Peppler, K. & Kafai, Y. B. (2010). Gaming Fluencies: Pathways into a Participatory Culture in a Community Design Studio. *International Journal of Learning and Media*, 1(4), pp. 1-14.

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

Kylie A. Peppler
Indiana University, Bloomington

Yasmin B. Kafai
University of Pennsylvania

Abstract. Many new literacies efforts promote creative media production as a way to foster youth' 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 design studio in a community technology center. Our analyses focused 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 give 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.

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

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. Along with others, 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 shapes perception, and to become knowledgeable of emerging ethical standards that shape their practices as media makers and participants in online communities. As 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 to consider digital games as a "cultural medium" in their own right and referred to the ability to critically read and produce game texts as "game literacy" (Buckingham & Burn, 2007) or "gaming literacies" (Salen, 2007).

In this paper, we focus on creative media production (Peppler & 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. Our efforts draw on early work in this area (Kafai, 1995; 2006a) and more recent studies on using game design to engage girls in new technologies (Denner & Campe, 2008; Flanagan, 2008), in articulating their thinking about gender (Pelletier, 2008), and in design thinking (Hayes & Games, 2007; Salen, 2007). There is, however, a distinct take in our approach not present in other efforts: we use game production not just as a way to promote gaming literacy in the broadest sense but also to enhance technology fluency particularly needed by disadvantaged youth (Kafai, Peppler & Chiu, 2007). For that reason, 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; Papert, 1991) that aims at developing pedagogies to promote technology fluency rather than computer literacy, thus moving beyond techno-centric perspectives.

To illustrate our view on gaming fluencies, we examine the artifacts of game production that took place in a Computer Clubhouse, a community technology center (Kafai Peppler, & Chapman, 2009). Youth in this 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 beginning in 2005 as part of a larger research project (Kafai, Peppler & 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% of the game archive that contained over 643

Scratch projects produced by Clubhouse members. The range of projects contained in the random sample is reflective of youths' evolving knowledge of game design developed and reshaped by the local community. We further examined three games in more detail to illustrate the design choices, interface features, and narrative elements used by youth. These investigations will allow us to address the following general research question: What do youth learn in the process of designing games? More specifically, we're interested in individual pathways into gaming fluencies, the various aspects of gaming fluencies that are expressed in their 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 for learning is a relatively recent addition to the fields of education, technology, and media literacy studies. The academic community in the 80s and 90s concentrated on the psychological impact of game play by discussing gender differences in spatial reasoning (Loftus & Loftus, 1980) 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, and 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 like they would focus on commercial or political intent in television advertisement or newspaper articles. The very idea of production in hands of the lay viewer or game player was met with resistance 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 motivational benefits (Malone & 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 thus 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 90's (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 "Sciences of the Artificial" (1981) and Schoen's "The Reflective Practitioner" (1983) 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 that illustrated how students as designers of instructional mathematics software could provide a long-term, meaningful, and integrated context for learning programming and mathematics (Palumbo, 1993). Many but not all of the educational approaches 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 & Ching, 2001), historical presentation (Erickson & Lehrer, 1998), mechanical device (Penner, Schaeuble, & Lehrer, 1998), or engineering design (Hmelo, Colton, & Kolodner, 2000) – but differed in how they conceptualized the design process and artifact in the service of learning. This last point merits further discussion later on as we will find similar differences in how game design is currently viewed in education.

These developments provided the background for one of the first empirical studies of game design in schools (Kafai, 1995). A class of 16 elementary 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 and created packaging and advertisements for their games. The designers met about once a month with their intended players, a group of younger students who provided them with feedback on various aspects of their games (see also Kafai, 1998a). The observed gender differences in games' narrative, components, aesthetics, and mechanics have received the most attention (Kafai, 1996a; 1998b) – 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 & Jenkins, 1998). Much less attention was given to the equally important aspects of design practices, collaborative planning, and public critiques that contributed to students' understanding (Kafai, 1996b) that resonate more with current approaches to game designs for learning.

Indeed, the 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 saw games as another example of multimodal texts that integrated sound, music, graphics, writing, and more. For them, game design combined 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 involved transformation of different resources and assets pertaining to games. These aspects were also captured in Salen's (2007) notion of gaming 'literacies' because it 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 experts in content areas specific to a game" (p. 318). Both of these proposals 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 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. While Jenkins and colleagues (2006) view the participation gap as the 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 apply it specifically to game production, a field which has been dominated by white males. This has been problematic for many reasons, including the lack of representation of women and minority avatars in games, the reduction of these groups to exaggerated stereotypes, and the overabundance of games marketed towards white males (for a review see Kafai, 2009). Our intention is that by providing opportunities for underrepresented youth to participate in making games, 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 that are oftentimes seen as consumers of new media and are rarely positioned as designers, especially as those of games.

