



University of Pennsylvania
ScholarlyCommons

GSE Publications

Graduate School of Education

2010

Gaming Fluencies: Pathways into Participatory Culture in a Community Design Studio

Kylie A. Peppler

Indiana University - Bloomington

Yasmin B. Kafal

University of Pennsylvania, kafal@gse.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/gse_pubs

 Part of the [Education Commons](#)

Recommended Citation

Peppler, K. A., & Kafal, Y. B. (2010). Gaming Fluencies: Pathways into Participatory Culture in a Community Design Studio. Retrieved from http://repository.upenn.edu/gse_pubs/222

Suggested Citation:

Peppler, K. and Kafal, Y. (2010). Gaming Fluencies: Pathways into Participatory Culture in a Community Design Studio. *International Journal of Learning and Media*. Vol. 1(4). pp. 45-58.

© 2010 MIT Press

<http://www.mitpressjournals.org/loi/ijlm>

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/gse_pubs/222

For more information, please contact repository@pobox.upenn.edu.

Gaming Fluencies: Pathways into Participatory Culture in a Community Design Studio

Abstract

Many recent efforts to promote new literacies involve the promotion of creative media production as a way to foster youth's literate engagement with digital media. Those interested in gaming literacies view game design as a way to engage youth in reflective and critical reading of the gaming culture. In this paper, we propose the concept of "gaming fluencies" to promote game design as a context in which youth not only learn to read but also to produce digital media in creative ways. Gaming fluencies also present the added benefit of addressing equity issues of participation in the new media literacy landscape. We report on an ethnographic study that documented urban youth producing digital games in a community technology center. Our analyses focus on an archive of 643 game designs collected over a 24-month period, selecting a random sample to identify evidence of creative and technical dimensions in game designs. In addition, we highlight three case studies of game designs to identify different pathways into the participatory culture. Our goal is to illustrate how gaming fluencies allow for a wide range of designs, provide low thresholds and high ceilings for complex projects, and make room for creative expression. In our discussion, we address how gaming fluencies represent a complementary pathway for learning and participation in today's media culture.

Disciplines

Education

Comments

Suggested Citation:

Pepler, K. and Kafal, Y. (2010). Gaming Fluencies: Pathways into Participatory Culture in a Community Design Studio. *International Journal of Learning and Media*. Vol. 1(4). pp. 45-58.

© 2010 MIT Press

<http://www.mitpressjournals.org/loi/ijlm>

Gaming Fluencies: Pathways into Participatory Culture in a Community Design Studio

IJLM

Kylie A. Peppler
Indiana University, Bloomington
kpeppler@indiana.edu

Yasmin B. Kafai
University of Pennsylvania
kafai@gse.upenn.edu

Keywords

game design
informal learning
Scratch
media education
new literacies
technology fluency

Abstract

Many recent efforts to promote new literacies involve the promotion of creative media production as a way to foster youth's literate engagement with digital media. Those interested in gaming literacies view game design as a way to engage youth in reflective and critical reading of the gaming culture. In this paper, we propose the concept of "gaming fluencies" to promote game design as a context in which youth not only learn to read but also to produce digital media in creative ways. Gaming fluencies also present the added benefit of addressing equity issues of participation in the new media literacy landscape. We report on an ethnographic study that documented urban youth producing digital games in a community technology center. Our analyses focus on an archive of 643 game designs collected over a 24-month period, selecting a random sample to identify evidence of creative and technical dimensions in game designs. In addition, we highlight three case studies of game designs to identify different pathways into the participatory culture. Our goal is to illustrate how gaming fluencies allow for a wide range of designs, provide low thresholds and high ceilings for complex projects, and make room for creative expression. In our discussion, we address how gaming fluencies represent a complementary pathway for learning and participation in today's media culture.

 Visit IJLM.net

doi:10.1162/ijlm_a_00032

© 2010 Massachusetts Institute of Technology
Published under [Creative Commons Attribution-Noncommercial-No
Derivative Works 3.0 Unported license](#)

Volume 1, Number 4

Introduction

The recent discussions about games and learning are part of a larger debate about the new literacies (New London Group 1996) needed in the 21st-century participatory culture. Jenkins and colleagues (2006) have argued that youth need to understand and produce content across different media platforms, to articulate their understanding of how media shape perception, and to become knowledgeable of emerging ethical standards that shape their practices as media makers and participants in online communities. Because digital game play provides essential “entry points for many young people into digital literacy, social communities, and tech-savvy identities” (Salen 2007, p. 302), some have proposed that digital games be considered a “cultural medium” in their own right and have referred to the ability to critically read and produce game texts as “game literacy” (Buckingham and Burn 2007) or “gaming literacies” (Salen 2007).

In this article, we focus on creative media production (Peppler and Kafai 2007) in gaming literacies, an area that has received relatively little attention in the past but is now a growing area of research and design in education. We define gaming fluencies to include not only the critical evaluation of game designs but the creative production and technological implementation of those designs. Gaming fluencies include the abilities to design, modify, and play games, to reflect critically on games, to engage in artistic production, and to bring to bear the technology fluencies needed to produce and play games.

Our efforts draw on early work in this area (Kafai 1995, 2006a), as well as on more recent studies on using game design to engage girls in new technologies (Denner and Campe 2008; Flanagan and Nissenbaum 2008), in articulating their thinking about gender (Pelletier 2008), and in thinking about design (Salen 2007; Hayes and Games 2008). Our approach differs from these other efforts in that we use game production not just as a way to promote gaming literacy in the broadest sense but to enhance the technological fluency that disadvantaged youth particularly need (Kafai, Peppler, and Chiu 2007). We have chosen to make “gaming fluencies” the focal point of our approach because we consider game design a promising pathway into the technology culture at large. We align our proposal with work in the constructionist tradition (Kafai 2006b) that aims at developing pedagogies to promote technology fluency rather than

computer literacy, thus moving beyond technocentric perspectives.

To illustrate our view on gaming fluencies, we examine the artifacts of game production that took place at the Computer Clubhouse, a community design studio and technology center in south Los Angeles and a part of the Intel Computer Clubhouse Network (<http://www.computerclubhouse.org>; Kafai, Peppler, and Chapman 2009). Youth in the Clubhouse had access to a wide range of design tools, but our attention focused particularly on their use of Scratch (Resnick et al. 2009), a media-rich programming environment that we introduced in 2005 as part of a larger research project (Kafai, Peppler, and Chiu 2007). During the following 24 months, we archived all Scratch projects and documented daily design work in field notes. Our analyses focus on a random sample of 40 percent of a game archive containing 643 Scratch projects produced by Clubhouse members. The range of projects contained in the random sample reflects youths’ evolving knowledge of game design, developed and reshaped by the local community. We further examine three games in more detail to illustrate the design choices, interface features, and narrative elements used by youth. These investigations allow us to address the following general research question: What do youth learn in the process of designing games? We are interested in individual pathways into gaming fluencies, the various aspects of gaming fluencies that are expressed in youths’ game designs, and the types of game genres that youth create as well as what types of game components are incorporated into the games. In our discussion, we address how gaming fluencies represent a complementary pathway for learning and participation in today’s media culture.

