Creative Coding: Programming for Personal Expression

Kylie A. Peppler
University of California, Los Angeles
2148 Moore Hall
Los Angeles, CA 90095-1551
Phone: 001-310-497-8921
Email: kpeppler@gmail.com

Yasmin B. Kafai
University of California, Los Angeles
2331 Moore Hall
Los Angeles, CA 90095-1551
Phone: 001-310-206-8150
Email: kafai@gseis.ucla.edu

ABSTRACT
Media Arts within primary and secondary education is a relatively new avenue of research. Within the context of the arts classroom, rarely is learning to program emphasized despite its importance for creative expression in a digital medium. We present outcomes from an extensive field study at a digital studio where youth accessed programming environments emphasizing graphic, music and video. Learning the language of creative coding is essential to expression in a digital medium — one with increasing importance for youth and society at large. Here, we argue that it's not just in the viewing or playing of digital media but also in the constructive — or coding — experience through which connections to art can be established.

KEY WORDS
Art, design, digital art, education, education design, media art.

INTRODUCTION
As John Maeda states in his book, *Creative Code*, computer technology "is not a tool; it is a new material for expression" [14, p.101]. Defining new work at the intersection of arts and technology is problematic as the field is continually changing and rapidly expanding. We have chosen, for the purposes of the paper, to use the term "media art" to encompass all forms of creative practice involving or referring to art that makes use of electronic equipment, computation, and new communication technologies. For the most part, this is the preferred term in the professional field, however, the terms digital arts, media, or new media are also used interchangeably [20, 25].

Accordingly, students need to cultivate the specific skills and competencies needed for expression in this new material. Importantly for primary and secondary education, there has been little recognition of the need for programming skills especially in the context of the arts classroom [2]. Recognition of digital media (i.e., computers) as a viable art form can help students achieve literacy goals in both the arts and information technologies (IT), expanding the scope and capabilities of both forms. This paper argues that it is not just in the viewing or playing of digital media but also in the constructive—or coding—experience that connections to art can be established.

Previously, various forms of multimedia design, particularly that which uses *Adobe Photoshop* and *Hyperstudio* software, have been taken as the main artistic expression of digital media in primary and secondary educational settings [1, 29, 30] whereas professional artists are using advanced programming to manipulate and create digital expressions [14, 28]]. If there is any movement in education, it has not paralleled the developments in professional art. Rarely in primary or secondary arts education is the need for kids to learn about new technologies discussed, and even less commonly is there a call for them to learn programming skills.

Further complicating the matter, even when computer science is a part of the traditional curriculum, programming projects tend to focus on mathematical and science content [21, 23]. As an alternative, we emphasize programming projects that use graphic, music and video — media that have been found to be at the core of technology interests for youth and also making them more amenable to an arts environment [24]. This approach situates programming within a more media-rich context, focusing less on the manipulation and interaction with numeric symbols but more on digital media [16, 26]. Media arts projects like these use programming as a means to understand the production and manipulation of familiar media. These types of design activities allow kids to move from consumers of media, software and technologies to producers as they create new work. In an effort to introduce the essentials of programming to young artists, we argue that learning to code creatively is an important building block for creation and expression in digital media.

It is also important to note that, as evidenced by the artistic contributions in the Prix Ars Electronica festival, where artists as young as 8 years of age have received recognition in the "u19 freestyle computing" competition [13], youth are already more than consumers of new media, they are creators. In this competition, youths' work is increasingly sophisticated producing animation shorts, web design, and digital music. However, prevalent these few young artists may be, there is dearth of activity in formal and informal educational settings as well as women and minorities are still underrepresented in the submissions for such competitions. Issues of equity and access are not absent when considering media arts. Our focus is to examine the potential of creative coding, or programming, to open alternative venues into digital technologies. In this paper, we present outcomes from an extensive field study at a studio where youth accessed programming environments emphasizing graphic, music and video. The young artists in this study use programming to alter or change media that they have accessed via the Web. The resulting product is presented here with a more detailed analysis of both the computational and artistic elements. These elements connect the artist to the value of artistic production and a deeper understanding of what they would otherwise only consume.