It is perhaps in the conception of tools for game design where the differences between gaming literacies and fluencies become the most apparent. All of them 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, "[g]ame 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." (2007, p. 318). Those interested in gaming literacy or literacies have focused on creating game-specific toolkits such as Missionmaker (Buckingham & Burns, 2007) or Gamestar Mechanic (Salen, 2007), which focus on key elements in game production such as the system-based thinking "to enable a deep understanding of how a system of rules, behaviors, and relationships guide the design of an interactive experience, activated through play" (p. 314, Salen, 2007).

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) is one example, while Alice (Kelleher, 2008) or Peeps (Flanagan, 2008) are others of this type of general tool. Scratch projects allow designers to move fluidly between different media forms (Resnick & Silverman, 2005), games just being one of them, and allow for a variety of designers' artistic choices (Peppler, in press). 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 & Games, 2008). Yet, as we will argue in this paper, such a distinction is quite arbitrary and neglects to take into account the design process and community in which game design with Scratch is situated.

The type of design that we propose for gaming fluencies is based on a well-articulated theory of learning and subsequent pedagogy behind it, particularly one that emphasizes the expression of ideas, values and genres. A pedagogy with such a design orientation can be found in constructionism, which places learners in roles of creating artifacts that are of relevance to a larger community (Papert, 1980; Kafai, 2006b). While constructionism places importance on the individual learner, it also places equal importance on the role of social participation. Here the individual, the artifact, and collaborative input of the community shape learning, participation, and sharing. In the case of game production today, the community could be described as 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). Accordingly, a design tool like Scratch can promote the developmental relationship between the individual and the community and will enable youth to express their cultural heritage, have a broad communicative value, and allow for an information and resource exchange (Pinkett, 2000). Indeed, we have found evidence in previous analyses of game productions (Peppler & Kafai, 2007) that youth begin to understand how media shapes the perceptions of the world and the challenges of public ownership in media production – described by Jenkins and colleagues (2006) as the transparency and ethics issues respectively.

In this paper, we want to focus more explicitly on the creative and technical aspects in game production with Scratch. More specifically, we want to examine how choices of genre, narrative elements and visual designs in conjunction with the design of interface features lead Scratch designers in developing gaming fluencies. Like in Gamestar Mechanic, they can begin design activities with little to no prior knowledge but also push the boundaries to build complex designs (Peppler & Kafai, 2007). Unlike more specific game design tool kits, we would expect a broader range of game designs in Scratch given that the tool has what Resnick and Silverman (2005) called “wide walls”, allowing for a range of design choices. Furthermore, we wanted to identify which aspects of game design in Scratch involve the type of systems-based thinking that Salen 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 does so in the context of 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 & Kafai, 2007). The studio was situated at a storefront location in one of the city’s poorest areas and served over 1000 high-poverty African American and Hispanic youth. Youth worked individually and in small groups on projects, and ranged from 8-18 years of age, but most were between the ages of 10-14. A community of mixed-age and -ability learners, consisting of novice and expert youth and an array of local and partnering University mentors, supported game design efforts. However, it’s essential to note that mentors were novices with computer technology and had little to no programming or game design experience (Kafai et al., 2008).

At the particular Clubhouse where this study was conducted, all computers 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 the programming environment, Scratch, as well as Microsoft Office, Bryce 5, Painter 7, RPG Maker™, in addition to video, photography, and sound editing software. Scratch, in particular, differs from other visual programming environments (Guzdial, 2004) by using a familiar building block command structure (Maloney, Burd, Kafai, & Rusk, 2004; Resnick, Kafai, & Maeda, 2003), 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 enabling greater creative expression and can be easier entry for novices than three-dimensional programming.

