scratch screen (top) and touchpad designs (bottom): (1) Initial design (2) More detailed design that short circuited (3) final design with touch pad pieces separated.

A group comprised of three middle school girls that had the most successful augmented board game design created a start-to-finish board game entitled *Road Trip* where players had to navigate challenges they devised before reaching the END square. To operationalize these challenges, they had three kinds of cards: Trouble (yielding a bad consequence e.g. you're car broke down, skip a turn), Danger (trivia questions) and Lucky (yielding a good consequence, e.g. you got a free tank of gas, advance 3 paces). The girls enthusiastically drafted a sample game in their first brainstorming session (see Figure 3). Then, they transitioned to their playtest board being intentional about using color and images to make the game play experience more authentic. During play testing the girls observed that their board was too simple and too short in terms of the number of actual spaces on their board (which they determined because their peers were able to "win" or "lose" the game quickly), so they chose to make their final game board both longer and more complex (see Figure 3, bottom image).

It was at this time the group initially augmented one part of their board—choosing only to digitize their dice, using the Scratch code we had provided for them, and then later the playing cards (e.g., trouble, danger, lucky) that were originally paper cards (see Figure 3). In order to do so, they had to determine where to integrate hotspots and remix the Scratch code for playing cards (that we had provided). In their final version, the girls maintained their original symbols for danger ("X"), trouble ("!") and lucky cards (a picture of a shamrock) and also made sure that the hot spots were aligned. On their final game board, they incorporated space for their digital dice and a connection to earth (or a connection to ground the circuit required by MaKey MaKey), drawing an actual dice and the aesthetic of earth as visual cues for each of these features. They used decorative tape to embellish their board game and eventually printed their rules (see Figure 3, bottom image).

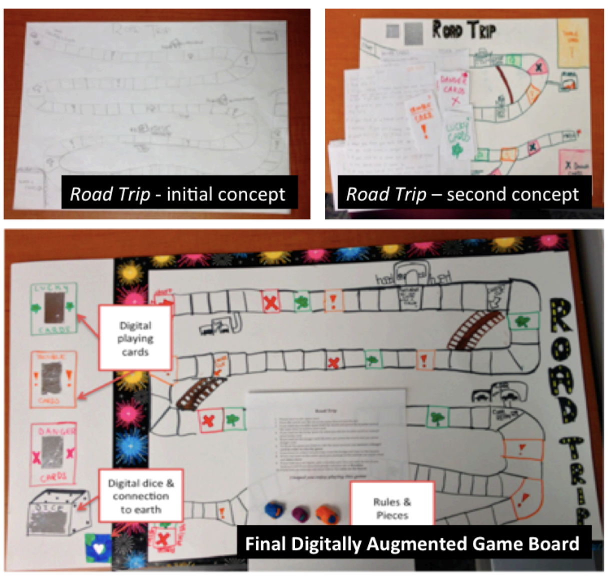

Figure 3: Road Trip Board Game Design Iterations.

Finally, in the third workshop, students designed wearable controllers for Flappy Bird Scratch games. The game's premise is to keep a bird afloat in the air while dodging a set of scrolling challenges. We provided students with a simplified Scratch game code including the scrolling background and the gravity effect (for the bird) for their remixes. We also shared two prototypes with students of wearable controllers and provided opportunities to test them out. Of the ten students who consented to research, eight designed wearable controllers and two designed non-wearables. For instance, Sara's project was inspired by Michael, a member of the boy band *Five Seconds of Summer*. Sara remixed a more sophisticated version of Flappy Birds in Scratch. Instead of a bird, she replaced it with an image of Michael with angel wings, then, in lieu of traditional pipes Sara chose scrolling obstacles like a little bat with a camera to represent the paparazzi that might follow the singer and objects like donuts, bananas and other things that would also act as deterrents to Michael staying afloat. Another twist she added to the game was to make it possible for Michael to stand up or float. For the soundtrack, she used a song by the group Five Seconds of Summer. The sprite (or object) that represented his character was a picture of him that she had then remixed using the Scratch paint app. She created all of the code for her scrolling challenges using trial and error.