Background

The focus on game design as an approach to learning is a relatively recent addition to the fields of education, technology, and media literacy studies. The academic community in the 1980s and 1990s concentrated on the psychological impact of game play by discussing gender differences in spatial reasoning (Loftus and Loftus 1983) or perception of violence (Provenzo 1991), with the occasional studies evaluating educational applications of game design (for a more extensive review, see Egenfeldt-Nielsen 2007). Designers at game companies, but not players at home, were seen as the producers of

content and mechanics. This might explain why media literacy was understood in terms of players being able to critically evaluate the content and production of games, just as they would focus on commercial or political intent in television advertisements or newspaper articles. The very idea of production in the hands of the lay viewer or game player was resisted on several grounds. Researchers in media studies perceived viewers mainly as consumers of media and did not see critical inquiry reflected in production work (Jenkins 1992), whereas researchers in education questioned the general learning value of games and focused mainly on the motivational benefits of games (Malone and Lepper 1987). Game companies simply saw the production of game content as their domain.

One of the first studies to address game production for learning did not come out of the traditional strands in these academic communities but built on efforts to construe design as a new pedagogy for learning with technology. When the research for computer game design as a context for learning was conducted in the early 1990s (Kafai 1995), the idea of design for learning had just developed a foothold in the larger education community (Perkins 1986; Collins 1992). Inspired by Simon's (1981) *The Sciences of the Artificial* and Schön's (1983) *The Reflective Practitioner*, professional practice in design disciplines became seen as contexts that promoted open-ended forms of problem solving and situated the learning and application of academic contents in the design of meaningful, authentic applications. Harel's (1991) seminal work was one of the first studies to illustrate how students as designers of instructional mathematics software could engage in long-term, meaningful, and integrated learning of programming and mathematics (Palumbo 1990). Many of the projects sought to use technology as a design rather than a teaching tool, the latter being the prominent focus of most early technology applications (Cuban 1986). All of these approaches saw students as designers of a product—be it an instructional simulation (Kafai and Ching 2001), historical presentation (Erickson and Lehrer 1998), mechanical device (Penner, Lehrer, and Schauble 1998), or engineering design (Hmelo, Holton, and Kolodner 2000)—but differed in how they conceptualized the design process and artifact in the service of learning. Similar differences exist in how game design is currently viewed in education.

These developments formed the background for one of the first empirical studies of game design in

schools (Kafai 1995). A class of 16 elementary school students, ages 10–11 years, was invited to design computer games that would teach younger students in their school about fractions. Over a period of six months, student designers set out to write and implement their own game designs using Logo programming. They also created packaging and advertisements for their games. The designers met about once a month with the intended players, a group of younger students who provided them with feedback on various aspects of the games (see also Kafai 1998). The observed gender differences in the narratives, components, aesthetics, and mechanics of the students' games have received the most attention (Kafai 1996a, 1998)—most likely because these findings aligned well with then-popular discourse about gender differences in interest and performance in technology and games (e.g., Cassell and Jenkins 1998). Much less attention was given to the equally important aspects—such as design practice, collaborative planning, and public critique—that contribute to students' understanding (Kafai 1996b) but resonate more with current approaches to game designs for learning.

Indeed, current proposals for game design draw their inspiration from the fields of media studies and game studies. For instance, Buckingham and Burn (2007) suggested the term “gaming literacy” because they see games as another example of multimodal texts that integrate sound, music, graphics, writing, and more. For them, game design combines cultural experiences that vary by age, gender, cross-media knowledge, and appreciation of particular features and genres. Gaming literacy is the understanding of the narratives, rules, and economies that provide the semiotic structures of games, whereas creative production involves transformation of different resources and assets pertaining to games. The transformation of these aspects was also captured in Salen's notion of “gaming literacies,” which encapsulates multiple professional practices, including expertise “in graphic design (visual design, interface design, information architecture), product design (input and output devices), programming, animation, interactive design (human computer interaction), writing, and audio design, as well as . . . in content areas specific to a game” (Salen 2007, p. 318). Both Buckingham and Burn (2007) and Salen (2007) suggest that game play is not the only approach to becoming literate in gaming; writing reviews or “modding” game components can be equally instrumental.

In our proposal for gaming fluency, we recognize that participation in media culture can take various forms, all valid and valuable contributions to the larger ecology of gaming. But we see game production as not limited to the context of gaming but having the potential to address broader equity issues. In the conceptualization of gaming fluencies, we see thinking and producing like a game designer as a valuable starting point. Whereas Jenkins and colleagues (2006) define the participation gap as unequal access to the opportunities, experiences, skills, and knowledge necessary to prepare youth for full participation in a digital culture, we expand upon this notion and specifically apply it to game production, a field that has been dominated by white males. This dominance has been problematic for many reasons, including the lack of representation of women and minority avatars in games, the reduction of such groups to exaggerated stereotypes, and the overabundance of games marketed toward white males (for a review, see Kafai 2009). By providing opportunities for underrepresented youth to participate in making games, we hope they can be vehicles of change as both critical consumers and designers in an industry that has an increasing importance for schools and society at large. Our study focuses on urban youth, a group often seen as consumers of new media but rarely positioned as designers, especially as designers of games.

The differences between gaming literacies and fluencies become the most apparent in the conception of tools for game design. All such tools provide a design language that will allow the producer or designer to manipulate input/output, sound, and graphic aspects of the game. As Salen argued, "Game designers must know how to speak the 'language' of each of these fields in order to see the possibilities and constraints of their design. The intersection of constraints from each area with the rules of play shapes the game in innumerable ways and drive[s] the design process forward" (Salen 2007, p. 318). Those interested in gaming literacy or literacies have focused on creating game-specific toolkits such as MissionMaker (Buckingham and Burn 2007) or Gamestar Mechanic (Salen 2007), which focus on key elements in game production, such as the system-based thinking needed "to enable a deep understanding of how a system of rules, behaviors, and relationships guide the design of an interactive experience, activated through play" (Salen 2007, p. 314).

Our interest in gaming fluency has involved the creation of a more general tool that is not specific

to game design. Scratch (Resnick et al. 2009), Alice (Kelleher 2008), and Peeps (Flanagan and Nissenbaum 2008) are examples of this type of general tool. Scratch projects allow designers to move fluidly between different forms of media (Resnick and Silverman 2005), of which games are but one, and allow for a variety of artistic choices (Peppler, forthcoming). Some proponents of gaming literacies have claimed that general construction tools such as Scratch put too much emphasis on technical aspects such as writing code to the detriment of reflective design engagement (Hayes and Games 2008). We believe this distinction is arbitrary and neglects to take into account the design process and community in which game design with Scratch is situated.