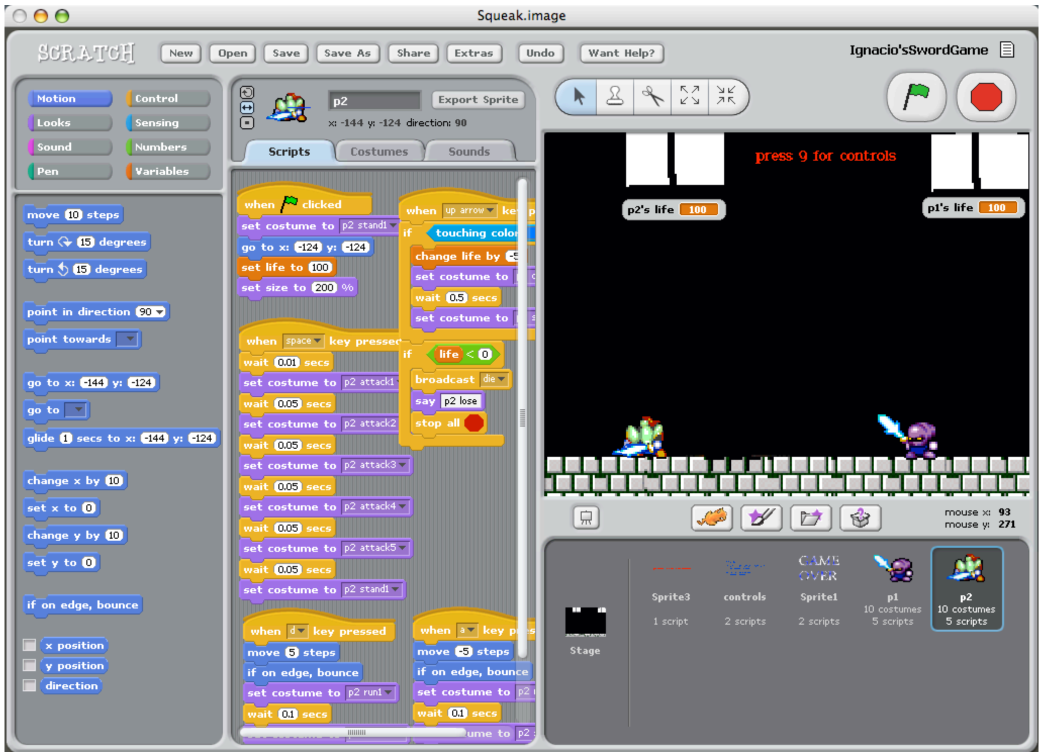

Figure 1: Screenshot of the Scratch user interface with sample game, “Ignacio’s SwordGame”.

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*, allowing youth to control and manipulate sound, images, motion, and various types of input from the players. In the lower right side of the screen there 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, stacks of commands that the creator has stacked together to control a particular sprite are displayed in the center panel. If the other two tabs at the top of this panel were to be clicked, information about the sprite’s costumes (see Figure 5) or sounds would be displayed. 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.

The Scratch vocabulary of roughly ninety commands includes commands for motion, image transformations (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 done 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 in 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% 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. In particular, 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, etc.); the second was used to capture the creative aspects of game design (e.g., type of animation used and aesthetic elements like perspective, motion paths, audio design); 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). The goal of these sets of codes was to capture the range of game design fluencies that emerged over the course of the study.

In addition, we collected detailed profiles of participants, which involved interviews, videotaped observations, extensive ethnographic field notes (Creswell, 2003), and other artifacts. As part of our observations of the emerging game design culture at this site, we focused additionally on three case studies of how individuals of different backgrounds and abilities navigated this landscape. Our 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. In the following section, we introduce case studies and situate them within the larger community of game designers, giving examples from the field notes and interviews where they describe literacy events and practices, with an emphasis on those involving game design.

Game Productions in the Design Studio

We begin by providing an overview of the range of digital design productions in the Computer Clubhouse. Based on an analysis of all files (n = 11,926) saved on the central server during the first 24 months of introducing Scratch we found that saving of images on the web (> 75% of the files saved, n = 8944) was the most popular activity, followed by Scratch (5.3% of the files saved, n = 643) as the most popular design software used during this period. Other popular design software included *Microsoft Word* (3.5% files saved, n = 461), *Bryce5* (2% of the files saved, n = 270), and *Kai's SuperGoo* (1.1% of the files saved, n = 143). Notably further down on the list of popular design software is RPG Maker game files (n = 43), despite its presence in the community for a number of years prior to the introduction of Scratch. While we focus in the following sections on game designs in Scratch alone, these numbers provide a general illustration of the popularity of digital design production in which youth were engaged. We think that Scratch's popularity was due in large part because its ease of use and flexibility allowed for a greater variety of game genres, components, and aesthetics. In addition, Scratch was quick to be adopted in the Clubhouse space 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 have been created within the design culture, but what can we say about the creators, themselves? Overall, the Clubhouse serviced over 1000 youth, 252 of whom created an individual folder on the central server. This was one of the first initiation practices of becoming a member of the design studio. Based on the Computer Clubhouse's estimate, this would indicate that about one quarter of the youth had been introduced by peers, mentors, or coordinators to creating a folder and saving work (n = 252). Of this group, over 80 youth saved a Scratch project to the server, roughly 8% of all Clubhouse youth. Overall, 40% of the Scratch archive was created by male members, 29% by female members, 9% by small groups, and 22% have unknown designers.

In the second year of the project, we saw that some new aspects of a video game making culture emerged. For example, video game production was a high status marker, local and global game design experts emerged, and work in Scratch established membership within the community (Kafai, Peppler, & Chiu, 2007). In addition, we began to see peer-to-peer mentoring in video game design for the first time and there was increased appropriation of Scratch as a video game design tool. As a result, new types of game genres emerged and individuals worked together in groups with increased frequency, with some youth specializing in aspects of game design akin to more professional settings, such as graphic designers, animators, and programmers.

The archive of youth's 643 Scratch projects became the source material for our case studies and analyses of game designs to illustrate the different access points used by youth to engage with video game making and highlight what we consider to be significant events and practices surrounding game production. In the following sections, we present the three case studies exemplifying how youth navigated game production with Scratch, gaining fluency in a wide array of academic areas important to game design. We follow with an archival analysis, which sets a broader context for understanding the pathways towards game design fluency illustrated in these three cases.

Game Designs: Three Case Studies

In the following case studies of game designs we illustrate 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. We selected these three cases because they represent three very different approaches to game genres and they also varied in complexity of programming and visual design.