When it came time to create a wearable controller Sara opted to make something similar to the prototype we shared with students. First, she sewed a felt hand using white felt by cutting a pattern out then, sewing it together. Next, she used red feathers to symbolize Michael's hair by using a hot glue gun to affix them to the top of her controller (see Figure 4, bottom). Then she added a face by affixing "googly" eyes using a hot glue gun and drawing a small mouth using a fabric marker. Black "pants" completed the look. Once the main construction was completed Sara began testing out the game with her controller. She temporarily pinned electronic fabric to the top and thumb section of her glove design to see if her code worked and if the flapping motion made sense with her game. When she had played it several times with success, she chose to hot glue the electronic fabric to the top and thumb section of the inside of her controller. To play, a user would have to "flap" the controller to make the two hotspots touch each other.

Figure 4: Sara's Scratch Screen (top) and wearable controller design and use (bottom).

4.2 Computing Approaches

Overall students engaged with various computational concepts and practices [3] in designing the games and interactions for their touchpads, game boards, and wearable controllers in each of the workshops. Making games are productive contexts for getting students to engage with a range of computational concepts because they involve parallel actions, sequential activity and event handling that is directed by conditional statements. For example, to win or lose a game, you have to have conditional statements that suggest when that condition is met (e.g., when Score > X). Similarly, games require parallelism. For example, in a game like *Flappy Birds*, the background has to continue to scroll, while the main character, the bird, stays afloat via user input. Event handling and sequence are also integral to game design activities, particularly in our workshops. For example, in Scratch, designers most often start a program by clicking the green flag, which is an event that begins all other activities. Often, students will use the initial event to catalyze another sequence that indicates important information to the user like the name of the game and what keys to press to start the game, then, the game actually begins. Finally, using data is key in games because of the importance of keeping score. For students to have a way to win or lose the game, score and similar concepts (e.g., health of a character) are values that they have to increment and maintain throughout the game and use in their conditional statements. Thus across workshops, students worked hard to have these events mirrored in their own projects.

Furthermore, we observed students engaged in various computational practices through the construction of their designs, on and off the screen. In particular, *reusing and remixing* were popular practices simply because students customized given code. Related computational practices that students also engaged in were *being incremental and iterative* and *testing and debugging*. For example, in the Augmented Board Games class, youth had to take the original code and remix it by changing the aesthetics, updating the text and functionality. Essentially, the process of building a sample board, then a playtest board and a final board, was a result of playing and debugging, and therefore youth could see what did and did not work. Similarly in the Flappy Birds class, students remixed both their Scratch games (drawing on both the concept of the game as well as code from existing versions). In addition, students remixed their wearable controller designs, building on (in some cases) the original prototype we provided and changing the functionality or aesthetics. Then, they iteratively engaged in testing and debugging by temporarily affixing their electronic fabric to their controllers to see if the code and controller worked together. They also had more opportunities to see their designs in action and make on-the-ground tweaks during the arcade. We saw this "in the moment" iteration most significantly in our touchpad controllers class. Students in each class also refined their projects through iterative cycles of imagining their designs, designing and constructing in small steps, trying out and then further developing their designs.

Finally, we also examined how students expressed computational perspectives like relevancy of computing and their perceptions of computing. Irene, who participated in the augmented board games workshop (and later Flappy Birds) expressed how being in the workshop had implications for how she might approach other assignments: "I've heard of Scratch, I've done it a few times, but I didn't … I wasn't that into technology, I mean, it's not … so now I know how to — for projects if I want to make a game or maybe some kind of presentation I can make it on Scratch." Here we see her expressing her initial lack of comfort with technology but also seeing how Scratch might be useful more broadly in her academic work. Students in the touchpad class were particularly shaped by seeing their games in context at the arcade. Students felt an increased sense of confidence when they observed younger students having fun with their games. For example, Earl explains that he originally thought his game was boring but then "when the fifth graders just played it a lot... they played it, said it was fun and that made me think that okay, it's... it's good, fun." Students also gained some valuable insights about design and usability from seeing others playing their game and having to make real-time adjustments. Another participant, James, eventually had to switch back to having users play with the regular keyboard because his interface was only intermittently working, causing him to reflect on how he could have improved his design. Throughout their feedback, students explicated that watching others play their games provided insights and gave them ideas that they hadn't otherwise considered.