The type of design we propose for gaming fluencies is based on constructionism, a well-articulated theory of learning and a pedagogical framework that emphasizes the expression of ideas, values, and genres. Constructionism places learners in roles of creating artifacts that are of relevance to a larger community (Papert 1980; Kafai 2006b). Constructionism places equal importance on the individual learner and the role of social participation. Here the individual, the artifact, and collaborative input from the community shape learning, participation, and sharing. In the case of game production today, the community is both the distributed online and offline community. Sociocultural constructionists further argue that the individual and the community develop reciprocally through "shared constructive activity that is resonant with both the social setting that encompasses a community of learners, as well as the cultural identity of the learners themselves" (Pinkett 2000). A design tool like Scratch can promote the reciprocal development of individual and community, enable youth to express their cultural heritage, have a broad communicative value, and allow for information and resource exchange (Pinkett 2000). We have found evidence in previous analyses of game productions (Peppler and Kafai 2007; Kafai, Fields, and Burke, forthcoming) that youth become familiar with the ambiguities of ownership in a digital realm and begin to understand how media shape their perceptions of the world—what Jenkins and colleagues (2006) describe as the ethics and transparency problems, respectively.

In this article, we focus on the creative and technical aspects of game production with Scratch. We examine how choices of genre, narrative elements, and visual design in conjunction with the design of

interface features lead Scratch designers to develop gaming fluencies. As with Gamestar Mechanic, designers using Scratch can begin design activities with little or no prior knowledge and still build complex designs (Peppler and Kafai 2007). We expect students using Scratch to develop a broader range of game designs than they would using specialized game design tool kits because Scratch has what Resnick and Silverman (2005) have called “wide walls,” which allow for a range of design choices. Furthermore, we identify which aspects of game design in Scratch involve the type of systems-based thinking that Salen (2007) valued in Gamestar Mechanic activities. Finally, artistic expression is particularly important as we struggle to be inclusive of individuals with diverse backgrounds. Our central argument is that gaming fluencies, like gaming literacies, promote valuable learning goals in the design of games but do so in addition to developing technology fluency (National Research Council 1999).

Context, Participants, Tools, and Designs

Our ethnographic work at the Computer Clubhouse in south Los Angeles was driven by a desire to better understand youths’ game design practices (Peppler and Kafai 2007). The Clubhouse was located in a storefront in one of the city’s poorest areas and served over 1,000 high-poverty African American and Hispanic youth. Youth worked on projects individually and in small groups and ranged from 8 to 18 years old, but most were 10–14 years old. The game-design efforts of a community of mixed-age and mixed-ability youth, consisting of novice and expert game designers, were supported by a small group of local mentors and, over the course of the study, mentors from a partnering local university. However, the mentors had little or no programming or game design experience (Kafai et al. 2008).

All computers at the Clubhouse were networked to a central server, where youth had a personal folder that served as an image archive and repository for finished and in-progress work. This facilitated long-term projects as well as sharing. The Clubhouse provided youth with an impressive variety of software, including Scratch (the programming environment), Microsoft Office, Bryce 5, Painter 7, and RPG Maker™, in addition to video, photography, and sound editing software. Scratch, in particular, differs from other visual programming environments (Guzdial 2004) in

its use of a familiar building block command structure (Resnick, Kafai, and Maeda 2003; Maloney et al. 2004), thus eliminating thorny debugging processes and the risk of syntax errors that can impede a novice designer from creating games (see figure 1). Programmed objects can be any imported two-dimensional graphic image, hand-drawn or chosen from a personal archive. Two-dimensional game design holds several advantages for youth wanting to make their own video games. For example, two-dimensional programming facilitates easy incorporation of third-party images, thus allowing greater creative expression, and can be easier for novices to learn than three-dimensional programming.

Figure 1 is a full screenshot of the Scratch interface used for game-making. On the left side of the screen is the palette of *programming command blocks*, which allow youth to control and manipulate sound, images, motion, and various types of input from the players. In the lower right side of the screen is a library of *sprites*, which can be any imported or hand-drawn character or object in the game. Above the library of sprites is the *stage* displaying the games that are in the process of being created or edited. The middle panel contains three tabs with information about the selected sprite. In this screenshot, commands that the creator has stacked together to control a particular sprite are displayed in the center panel. Clicking on the other two tabs at the top of the panel would reveal information about the sprite’s costumes (see figure 4) or sounds. The game can also be converted to play mode with the touch of a button. In this mode, the game can no longer be edited or changed by the player.

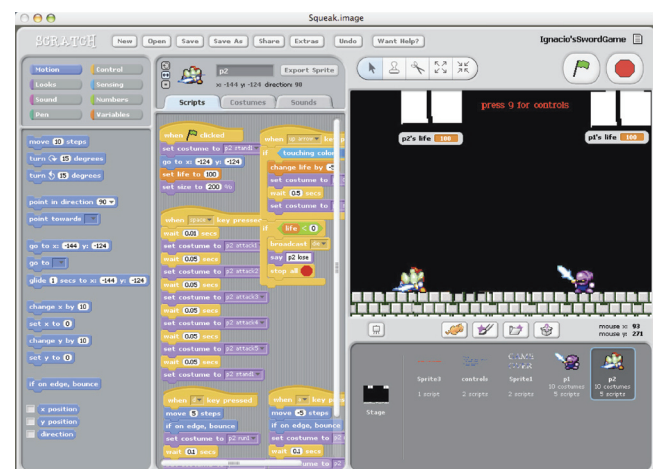


Figure 1 Screenshot of the Scratch user interface with sample game “Ignacio’s SwordGame.”

The Scratch vocabulary of roughly 90 commands includes commands for motion, image transformation (rotation, scaling, and effects such as fish-eye), stop-motion animation (switching between images), recorded-sound playback, musical note and drum sounds, and a programmable pen. From a programming standpoint, Scratch has a number of control structures, including conditionals (*if, if-else*), loops (*repeat, forever, repeat-until*), and event triggers (*when-clicked, when-key-pressed*). Communication is via named broadcasts. For example, one sprite might broadcast “you won!” causing another sprite to appear on the stage and play a victory song. One broadcast can trigger multiple scripts. A variant of the broadcast command waits for all triggered scripts to complete before going on, thus providing a simple form of synchronization. In addition, Scratch supports two kinds of variables. *Sprite variables* are visible only to the scripts within that sprite, while *global variables* are visible to all objects. Global variables are sometimes used in conjunction with broadcasting as a way to pass data between sprites.

Youths’ game designs using Scratch were collected on a weekly basis and entered into an archive for further analyses ($n = 643$). We analyzed a randomly selected sample of 260 projects, or close to 40 percent of the total archival data. A top-down approach to coding (Chi 1997) was derived from the literature on gaming literacies to code the archive. Three broad classes of top-down codes were used: The first was used to code and categorize the genres of game designs into several subcategories (e.g., interactive narratives, fighting games, mazes); the second was used to capture the creative aspects of game design (e.g., type of animation and whether aesthetic elements like perspective, motion paths, and audio design were included); the third was used to capture the technical aspects of game design, particularly the types of programming commands and interaction design used in the projects (e.g., loops, conditional statements, human-to-human interaction design). Our goal was to capture the range of gaming fluencies that emerged over the course of the study.