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 since the sport theme itself provides the context. The game itself also lacks any audio, such as voiceover, soundtrack or any other types of sounds. When the player presses a start button (the green flag in the upper corner of the Scratch interface), directions pop on screen and spin, saying "Freethrow in 60 seconds." 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

[1] All names in this study are pseudonyms.

basketball and says "Game Over", at which point everything is brought to a stop and the scoreboard ceases to register additional points.

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

While there isn't a win/lose condition explicitly built into the game, youth at the Clubhouse were quick to try to beat their personal best score or their peers', creating a de facto win/lose experience. The game itself was coded as having two dimensions, since the ball and other objects appear to be in a single plane with no background receding into the distance. This game design only utilized programmed animation effects (such as move commands) to make the ball move, but used a wide array of programming concepts in the creation of his game, 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, is an 11-year-old girl of mixed African-American/Latina heritage. She attended a public school near the Clubhouse location and was in the 7th grade at the time of this study. At the point of creating this piece, she had attended the Clubhouse sporadically for a year or two and had occasionally been involved in some aspects of creative production. The one-player game, "chandelle castle," was Chandelle's second Scratch project. The game features a narrative (or plotline) that is both prominent and familiar: a princess is locked in the tower of a castle by an evil witch and needs to be rescued. This game was categorized as an interactive narrative in the genre analyses. The game features a methodically designed background consisting of a distant castle in the middle of the screen in between two contrasting panels of color. Chandelle created a bi-colored 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 sky). Chandelle was one of the few Clubhouse youth that utilized 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, the princess calls for help and 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 3a). The player must find the princess by clicking around on the screen. Once the player clicks on the princess, the knight parachutes in from the sky and lands on the bridge, announcing to the witch, "stop right there," to rescue the princess (see Figure 3b).

Chandelle's game design didn't include written directions, and relied on verbally communicating the rules of the game to other players at the Clubhouse. In addition, other types of game components such as scoreboards, timers, or other types of components aside from her Sprites were absent. While there was a win/lose component (i.e., you have to find the princess to win), there was no time limit to the game. This win or lose component was actually accidentally discovered by Chandelle, as she was demonstrating the game to a group of Clubhouse boys. When she had trouble finding the princess herself, the boys perked up and all volunteered to give it a try. The *chandelle castle* project utilized several types of animation, including limited stop-motion animation (because of its limited number of frames), programmed visual effects (such as the "help me" getting larger), and programmed animation (movement). Chandelle's game was also one of the few to use three-dimensional perspective in the background and the motion of the characters appeared to move in three-dimensions. Chandelle's project utilized 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.

Figures 3a and 3b: Two screenshots of chandelle castle in play mode at various time points in the narrative.

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 videogame, *Mortal Kombat: Deadly Alliance*. The original Mortal Kombat (MK) is a series of commercial fighting games that began in arcades and quickly became popular for at-home consoles. The game is especially known for its use of digitized sprites (as opposed the hand-drawn sprites of other contemporary games), the specialized key moves that require a complicated sequence of buttons, and, perhaps most notoriously, for the inordinate amount of graphic violence.

Sharing many similarities to the original, Jorge's Mortal Kombat was also a fighting game. 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). This game was coded as a two-player game since it allowed for two separate people to control separate avatars. As the players fight, they are taken through a series of 16 backgrounds, which were intended to act eventually as levels. The backgrounds in the Mortal Kombat games contained three-dimensional images that were found on the web and downloaded for use in his project. While the background had an illusion of a three-dimensional play space, the characters moved in two-dimensions (i.e., up, down, right, and left).

After finding that players had difficulty remembering the buttons needed to control the sprites, Jorge posted directions on the first level of his game in the upper right hand corner of the play screen. Based on these experiences, Jorge deduced why there might be conventional keystrokes to control on-screen sprites (e.g., arrow keys) because they were easier to remember and didn't require overlapping areas of the keyboard, indicating that he was learning something about human-to-computer interaction design. In order to build this type of interactivity, Jorge used several different programming concepts repeatedly in his project, including sequential execution (i.e., a stack of commands with more than one block), threads (i.e., multiple scripts running in parallel), conditional statements (i.e., if then statements), and user interaction (i.e., use of keyboard or mouse input).