5. DISCUSSION

In this paper we investigated the potential of expanding constructionist gaming beyond the screen with hybrid crafting activities. Like in previous constructionist gaming projects, we saw a great deal of personal and creative expression [18] that the novice designers brought to bear not only on the screen but also in their tangible game designs. While these findings are based on a set of small studies, the overall results illustrate the promise of using hybrid crafting activities to expand constructionist gaming beyond the screen. In the following sections, we discuss the affordances for learning computational concepts, practices, and perspectives and the possibilities for including collaboration in computing activities.

5.1 Considering Learning Affordances

Middle school student designers were able to engage with crafting and coding by connecting craft with technology. By going through this process students learned about basic computational concepts in Scratch in addition to learning about conductivity of materials and circuit design—aspects which we did not investigate in these studies. To make hybrid crafting feasible within the constraints of a classroom context and in working with novice programmers, we heavily leveraged the computational practice of remixing by giving students essential pieces of code and even sample controllers designs. This approach offered novice designers a launch pad to develop workable games and controllers within the time constraints of a school setting. Most importantly, the large majority of students went beyond surface changes in remixing code and designs thus supporting it as a valid approach for beginning programmers. We certainly agree that more research and implementation studies are needed to fine-tune the introduction of these crafting and coding activities.

In reviewing the three workshops, it also became clear that the projects varied in complexity, partially because they differed in how much time students had to design and make their game controllers and augmented board games. There is a whole other skill set to be learned in using crafts and materials in additional to coding concepts and practices. Much of this learning links to current efforts to bring maker activities into schools that promote these types of hands-on designs with digital elements [9]. What was particular about the constructionist gaming activities beyond the screen is that they valued crafting and computing equally, meaning in designing game controllers and augmented board games, the designs on the screen and off the screen are equally important. This is distinct from many other hybrid crafting activities such as electronic textiles [2] in which on screen activities are usually limited to writing code which then get downloaded to control the behaviors of actuators and interactions with sensors on the wearable artifact. In our projects, the control was both on and off the screen in the touchpads or game boards, privileging neither modality.

5.2 Expanding into Collaboration

While students were organized in teams only in one class, this arrangement suggests opportunities to deal with the complexity of hybrid designs where students need to learn not only about programming but also about crafting and circuitry. One direction that we can further explore is how we can design and structure the hybrid designs in such a fashion that they engage students with particular computational concepts and practices. In a previous Scratch project that engaged students in designing collaborative music video mash-up on the screen [6] we structured that task in such a ways that initializing and broadcasting (two computational concepts most novice programmers often don't use on their own in Scratch) became a prerequisite for joining seamlessly together the different music sections designed by individual team members. These design constraints proved to be successful in getting all team members to include these concepts in their program code. In the hybrid design space, we could focus on particular controller interactions or features to introduce students to computational concepts we deem important.

Moving game design beyond the screen connected making and playing games, Students, alone or together, not just designed games on and off the screen but they also played them among themselves and in final arcades shared their designs with other students. This gets student designers to think about their games not just as a school assignment for the teacher but actually as a usable artifact that can be shared with others. It also helps to position designers as players and situates within the broader gaming communities. While many of the above reasons focus on the intellectual benefits of constructionist gaming, on or off the screen, ultimately we want to think about computation also as a form of participation that connects playing and making and nurtures learning and literacy [13].

6. ACKNOWLEDGMENTS

The research reported in this paper was supported by a collaborative grant from the National Science Foundation to Yasmin Kafai (NSF-CDI-1027736). The views expressed are those of the authors and do not necessarily represent the views of the Foundation or the University of Pennsylvania. We also would like to thank Peter Endriss for his support and Lei Radium Yang for her help with creating prototypes and Ziyou Annie Meng for helping with data collection and analyses.

7. REFERENCES

[1] Blikstein, P. (2013). The Gears of our childhood: constructionist toolkits, robotics, and physical computing, past and future. In *Proceedings of the 12th International International Conference on Interaction Design and Children* (pp. 173-182). New York, NY: ACM.