In addition, we collected detailed profiles of participants, using interviews, videotaped observations, extensive ethnographic field notes (Creswell 2003), and other artifacts. As part of our observations of the emerging game-design culture at the Clubhouse, we focused on three case studies of how individuals of different backgrounds and abilities navigated this

landscape. The case studies were selected from over 30 participants based on the prototypical nature of their work and their persistent interest (over a period of multiple weeks) in using computer programming for game design.

Game Productions in the Design Studio

An analysis of all files ($n = 11,926$) saved on the Clubhouse central server during the first 24 months after Scratch was introduced reveals that the most popular activity was saving images found on the Web (>75 percent of files saved, $n = 8,944$). The second most popular activity, and the most popular design software used during the period, was Scratch (5.3 percent of files saved, $n = 643$). Other popular design software during the period included Microsoft Word (3.5 percent of files saved, $n = 461$); Bryce 5, a 3D modeling, animation, and rendering program (2 percent of files saved, $n = 270$); and Kai’s SuperGoo, an image distortion program (1.1 percent of files saved, $n = 143$). Notably further down on the list of popular design software was RPG Maker (0.4 percent of files saved, $n = 43$), a software package used to design role-playing games that was available at the Clubhouse for a number of years before the introduction of Scratch. The number of files created by the various design programs available at the Clubhouse illustrate that digital design production was popular among youth at the Clubhouse. We think Scratch’s popularity was due in large part to its ease of use; its flexibility allowed for a greater variety of game genres, components, and aesthetics. In addition, Scratch was quick to be adopted in the Clubhouse because of rules and norms that supported a design-based approach to learning as well as the presence of mentors and knowledgeable peers.

We know that a number of projects were created within the design culture of the Clubhouse, but what of the creators themselves? Over the course of the study, the Clubhouse served more than 1,000 youth, 252 of whom created an individual folder on the central server, indicating that they had been introduced to the design studio by a peer, mentor, or coordinator. Of the roughly one-quarter of Clubhouse attendees who were initiated into the design studio, more than 80, or roughly 8 percent of all Clubhouse youth, saved a Scratch project to the server. Overall, 40 percent of the Scratch archive was created by male members, 29 percent by female members, 9 percent

by small groups, and 22 percent by designers whose gender is not known.

In the second year of the project, we observed new aspects of a video game-making culture emerge. For example, video game production became a marker of high status, and two types of experts emerged: some youth developed expertise in highly focused (local) aspects of game design, and others focused more on general (global) principles of game design. Work in Scratch established one's membership within the Clubhouse community (Kafai, Peppler, and Chiu 2007). In addition, we began to see peer-to-peer mentoring in video game design and increased appropriation of Scratch as a video game design tool. As a result, new game genres emerged, and individuals worked in groups with increased frequency, with some youth specializing in specific aspects of game design such as graphic design, animation, and programming—a level of focus more commonly seen in professional video game-design settings.

Game Designs: Three Case Studies

The archive of 643 Scratch projects is the source material for three case studies of game designs illustrating how youth engage in video game practices as they learn visual programming, narration, animation, logic, sound design, aesthetics, and interaction design—the basis for what we consider gaming fluencies. The three cases we selected represent different approaches to game genres and also varied in complexity of programming and visual design.

Jerrell: "Jerrell (Hoops)"

"Jerrell (hoops)" is a sports game created by a 13-year-old African-American game designer named Jerrell.¹ "Jerrell (hoops)" was the designer's first project in Scratch and featured several game components, including a clip-art basketball hoop, basketball, directions, a timer, and a scoreboard. This one-player game had no particular narrative, which is probably typical of most commercial sports games because the sport theme itself provides the context. The game also lacks any audio, such as voiceover, soundtrack, or other types of sound. When the player presses the start button (a green flag in the upper corner of the Scratch interface), directions saying "Freethrow in 60 seconds" pop on screen and spin. The object of the game is to score as many points as possible in 60 seconds using the arrow keys to direct the basketball into the net.

If the player makes a basket, the scoreboard tallies a point and the basketball moves to a random part of the screen. Once more than 50 seconds has gone by, a "Hurry Up" sign appears in the upper-right portion of the screen (see figure 2). After 60 seconds, a speech balloon appears from the basketball and says, "Game Over," at which point everything is brought to a stop and the scoreboard ceases to register additional points.

Despite the lack of a win/lose condition explicitly built into the game, youth at the Clubhouse were quick to try to beat their or their peers' personal best scores, creating a de facto win/lose experience. The game itself was coded as having two dimensions: the ball and other objects appear to be in a single plane with no background receding into the distance. Although Jerrell's game design used programmed animation effects (such as move commands) only to make the ball move, overall he used a wide array of programming concepts, including random numbers, loops, conditional statements, variables (such as those needed for the scoreboard), player interaction (i.e., use of keyboard inputs), and communication/synchronization (i.e., use of broadcast and *when-receive* blocks).

Chandelle: "Chandelle Castle"

The designer of "Chandelle castle," Chandelle, was an 11-year-old girl of mixed African-American/Latina heritage. She attended a public school near the Clubhouse location and was in grade 7 at the time of the study. At the time she created her game, she had been sporadically attending the Clubhouse for a year

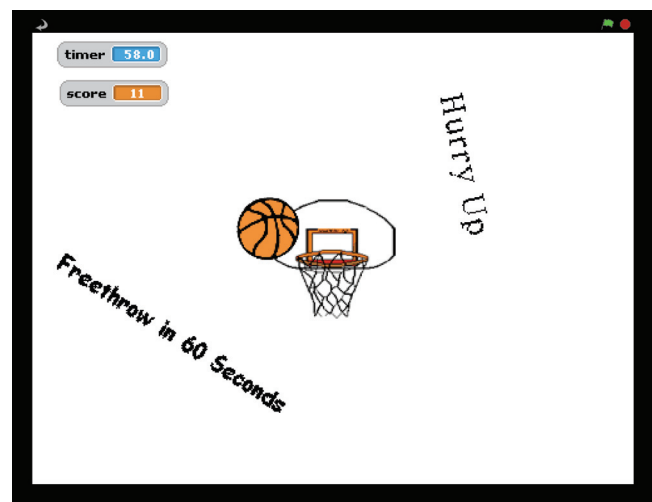


Figure 2 Screenshot of "Jerrell (hoops)" in play mode.

or two and had occasionally been involved in some aspects of creative production. “Chandelle castle,” a one-person game, was Chandelle’s second Scratch project. The game, categorized as an interactive narrative in the genre analyses, features a narrative (or plotline) that is both prominent and familiar: A princess is locked in a castle tower by an evil witch and needs to be rescued. The game features a methodically designed background consisting of a distant castle in the middle of the screen between two contrasting panels of color. Chandelle created a bicolor background as a metaphor for the evil witch, who portrays herself as kind (represented by the yellow half of the sky and the bright colors of the castle) but is actually full of evil intentions (represented by the black half of the sky). Chandelle was one of the few Clubhouse youth to make use of mouse movements in her player interface design: When the player clicks on either half of the background, the evil witch appears in the doorway of the castle and a speech balloon appears accompanied by a crackly voiceover sound (recorded by Chandelle) of the witch wickedly laughing. At the same time, a series of hand-drawn “HELP ME!” texts zoom toward the player from the top of the castle as the princess repeatedly cries, “Help me!” The player can see someone in the distance suspended by a parachute, but nothing happens until the player makes the next move to locate the princess (see figure 3, left panel). The player must find the princess by clicking around on the screen. Once the player clicks on the princess, the knight parachutes in and lands on the bridge, announces to the witch, “stop right there,” and rescues the princess (see figure 3, right panel).