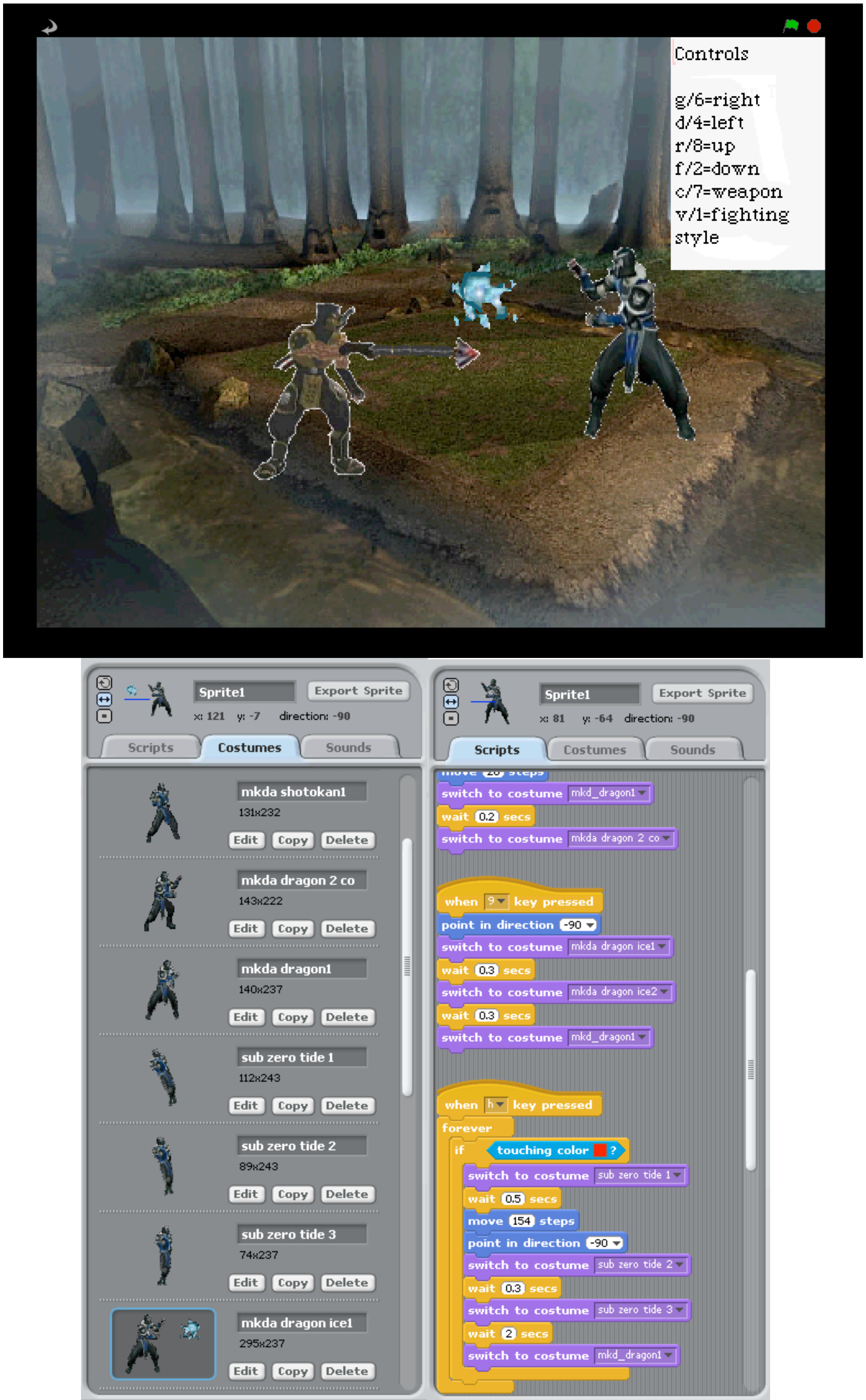

Figure 4: Partial screenshots of Jorge's *Mortal Kombat* game.

In addition to the computer programming in the project, each sprite has 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 an file downloaded from the web and imported into Scratch but was altered by Jorge's hand manipulation of the original image to create a fluid effect of animation. In this case, the figures looked as if they were throwing weapons at one another, walking, and spinning. In addition to being coded for its stop-motion animation, this game was coded for its use programmed animation effects (such as moving up, down, right and left). There were no programmed visual effects (other than the stop-motion animation) found in this game. Mortal Kombat lacked a strong narrative (other than two characters fighting one another). Jorge was ultimately unable to finish this particular game because of a switch between Alpha and Beta versions of Scratch that took place at the Clubhouse while he was working. At this point, he began his own Metal Slug series of games (Peppler & Kafai, 2007).

Development of a Game Design Community: Archival Analyses

These three case studies illustrated the diversity of game designs created in the Clubhouse community. But how representative were these game designs in terms of the full range of gaming fluencies? The archive of Scratch game projects stored on the central server allows us to address this question. In order to better understand this, we took a closer look at 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 design) found in the projects.

Game Genres. As a starting point, we coded for the range of game genres produced by the youth (see Table 1). We found that 50.8% were interactive narrative projects (e.g., animations, interactive art and narrative games); 7.3% focused on sports games; 2.3% were simulation games; and the remaining 11.1% were distributed among the other games categories, such as mazes, rhythmic games, role playing games, interactive shooter games, racing games or platform games. Augmenting this finding, we observed from the field notes that several members tended to cluster together to share ideas and provide support for each other's designs within certain genres, indicating the existence of numerous subcultures of game makers. Some projects defied categorization; for example, 22.3% of projects were graphics-only files without any game mechanics associated with the project and 8.5% were empty projects with titles only. Neither of these categories was coded for genre.

Table 1: Table of game genre types found in a random sub-sample of the archive.