[2] Buechley, L., Peppler, K. A., Eisenberg, M. & Kafai, Y. B. (2013) (Eds.). *Textile Messages: Dispatches from the Word of Electronic Textiles and Education*. New York: Peter Lang Publishers.

[3] Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Paper presented at the annual meeting of the American Educational Research Association, Vancouver, BC, Canada.

[4] Campe, S. & Denner, J. (2015). Programming Games for Learning: A Research Synthesis. Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL.

[5] Eisenberg, M. Elumeze, N., MacFerrin, M. & Buechley, L. (2009). Children's Programming Reconsidered. Settings, Stuff and Surfaces. In Proceedings of the 8th International Conference on Interaction Design and Children (pp. 1-8). New York, NY: ACM.

[6] Fields, D. A., Vasudevan, V. & Kafai, Y. B. (2015). The Programmers' Collective: Fostering Participatory Culture by Making Music Videos in a High School Scratch coding Workshop. *Interactive Learning environments, 23*(5), 613-633.

[7] Golstejn, C., van den Hoven, E., Frohlich, D., & Sellen, A. (2014). Hybrid crafting: towards an integrated practice of crafting with physical and digital components. *Personal and Ubiquitous Computing, 18*(4), 593-611.

[8] Hayes, E. R., & Games, I. A. (2008). Making computer games and design thinking: A review of current software and strategies. *Games and Culture, 3*(4), 309-322.

[9] Honey, M. & Canter, D. (2013) (Eds.). *Design Make Play*. New York, NY: Routledge.

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

[11] Kafai, Y. B. (1995). *Minds in Play: Computer Game Design as a Context for Children's Learning*. Mahwah, NJ: Lawrence Erlbaum.

[12] Kafai, Y. B. & Burke, Q. (2014). *Connected Code: Why Children Need to Learn Programming*. Cambridge, MA: The MIT Press.

[13] Kafai, Y.B. & Burke, Q. (forthcoming). *Connected Gaming: What Making Video Games Can Teach Us About Learning and Literacy*. Cambridge, MA: The MIT Press.

[14] Kafai, Y. B. & Peppler, K. (2011). Youth, Technology, and DIY: Developing Participatory Competencies in Creative Media Production. *Review of Research in Education, 35*, 89-119.

[15] Kelleher, R. & Pausch K. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, *37*(2), 83-137.

[16] Lee, E., Kafai, Y.B. Davis, R.L. & Vasudevan, V. (2014). Playing in the arcade: Designing tangible interfaces with MaKey MaKey for Scratch games. In A. Nijholt (Ed.), *Playful User Interfaces: Interfaces that invite social and physical interaction* (pp. 277 - 292). Singapore: Springer.

[17] Millner, A. (2010). Computer as Chalk. Unpublished Doctoral Dissertation. Cambridge, MA: Massachusetts Institute of Technology.

[18] Papert, S. (1980). *Mindstorms*. New York, NY: Basic Books.

[19] Resnick, M., Maloney, J., Hernández, A. M., Rusk, N., Eastmond, E., Brennan, B., Millner, A.D., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. B. (2009). Scratch: Programming for Everyone. *Communications of the ACM,* 52(11), 60-67.

[20] Richard, G. (2008). Employing physical computing in education: How teachers and students utilized physical computing to develop embodied and tangible learning objects. *The International Journal of Technology, Knowledge and Society, 4*(3), 93-102.

[21] Silver, J., Rosenbaum, E., & Shaw, D. (2012). MaKey MaKey Improvising Tangible and Nature-Based User Interfaces. *In Proceedings of the ACM Tangible Embedded and Embodied Interaction* (pp. 367-370). Kingston, Ontario, Canada: ACM.

[22] Vasudevan, V. & Kafai, Y. B., & Yang, L. (2015). Make, Wear, Play: Remix Designs of Wearable Controllers for Scratch Games by Middle School Youth. In Proceedings of the 15th International Conference on Interaction Design and Children (pp. 339-342). New York, NY: ACM.