Chandelle’s game design did not include written directions, so the rules had to be orally communicated to other players at the Clubhouse. The game design also omitted such game components as scoreboards and timers. And although the game had a win/lose component (i.e., players had to find the princess to win), this component was only accidentally discovered



Figure 3 Two screenshots of “Chandelle castle” in play mode at various points in the narrative.

by Chandelle while she was demonstrating the game to a group of Clubhouse boys. When she had trouble finding the princess herself, the boys perked up and volunteered to give it a try. The “Chandelle castle” project used several types of animation, including limited stop-motion animation (because of its limited number of frames), programmed visual effects (such as the “help me” texts getting larger), and programmed animation (movement). Chandelle’s game was also one of the few to use three-dimensional perspective in the background and to have characters appear to move in three dimensions. Chandelle’s project used a mix of programming concepts, including user interaction (use of mouse input), communication/synchronization (broadcast and *when-receive* commands), loops (repeat stack), and conditional statements.

Jorge: “Mortal Kombat”

The Scratch game “Mortal Kombat” was created by a 15-year-old Latino male software designer named Jorge, who modeled the piece after a similarly titled popular video game, *Mortal Kombat: Deadly Alliance*. The original Mortal Kombat (MK) is a series of commercial fighting games that first appeared in arcades and quickly became popular for at-home consoles. The game is well known for its use of digitized sprites (as opposed to the hand-drawn sprites of other contemporary games), for specialized moves that require a complicated sequence of buttons, and, perhaps most notorious, for an inordinate amount of graphic violence.

Jorge’s “Mortal Kombat” was also a fighting game and had many similarities to the original. At the start of Jorge’s game, the player can click on the background to start the Mortal Kombat theme music and then can use a series of keys on the keyboard to control a fight between two animated sprites, recognizable by MK fans as Scorpion and Sub-Zero (see figure 4). Jorge’s game was coded as a two-player game in which each player controls his or her own avatar. As the players fight, they are taken through a series of 16 backgrounds, which were intended eventually to act as levels. The backgrounds used three-dimensional images found on the Web and downloaded for use in Jorge’s project. The background created an illusion of three-dimensional play space, but the characters moved only in two dimensions (up/down, right/left).

After finding that players had difficulty remembering which keys were used to control the sprites,

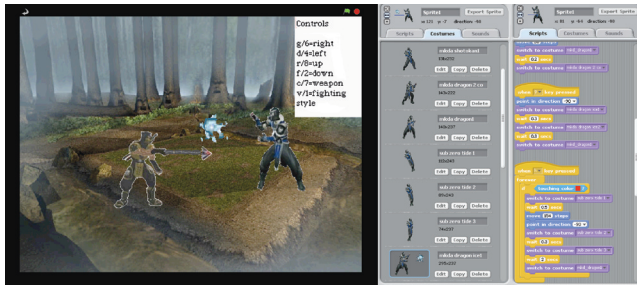


Figure 4 Partial screenshots of Jorge's "Mortal Kombat" game.

Jorge posted directions on the first level of the game in the upper right-hand corner of the play screen. Based on these experiences, Jorge deduced why many similar games use conventional keystrokes (e.g., arrow keys) to control on-screen sprites: they are easier to remember and do not require overlapping areas of the keyboard. This insight indicated that he was learning something about human-to-computer interaction design. In order to build this type of interactivity, Jorge repeatedly used several different programming concepts in his project, including sequential execution (a stack of commands with more than one block), threads (multiple scripts running in parallel), conditional statements (if-then statements), and user interaction (use of keyboard or mouse input).

Each of Jorge's sprites was programmed with a series of 12–15 costumes that were used for stop-motion animation and accompanying voiceover sounds. Each of the still images was based on a file downloaded from the Web and imported into Scratch but altered by Jorge's hand manipulation of the original image to create a fluid animation effect. His figures looked as if they were throwing weapons at one another, walking, and spinning. In addition to coding stop-motion animation effects (such as moving up, down, right, and left). No other visual effects were programmed, and the game lacked a narrative beyond that of two characters fighting one another. Jorge was ultimately unable to finish "Mortal Combat" because of a switch between alpha and beta versions of Scratch at the Clubhouse. After the new version of Scratch was installed, Jorge began to program his own version of the Metal Slug video game series (Peppler and Kafai 2007).

Development of a Game Design Community: Archival Analyses

These three case studies illustrate the diversity of game designs created in the Clubhouse community.

But do these game designs represent the full range of gaming fluencies? The archive of Scratch game projects stored on the Clubhouse central server allowed us to examine the various types and extent of animation used, the types of artistic and audio design components used, as well the extent of high-level programming concepts and interaction design (both human-to-computer and human-to-human) found in the projects.

Game Genres

As a starting point, we coded the youths' projects by game genre (see table 1). We found that 50.8 percent were interactive narrative projects (e.g., animations, interactive art, and narrative games); 7.3 percent were sports games; 2.3 percent were simulation games; and 8.8 percent were distributed among other game categories, such as mazes, rhythmic games, role-playing games, interactive shooter games, racing games, or platform games. Some projects defied categorization; for example, 22.3 percent were graphics-only files with no associated game mechanics, and 8.5 percent were empty projects with titles only. Neither of these categories was coded for genre. The finding of numerous game genres among

Table 1. Table of Game Genre Types Found in a Random Subsample of the Archive

Game Genre		Number of Games within Archival Sample	% of Total
Interactive narratives/animations		132	50.8%
Standard game genres	Sports games	19	7.30%
	Simulation games	6	2.30%
	Mazes	4	1.50%
	Rhythmic games	4	1.50%
	Role-playing games	4	1.50%
	Interactive shooter games	4	1.50%
	Driving/racing games	3	1.20%
	Fighting games	1	0.40%
	Platform games	1	0.40%
Other genre type		2	0.80%
No game mechanics/empty projects		80	30.8%
TOTAL SAMPLE		260	100%

the youths' projects was augmented by observations from our field notes that Clubhouse youths would often cluster together to share ideas and provide support for one another's designs within certain genres, indicating the existence of numerous subcultures of game-makers.