Game Genres		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%

Overwhelmingly at this particular Clubhouse site, youth were very interested in creating interactive narratives. 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. Overall, the narratives took three forms: they could contain voiceovers that had narrative elements, they could contain thought balloons or speech bubbles that advanced a storyline, or, conversely, they could have visual elements that told a story without the use of words. Across all of the game genres, storytelling was present in nearly half, or 47%, of the games (n = 122), most of which were found in the interactive narratives or animations.

We further looked across the game genres to determine which game components were most commonly used. We found that 20 of the games contained scoreboards, seven of the games contained timers, and nearly all—or 240 games of the 260 in the selected archive—contained Sprites or central game characters, many of which containing multiple Sprites (averaging 5.76 Sprites per game).

Creative Aspects. There were three broad classes of design elements that we examined when we were considering the creative aspects of games:

types and extent of animation, use of perspective, and design of sound. Many of these Sprites were animated in various ways, the most common approach being programmed animation, where the creators used movement commands to animate static (or sometimes stop-motion animated Sprites). In the archive, we found that 181 games contained programmed animation effects. One common example of this type of animation would be to make a figure glide across the screen. The second type of animation came from programmed visual effects, ranging from a changing color effect to an increase/decrease in size. In sum, 117 of the games contained some sort of visual effect design. The last type of animation, stop-motion animation, which was the original technique used to create an animated sequence, was used in 69 of the games. Stop-motion animation is created from a series of hand drawn or downloaded frames that are independent images but when changed rapidly in close secession, creates an illusion of movement. These findings demonstrate that the Clubhouse culture was knowledgeable in some aspects of animation and game design, as evidenced through the usage of animation and other sophisticated gaming elements in youths' work.

Correspondingly, we examined some of the aesthetic elements found in many commercial games. In other work we have explored the artistic content of the youths' projects (Peppler, in press), which found that youth at the Clubhouse were engaged in learning about the professional practices of media artists and several of the works resembled that of contemporary professionals in the field. Here, we focused on the type of perspective (two-dimensional and three-dimensional perspective) 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 an additional 173 games used three-dimensional graphics or motion paths. The remaining games did not have any two- or three-dimensional graphics or motion paths associated with the projects. 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 further examining the technical aspects of game design fluencies, we considered both the programming and type of interaction design used in the project. Programming of the games became more complex over time, as indicated by the increasing number of stacks, Sprites, sounds, costumes, as well as the types of programming concepts contained in the projects during the second year of observation (Maloney, Peppler, Kafai, Resnick, & Rusk, 2008). These concepts included user interaction, loops, conditional statements, communications and synchronization, Boolean logic, variables, and random numbers, which are difficult concepts even for novice computer science majors at the college level (Spohrer & 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 games 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 through clicking on-screen objects while 8 games allowed the player to interact with the game through mouse movements along an x and y axis, or through mouse clicks to control aspects of the game (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. This suggests that more complex forms of human-to-computer interaction need to be an explicit part of the exploration in Scratch.

Table 2: Types of Human to Computer Interaction Design Used 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%

Table 3: Types of Human-to-Human Interaction Design Used 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**

We also examined the human-to-human interaction design present in these games (see Table 3). While 80 of the games had no form of interaction, 146 of them were one-player games or other types of projects. A small amount 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 types of interaction design, but youth at this site tended to design for human-to-computer interaction, particularly within one-player game environments. Across the projects, we can see that youth engaged heavily both in learning to computer program as well as designing the types of interaction that the player would have with each other as well as the computer – crucial technology fluencies that underpin most of our technologies today.

Discussion

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

While previous approaches have focused on game design with specialized toolkits (solely emphasizing a limited array of design aspects), we proposed using a more comprehensive, media-rich programming environment as a way to engage youth in reading and understanding game texts. We contended that learning to computer program became but one of many types of practices that youth engaged in while in the process of game design. Instead, 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 proposed to call this intermix of technology and gaming practices "gaming fluencies" because youth became not only fluent in game design but also in technology design. The findings provided 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.

In this discussion, we want to take the opportunity to reflect on aspects of gaming fluencies that we saw develop in this Computer Clubhouse. Jorge's game design provides a compelling example, mostly because it mimicked most closely its commercial inspirations. It is also notable for the way that it acted as more of a pedagogical tool than an actual game in the Clubhouse space, both a showpiece for the programming potential of games designed by youth in Scratch, as well as 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 the mechanics. Rather than getting fully immersed in game play, youth would come up and 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 in the project. Soon after his Mortal Kombat debuted in the Clubhouse space, Jorge began to mentor other inspired youth in the creation their own games.

One could argue that the case studies presented here were not full-fledged games, and thus by extension could not engage the youth in the type of "thinking like a game designer" and associated benefits as part of the gaming literacies approach proposed by Salen and Buckingham. Yet Jorge's game, and also those of Chandelle's castle and Jerrell's hoops, dealt with a host of complex interface design issues that reveal the underpinnings of software interactions. Such understandings are crucial for today's citizenship, as more aspects of life have moved into the digital domain. Interfaces happen to be one of the most difficult artifacts to design, as 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. In today's media culture, we lack a history of educating students about interface design, as most school activities are concerned with using rather than producing technologies. But we argue that the transparency challenge should be equally concerned with "opening the black box" of digital technologies, particularly games, as about media ownership and control issues. In reviewing our case studies, we have pushed the transparency issue to a new level that involved the participation and manipulation of the computer through computer programming and interaction design and how the Clubhouse created opportunities to youth to learn how to do this.

Equally important are ethical concerns prominent in digital culture, as violation of copyright can be just one mouse click away. We argue that game design activities offer a promising avenue for young people to develop ownership of media and a sense of appreciation what goes into creating them (see also, Kafai, Fields, & Burke, 2010). In the Clubhouse, there is an emphasis on repurposing media akin to a professional context. But there is also a sense of transgression if other members were to copy someone else's work without explicit acknowledgment. Jorge's Mortal Kombat variations are a good example of how the modifications within a genre can be minimal but still reference its originator.

We used our research at a Computer Clubhouse as a context for game design activities. We should note upfront that many other design activities took place concurrently that would well qualify the Clubhouse participants to be members of a participatory culture, including youths' creations in Photoshop, 3D graphics in Bryce, etc. But for us, the most important aspect relates to who is participating in the game design culture. Based on prior research, we know that the game industry is not a welcoming place for women and minorities (Consalvo, 2008). Here, we have seen urban youth and English Language Learners engaged in game design activities and become participants in gaming communities. Through the general scope of Clubhouse game production and the extent of case studies' game designs, we can find evidence that game-making activities can provide multiple pathways into participation through a mix of game genres and project ideas. We see the approach of gaming fluencies 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, there is still a large rift between those who own and control media and those that have the possibilities of creating them. To be a full member in today's participatory culture should mean much more than knowing how to play videogames; it should also mean knowing how to create one.

This study contributes to the research on gaming literacies in several respects. First, to our knowledge this is one of the first studies to emphasize game design in the work on gaming literacies, which moves us to thinking about the writing and not just the playing of these types of texts. Second, this study moves beyond a case-based approach to understanding game design fluencies, and instead looks across a community archive of work to ground our understanding, which helps us to move beyond isolated cases to larger trends in the database. And lastly, our focus on at-risk youth alerts us to the need for tools that allow youth from a variety of backgrounds and abilities to find a place in this landscape in order to not further existing inequities in our educational system. Along these lines, it's important to have tools that allow for a wide range of project ideas (i.e., wide walls) so that youth like Jerrell, Chandelle, and Jorge can all implement ideas that are personally meaningful to them without having to conform to the designer's aesthetics and choice of genre that may not be appealing to marginalized groups.

References

Buckingham, D. & Burn, A. (2007). Game literacy in theory and practice. *Journal of Educational Multimedia and Hypermedia*, *16*(3), 323-349.

Cassell, J. & Jenkins, H. (1998). (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. *The Journal of the Learning Sciences, 6*(3), 271-315.

Collins, A. (1992). Toward a design science of education. In E. Scanlon & T. O'Shea (Eds.) *New directions in educational technology*. Berlin: Springer-Verlag.

Consalvo, M. (2008). Game over? A preliminary study of women game developers and factors influencing career success and failure. In Y. Kafai, C. Heeter, J. Denner, & J. Sun (Eds.), *Beyond Barbie and Mortal Kombat: New perspectives on gender and games* (177-193). 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. & Campe, S. (2008). What do girls want? What games made by girls can tell us. In Y. Kafai, C. Heeter, J. Denner, & J. Sun (Eds.), *Beyond barbie and Mortal Kombat: New perspectives on gender and games* (129-145). Cambridge MA: MIT Press.

Egenfeldt-Nielsen, S. (2007). *Beyond edutainment: The educational potential of computer games*. London: Continuum Press.

Erickson, J., & Lehrer, R. (1998). The evolution of critical standards as students design hypermedia documents. *The Journal of the Learning Sciences, 7*(3 & 4), 351-386.

Flanagan, M. & Nissenbaum, H. (2008). Design Heuristics for Activist Games. In Y. Kafai, C. Heeter, J. Denner, & J. Sun (Eds.), *Beyond Barbie and Mortal Kombat: New perspectives on gender and games* (265-281). Cambridge, MA: MIT Press.

Guzdial, M. (2004). Programming environments for novices. In S. Fincher and M. Petre (Eds.), *Computer Science Education Research* (pp. 127-154). London and New York: Routlegde 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. & Games, I. A. (2008). Making computer games and design thinking. *Games & Culture, 3*(3), 309-332.

Hmelo, C.E., Holton, D.L., & Kolodner, J.L. (2000). Designing to learn about complex systems. *Journal of the Learning Sciences*, 9(3), 47-298.

Jenkins, H., Clinton, K., Purushotma, R., Robison, A., & Weigel, M. (2006). Confronting the challenges of participation culture: Media education for the 21st century. White Paper. Chicago, IL: The John D. and Catherine T. MacArthur Foundation.