The predominance of interactive narrative projects among youth at the Clubhouse indicates their strong interest in creating projects within this genre. Interactive narratives included short animations, choose-your-own-adventure stories, and other types of art projects with interactive components. Newer forms of games also make use of these types of narrative features (not all games are traditional fighting or first-person shooter games). The narratives generally took one of three forms: they might contain voiceovers that had narrative elements; they might contain thought balloons or speech bubbles that advanced a story line; or, conversely, they might include visual elements that told a story without the use of words. Across all of the game genres, storytelling was present in 47 percent of the games ($n = 122$), most of which were found in the interactive narratives or animations.

We also found that 20 games contained scoreboards, seven contained timers, and nearly all (240 out of 260) contained sprites or central game characters. Many games contained multiple sprites (the average was 5.76 per game).

Creative Aspects

When considering the creative aspects of the games, we examined three broad classes of design elements: types and extent of animation, use of perspective, and design of sound. Sprites were animated in various ways, the most common being programmed animation, where the creator uses movement commands to animate static (or sometimes stop-motion animated) sprites. In the archive, we found that 181 games contained programmed animation effects such as a figure gliding across the screen. A second type of animation, one used in 117 of the games, involved programmed visual effects, ranging from a changing color effect to an increase/decrease in size. The last type of animation, stop-motion, was used in 69 of the games. Stop-motion animation, which was used in the first filmed animation sequences and thus predates digital visual effects design by many years, is created from a series of hand-drawn or downloaded frames that are independent images but when changed rapidly in close

succession create an illusion of movement. The use of animation and other sophisticated gaming elements in youths' work at the Clubhouse demonstrates a Clubhouse culture that was knowledgeable in some aspects of animation and game design.

We also examined Clubhouse members' use of aesthetic elements found in many commercial games. In other work we have explored the artistic content of the youths' projects (Peppler, forthcoming) and found that youth at the Clubhouse were engaged in learning about the professional practices of media artists and that several of the youths' works resembled those of contemporary professionals in the field. For the present study, we focused on the type of perspective (two-dimensional and three-dimensional) that the designers used in their games, as well as the extent of audio design used. We found that 72 games used backgrounds or motions that were two-dimensional and that an additional 173 games used three-dimensional graphics or motion paths. The remaining games did not include any two- or three-dimensional graphics or motion paths. Additionally, 89 games contained sound. Sounds included original compositions, voiceovers, imported music loops, or original recordings. These findings indicate that the Clubhouse members were widely incorporating aspects of aesthetic and audio design, enhancing the professional value and personal quality of their work.

Technical Aspects

When examining the technical aspects of gaming fluencies, we considered both the programming and type of interaction design used in the project. Programming of the games became more complex during the second year of observation, as indicated by the increasing number of stacks, sprites, sounds, and costumes, as well as by the types of programming concepts contained in the projects (Maloney et al. 2008). These concepts included user interaction, loops, conditional statements, communications and synchronization, Boolean logic, variables, and random numbers—difficult concepts even for novice computer science majors at the college level (Spohrer and Soloway 1985). The increased breadth and frequency of challenging programming concepts in the second year of the study indicates that the Clubhouse community had become more adept at programming with experience—another central component of creating a sustainable video game design community.

In addition, we took a closer look at the types and extent of interactivity that occurred in the projects.

We examined two types of interaction design: human-to-computer interaction and human-to-human interaction. (Human-to-computer interaction design involves the types of game devices used to interact with the game.) Of the 260 games that we examined, youth designed 160 containing simple interactions with the game through a start button (or green flag). Of the 101 games that allowed for player interaction through keys pressed on the computer keyboard, 15 projects allowed the player to interact with the game by clicking on-screen objects, and eight games allowed the player to interact with or control aspects of the game through mouse movements along an x- and y-axis or through mouse clicks (see table 2). Games could also involve one or more of these types of human-to-computer interactions, but we did not take the overlap into account in our analyses. The use of relatively simple human-to-computer interactivity in the majority of projects that included this capability suggests that more complex forms of human-to-computer interaction need to be an explicit part of youths' exploration of Scratch.

We also examined the human-to-human interaction design present in the games (see table 3). Although 80 of the games had no form of interaction, 146 were one-player games or other types of projects.

Table 2. Types of Human-to-Computer Interaction Design Found in the Game Archive

Type of Game Device Interaction Used	Percentage of Sample*	Number within Archival Sample*
Start button clicked	61.5%	160 games
Key pressed	38.8%	101 games
Object clicked	5.8%	15 games
Mouse movements	3.1%	8 games

* Some projects were coded as having two or more types of interaction, so numbers do not add to 100 percent.

Table 3. Types of Human-to-Human Interaction Design Found in the Game Archive

Number of Players	Percentage of Sample	Number within Archival Sample
None	30.7%	80 games
One-player games	56.2%	146 games
Two-player games	11.1%	29 games
Three-player games	1.2%	3 games
Four-player games	0.8%	2 games
TOTAL	100%	260 games

A small portion of the archive contained multiplayer games, including two-player games ($n = 29$), three-player games ($n = 3$), and four-player games ($n = 2$). Scratch seemed to facilitate both human-to-computer and human-to-human interaction design, but youth at the Clubhouse tended to design for human-to-computer interaction, particularly within one-player game environments. The archive shows that youth engaged heavily both in learning to program and in designing the types of interactions that players would have with one another as well as with the computer-crucial technology fluencies that underpin most contemporary technologies.

Discussion

In this article, we have presented empirical findings on how new fluencies were cultivated through game design. The Computer Clubhouse space, with its constructionist approach to design-based learning, allowed youth to build upon their emergent understandings of the elements of a video game, tasking them to search for their own solutions to problems that would arise from the process of designing their own games. By having a clear vision of what their final products should look like (more often than not, imitations of existing games or other types of media), youth were faced with finding the technical or creative means to reach these ends. The tinkering ability of Scratch, with its "low floors," "wide walls," and "high ceilings" (Resnick and Silverman 2005, p. 2), provided a platform for youth to explore the multiple components of game design, and Scratch's easily navigated series of control panels allowed youth to easily find the tools necessary to implement their ideas. This was crucial in the transformation of the youth from design novices to youth familiar with the multiple fluencies involved in the creation of the games they see everyday, those developed both by their peers and by professional designers in the commercial market.

Where previous approaches have focused on game design with specialized toolkits (solely emphasizing a limited array of design aspects), we propose using a more comprehensive, media-rich programming environment as a way to engage youth in reading and understanding game texts. We contend that learning to computer program became but one of many types of practices that youth engaged in while in the process of game design. Youth learned to juggle a wide range

of technical and creative fluencies that are complexly intertwined, helping them to organize and coordinate multiple events and types of meaning-making systems in the process. We call this intermix of technology and gaming practices “gaming fluencies” because youth became fluent not only in game design but in technology design. Our findings provide a perspective on how urban youths’ informal video game-making culture and practices can be used to support alternative pathways toward gaming fluencies and, more broadly, the new literacies important to 21st-century learning.

Jorge’s game design provides a compelling example of the gaming fluencies we observed being developed at the Computer Clubhouse, because it mimicked most closely its commercial inspirations. Jorge’s game is also notable for functioning more as a pedagogical tool than as an actual game in the Clubhouse space, both a showpiece for the programming potential of games designed by youth in Scratch and a springboard for several youths’ game-design aspirations. Most of the youth in the study recognized Jorge’s “Mortal Kombat” as a game and had fun with its mechanics. Rather than become fully immersed in game play, however, youth would study the various costume changes, how the characters interacted on screen if they hit one another, and the amount of programming that Jorge had done for the project. Soon after his “Mortal Kombat” debuted in the Clubhouse space, Jorge began to mentor other inspired youth in the creation of their own games.

One could argue that the game designs presented here were not full-fledged games and, thus, by extension could not engage youth in the type of “thinking like a game designer” and associated benefits as part of the gaming literacies approach proposed by Salen (2007) and Buckingham and Burn (2007). Yet Jorge’s game, as well as “Chandelle’s castle” and “Jerrell (hoops),” dealt with a host of complex interface design issues that reveal the underpinnings of software interactions. Such understanding is crucial for today’s citizenship, when more aspects of life have moved into the digital domain. Interfaces happen to be one of the most difficult artifacts to design, because many assumptions about human interaction are built in, assumptions that most people are not aware of unless faced with designing them. Jorge became acutely aware of these issues, and his game afforded him the opportunities to examine them with others in the Clubhouse. Because most school activities are concerned with using rather than producing technologies, we

have little history of educating students about interface design. But we argue that the transparency challenge (Jenkins et al. 2006) should be as concerned with opening the “black box” of digital technologies, particularly games, as it is with media ownership and control issues. In reviewing our case studies, we have pushed the transparency issue to a new level that involved the manipulation of the computer through computer programming and interaction design and how the Clubhouse created opportunities for youth to learn how to do this.

Game-design activities offer a promising avenue for young people to develop ownership of media and a sense of appreciation for what goes into creating them (see also Kafai, Fields, and Burke, forthcoming). However, with violation of copyright as close as one mouse click away, ethical concerns are prominent in digital culture. In the Clubhouse, the emphasis on repurposing media is akin to that which one would find in a professional context. Our observations of Clubhouse interactions lead us to suspect that members would feel a sense of transgression if other members were to copy their work without explicit acknowledgment. Jorge’s “Mortal Kombat” variations are good examples of how to modify others’ ideas in such a way that the reference to the originator of the idea is clear without being so minimal as to constitute a transgression of copyright.

Many of the design activities that took place concurrently with those activities involving game design—for example, creating images in Photoshop or 3D graphics in Bryce 5—would qualify Clubhouse participants to be members of a participatory culture. The current research, however, was focused on who was participating in the game-design culture. Earlier research has shown that the game industry is not a welcoming place for women and minorities (Consalvo 2008). At the Clubhouse we observed urban youth and English language learners engage in game-design activities and become participants in gaming communities. The scope of Clubhouse game production and the variety of game designs we observed provide evidence that game-making activities can authentically connect youth to multiple expert communities, including the game-design industry, the gamer community, programming communities, and visual arts and design communities. We see the use of game design as an appropriate and healthy counterpoint to a culture of consumption. While the boundaries between media consumers and producers are perhaps not as distinct as they used to be, a large rift still

separates those who own and control media from those who have the possibilities of creating media. To be a full member in today's participatory culture should mean much more than knowing how to play video games; it should also mean knowing how to create one.

This study contributes to the research on gaming literacies in several ways. First, it is one of the first studies to emphasize game design in the work on gaming literacies. The emphasis on game design invites us to think about the writing of, not just the playing of, these types of texts. Second, this study moves beyond a case-based approach to understanding gaming fluencies and instead looks at a community archive of work in order to ground our understanding and help us identify broader trends. Third, our focus on at-risk youth highlights the need for tools that do not further existing inequities in our educational system but instead allow youth from a variety of backgrounds and abilities to find their own place in the educational and game-influenced 21st-century digital media landscapes. Tools that allow for a wide range of project ideas (i.e., that offer "wide walls") are important because they allow youth like Jerrell, Chandelle, and Jorge to implement ideas that are personally meaningful to them without having to conform to a designer's aesthetics or to a choice of genre that may not be appealing to marginalized groups.

Acknowledgments

The research cited in this article was supported by a Dissertation Year Fellowship from the Spencer Foundation awarded to the first author and a grant from the National Science Foundation (NSF-0325828) awarded to the second author.

Note

1. All names in this study are pseudonyms.

References

- Buckingham, D., and A. Burn. 2007. Game literacy in theory and practice. *Journal of Educational Multimedia and Hypermedia* 16 (3):323–49.
- Cassell, J., and H. Jenkins, eds. 1998. *From Barbie to Mortal Kombat: Gender and computer games*. Cambridge, MA: MIT Press.
- Chi, M. T. H. 1997. Quantifying qualitative analyses of verbal data: A practical guide. *Journal of the Learning Sciences* 6 (3):271–315.
- Collins, A. 1992. Toward a design science of education. In *New directions in educational technology*, ed. E. Scanlon and T. O'Shea. Berlin: Springer-Verlag.
- Consalvo, M. 2008. Game over? A preliminary study of women game developers and factors influencing career success and failure. In *Beyond Barbie and Mortal Kombat: New perspectives on gender and games*, ed. Y. Kafai, C. Heeter, J. Denner, and J. Sun, 177–93. Cambridge, MA: MIT Press.
- Creswell, J. W. 2003. *Research design: Qualitative, quantitative, and mixed methods*. Thousand Oaks, CA: Sage Publications, Inc.
- Cuban, L. 1986. *Teachers and machines: The classroom use of technology since 1920*. New York: Teachers College Press, Columbia University.
- Denner, J., and S. Campe. 2008. What do girls want? What games made by girls can tell us. In *Beyond Barbie and Mortal Kombat: New perspectives on gender and games*, ed. Y. Kafai, C. Heeter, J. Denner, and J. Sun, 129–45. Cambridge, MA: MIT Press.
- Egenfeldt-Nielsen, S. 2007. *Beyond edutainment: The educational potential of computer games*. London: Continuum Press.
- Erickson, J., and R. Lehrer. 1998. The evolution of critical standards as students design hypermedia documents. *Journal of the Learning Sciences* 7 (3–4):351–86.
- Flanagan, M., and H. Nissenbaum. 2008. Design heuristics for activist games. In *Beyond Barbie and Mortal Kombat: New perspectives on gender and games*, ed. Y. Kafai, C. Heeter, J. Denner, and J. Sun, 265–81. Cambridge, MA: MIT Press.
- Guzdial, M. 2004. Programming environments for novices. In *Computer science education research*, ed. S. Fincher and M. Petre, 127–54. London: Routledge Falmer.
- Harel, I. 1991. *Children designers: Interdisciplinary constructions for learning and knowing mathematics in a computer-rich school*. Santa Barbara, CA: Greenwood Publishing Group, Inc.
- Hayes, E. R., and I. A. Games. 2008. Making computer games and design thinking. *Games and Culture* 3 (3):309–32.
- Hmelo, C. E., D. L. Holton, and J. L. Kolodner. 2000. Designing to learn about complex systems. *Journal of the Learning Sciences* 9 (3):247–98.
- Jenkins, H. 1992. *Textual poachers: Television fans and participatory culture*. New York: Routledge.
- Jenkins, H., K. Clinton, R. Purushotma, A. Robison, and M. Weigel. 2006. *Confronting the challenges of participation culture: Media education for the 21st century*. Chicago: John D. and Catherine T. MacArthur Foundation.
- Kafai, Y. B. 1995. *Minds in play: Computer game design as a context for children's learning*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Kafai, Y. B. 1996a. Gender differences in children's constructions of video games. In *Interacting with video*, ed. P. M. Greenfield and R. R. Cocking, 39–66. Norwood, NJ: Ablex Publishing Corporation.
- Kafai, Y. B. 1996b. Learning through making games: Children's development of design strategies in the creation of a computational artifact. In *Constructionism in practice*, ed. Y. Kafai and M. Resnick, 71–96. Mahwah, NJ: Lawrence Erlbaum Associates.
- Kafai, Y. B. 1998. Video game designs by children: Consistency and variability of gender differences. In *From*