Jenkins, H. (1992). *Textual poachers: Television fans and participatory culture*. New York: Routledge.

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 P. M. Greenfield & R.R. Cocking (Eds.). *Interacting with video* (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 Y. Kafai & M. Resnick (Eds.), *Constructionism in Practice* (pp. 71–96). Mawhaw, NJ: Lawrence Erlbaum Associates.

Kafai, Y. B. (1998a). Video game designs by children: Consistency and variability of gender differences. In J. Cassell & H. Jenkins (Eds.), *From Barbie to Mortal Kombat: Gender and Computer Games* (pp. 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 K. Sawyer (Ed.), *Cambridge Handbook of the Learning Sciences* (35-46). Cambridge, MA: Cambridge University Press.

Kafai, Y. B. & Ching, C. C. (2001). Affordances of collaborative software design planning for elementary students' science talk. *The Journal of the Learning Sciences*, *10*(3), 323–363.

Kafai, Y., Desai, S., Peppler, K., Chiu, G. & Moya, J. (2008). Mentoring partnerships in a community technology center: A constructionist approach for fostering equitable service learning. *Mentoring & Tutoring*, *16*(2), 191 - 205.

Kafai, Y. B., Fields, D. A., & Burke, W. (2010). Entering the Clubhouse: Case Studies of Young Programmers Joining the Scratch Community. *Journal of Organizational and End User Computing.*

Kafai, Y., Peppler, K., & Chiu, G. (2007). High tech programmers in low-income communities: Seeding reform in a community technology center. In C. Steinfeld, B. T. Pentland, M. Ackerman, & N. Contractor (Eds.), *Communities and technologies: Proceedings of the third communities and technologies conference, Michigan State University, 2007*. London, UK: Springer.

Kafai, Y., Peppler, K. & Chapman, R.N. (Eds.) (2009). *The Computer Clubhouse: Constructionism and creativity in youth communities.* New York: Teachers College Press.

Kelleher, C. (2008). Motivating middle school girls: Using computer programming as a means to the end of storytelling via 3D animated movies. In Y. Kafai, C. Heeter, J. Denner, & J. Sun (Eds.), *Beyond Barbie and Mortal Kombat: New perspectives on gender and games* (247-265). Cambridge, MA: MIT Press.

Loftus, G. R., & Loftus, E. F. (1983). *Minds at play*. New York: Basic Books.

New London Group. (1996). A pedagogy of multiliteracies: Designing social futures. *Harvard Educational Review*, *66*(1), 60-92.

Malone, T. W., & Lepper, M. R. (1987). Making learning fun: A taxonomy of intrinsic motivations for learning. In R. E. Snow & M. J. Farr (Eds.). *Aptitude, learning and instruction, volume 3: Cognitive and affective process analyses* (223-253). Hillsdale, NJ: Erlbaum Associates.

Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). *Scratch: A sneak preview.* Paper presented at the Second International Conference on Creating, Connecting, and Collaborating through Computing (Kyoto, Japan).

Maloney, J., Peppler, K., Kafai, Y., Resnick, M. & Rusk, N. (2008). *Digital media designs with scratch: What urban youth can learn about programming in a computer clubhouse.* Proceedings published in the 2008 International Conference of the Learning Sciences (ICLS) held at the University of Utrecht, Utrecht, Netherlands.

National Research Council (1999). *Being Fluent with information technology*. Washington, DC: National Academy Press.

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 Y. Kafai, C. Heeter, J. Denner, & J. Sun (Eds.), *Beyond Barbie and Mortal Kombat: New perspectives on gender and games* (145-161). Cambridge MA: MIT Press.

Penner, D.E., Lehrer, R., & Schauble, L. (1998). From physical models to biomechanical systems: A design-based modeling approach. *Journal of the Learning Sciences*, 7(3&4), 429-449.

Peppler, K. (2010). Media arts: Arts education for a digital age. *Teachers College Record,* 112(8).

Peppler, K. & Kafai, Y. (2007). From SuperGoo to Scratch: Exploring creative digital media production in informal learning. *Learning, Media, and Technology*, *32*(2), 149–166.

Perkins (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 (AERA), New Orleans, LA.

Provenzo, E. (1991). *Videokids: Making sense of Nintendo*. Cambridge, MA: Harvard University Press.

Resnick, M., Kafai, Y., & Maeda, J. (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. & Silverman, B. (2005). Some reflections on designing construction kits for kids. Proceedings of Interaction Design and Children Conference, Boulder, CO.

Resnick, M., Maloney, J., Hernandez, A.M., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Roenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for everyone. *Communications of the ACM*, *52*(11), 60-67.

Salen, K. (2007). Gaming literacies: A game design study in action. *Journal of Educational Multimedia and Hypermedia, 16*(3), 301-322.

Schön, D. A. (1983). *The reflective practitioner*. New York: Basic Books.

Simon, H.A. (1981). *The science of the artificial.* Cambridge, MA: MIT Press.

Spohrer, J.C. & Soloway, E. (1985). Putting it all together is hard for novice programmers. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, March, Tucson, AZ.