- Barbie to Mortal Kombat: Gender and computer games*, ed. J. Cassell and H. Jenkins, 90–114. Cambridge, MA: MIT Press.
- Kafai, Y. B. 2006a. Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture* 1 (1):34–40.
- Kafai, Y. B. 2006b. Constructionism. In *Cambridge handbook of the learning sciences*, ed. K. Sawyer, 35–46. Cambridge, MA: Cambridge University Press.
- Kafai, Y. B. 2009. Serious games for girls? Considering gender in learning with games. In *Serious games: Mechanisms and effects*, ed. U. Ritterfeld, M. Cody, and P. Vorderer, 219–33. New York: Routledge.
- Kafai, Y. B., and C. C. Ching. 2001. Affordances of collaborative software design planning for elementary students' science talk. *Journal of the Learning Sciences* 10 (3):323–63.
- Kafai, Y., S. Desai, K. Peppler, G. Chiu, and J. Moya. 2008. Mentoring partnerships in a community technology center: A constructionist approach for fostering equitable service learning. *Mentoring and Tutoring* 16 (2):191–205.
- Kafai, Y. B., D. A. Fields, and W. Burke. Forthcoming. Entering the Clubhouse: Case studies of young programmers joining the Scratch community. *Journal of Organizational and End User Computing*.
- Kafai, Y., K. Peppler, and R. N. Chapman, eds. 2009. *The Computer Clubhouse: Constructionism and creativity in youth communities*. New York: Teachers College Press.
- Kafai, Y., K. Peppler, and G. Chiu. 2007. High tech programmers in low-income communities: Seeding reform in a community technology center. In *Communities and technologies: Proceedings of the third communities and technologies conference, Michigan State University*, ed. C. Steinfeld, B. T. Pentland, M. Ackerman, and N. Contractor. London: Springer.
- Kelleher, C. 2008. Motivating middle school girls: Using computer programming as a means to the end of storytelling via 3D animated movies. In *Beyond Barbie and Mortal Kombat: New perspectives on gender and games*, ed. Y. Kafai, C. Heeter, J. Denner, and J. Sun, 247–65. Cambridge, MA: MIT Press.
- Loftus, G. R., and E. F. Loftus. 1983. *Minds at play*. New York: Basic Books.
- Malone, T. W., and M. R. Lepper. 1987. Making learning fun: A taxonomy of intrinsic motivations for learning. In *Aptitude, learning and instruction, volume 3: Cognitive and affective process analyses*, ed. R. E. Snow and M. J. Farr, 223–53. Hillsdale, NJ: Erlbaum Associates.
- Maloney, J., L. Burd, Y. Kafai, N. Rusk, B. Silverman, and M. Resnick. 2004. Scratch: A sneak preview. Paper presented at the Second International Conference on Creating, Connecting, and Collaborating through Computing, Kyoto, Japan.
- Maloney, J., K. Peppler, Y. Kafai, M. Resnick, and N. Rusk. 2008. Digital media designs with Scratch: What urban youth can learn about programming in a Computer Clubhouse. Paper presented at the 2008 International Conference of the Learning Sciences, Utrecht, Netherlands.
- National Research Council. 1999. *Being fluent with information technology*. Washington, DC: National Academy Press.
- New London Group. 1996. A pedagogy of multiliteracies: Designing social futures. *Harvard Educational Review* 66 (1):60–92.
- Palumbo, D. B. 1990. Programming language/problem-solving research: A review of relevant issues. *Review of Educational Research* 60 (1):65–90.
- Papert, S. 1980. *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Pelletier, C. 2008. Producing difference in studying and making computer games: How students construct games as gendered in order to construct themselves as gendered. In *Beyond Barbie and Mortal Kombat: New perspectives on gender and games*, ed. Y. Kafai, C. Heeter, J. Denner, and J. Sun, 145–61. Cambridge, MA: MIT Press.
- Penner, D. E., R. Lehrer, and L. Schauble. 1998. From physical models to biomechanical systems: A design-based modeling approach. *Journal of the Learning Sciences* 7 (3–4):429–49.
- Peppler, K. Forthcoming. Media arts: Arts education for a digital age. *Teachers College Record* 112 (8).
- Peppler, K., and Y. Kafai. 2007. From SuperGoo to Scratch: Exploring creative digital media production in informal learning. *Learning, Media, and Technology* 32 (2):149–66.
- Perkins, D. N. 1986. *Knowledge as design*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Pinkett, R. D. 2000. *Bridging the digital divide: Sociocultural constructionism and an asset-based approach to community technology and community building*. Paper presented at the 81st Annual Meeting of the American Educational Research Association, New Orleans, LA.
- Provenzo, E. 1991. *Videokids: Making sense of Nintendo*. Cambridge, MA: Harvard University Press.
- Resnick, M., Y. Kafai, and J. Maeda. 2003. ITR: A networked, media-rich programming environment to enhance technological fluency at after-school centers in economically disadvantaged communities. Proposal [funded] submitted to National Science Foundation, Arlington, VA.
- Resnick, M., J. Maloney, A. M. Hernandez et al. 2009. Scratch: Programming for everyone. *Communications of the ACM* 52 (11):60–67.
- Resnick, M., and B. Silverman. 2005. Some reflections on designing construction kits for kids. Paper presented at the Interaction Design and Children Conference, Boulder, CO. <http://ilk.media.mit.edu/papers/IDC-2005.pdf>.
- Salen, K. 2007. Gaming literacies: A game design study in action. *Journal of Educational Multimedia and Hypermedia* 16 (3):301–22.
- Schön, D. A. 1983. *The reflective practitioner*. New York: Basic Books.
- Simon, H. A. 1981. *The sciences of the artificial*. Cambridge, MA: MIT Press.
- Spohrer, J. C., and E. Soloway. 1985. Putting it all together is hard for novice programmers. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Tucson, AZ.