CONTEXTS FOR CREATIVE CODING
Our work can be situated within diverse efforts to introduce programming in an artistic manner to students. Before presenting the outcomes of an extensive field study at a digital design studio in Los Angeles, California, we will expand on a few examples of existing contexts for creative coding: videogame art; computational crafts or robotics; and animations or computer graphics. We will then provide an in depth example from our

work so that the reader has a better understanding of how programming can become creative within these projects. Our efforts, as well as the other bodies of work presented here, give youth opportunities to create while learning the foundations in computer science, media arts, and engineering, ultimately preparing them for wider cultural participation.

Videogame Art

For more than a decade, youth have been known to engage in playing videogames. The *Minds in Play projects* [11] provide an example of how students can engage in making video games when creating instructional games to teach fractions to younger students in their schools – a topic taught in primary mathematics. Studies have found that the game design activity offered a microworld in which both girls and boys could situate their preferred ideas and fantasies [12]. Additionally, in student choices of game themes and the programming of animation and interactions, they offered a glimpse into what they found appealing and unappealing in the games and stories they experience through other media. Making a game and its rules allowed the game designers to be in charge and to determine the player's place and role in a virtual world, with all the consequences. Young designers use code to create a novel videogame environment, which has many artistic elements in its own right.

There are a growing number of video gaming camps for young designers: One example is the Urban Video Game Academy in Baltimore, Washington, and Atlanta that holds free video-game creation seminars for at-risk teens [19]. Mario Armstrong, a technology correspondent for National Public Radio and an academy co-founder, said that in addition to computer programming skills, the classes expose students to geometry, algebra, physics, art and music. The program is designed to educate teenagers about career opportunities in game creation, to help change an industry in which minorities and women are often depicted in negative stereotypes and rarely as the heroes. However, no formal educational studies have actually investigated these settings.

Computational Craft and Robotics

Although the context of videogames offers a two-dimensional medium for exploring creative coding, there are other artists and educators working to incorporate coding and computation in three dimensions. One such example is the work of Michael Eisenberg, who has explored the arena of computational crafts and more specifically *HyperGami* [4]. The computational system, *HyperGarmi*, allows kids (and adults) to explore solid geometry in the context of expressive construction of polyhedral models and sculptures. Creating original paper sculptures with *HyperGami* entails the design and construction of novel polyhedral forms derived from simpler shapes. Students create, alter, and compile code using special software, sculpting a variety of three-dimensional forms, and ultimately creating innovative origami-like sculptures.

Another way for students to get involved in creative coding is found in the field of robotics and the computational construction kits created by the MIT Media Lab. One such example is a Robotics Design Studio, offered at Wellesley College, exposing Liberal Arts majors to the "big ideas" of engineering and programming [32]. Students from diverse educational backgrounds work over the course of a month to create robotic sculptures made out of LEGO© parts, sensors, motors, and small embedded computers and were programmed using *HandyLogo* software. This approach allowed room for artistic expression and resulted in a wide array of projects such as *Loom*, which could actually perform elementary weaving. Similar to our approach, the Robotics Design Studio utilized a "Creative Engineering Gallery" for displaying its artistic products at the end of the course. The gallery proved to be an important non-competitive learning tool, allowing students to discuss their work and gain feedback from a larger audience.

Animation and Computer Graphics

Perhaps the most common use of creative coding is the Flash animation and Web design projects that youth are creating. In the u19 competition, these types of projects tend to be the most heavily represented. In 2003, Manuel Fallmann's space adventure *system interrupted* won an Honorary Mention for his creation using Flash animation [13]. Although this is an exemplary piece, it nevertheless, exemplifies what youth are capable of in this medium. Within primary and secondary education, animation using code or programming is typically found in *Adobe Photoshop* and *Hyperstudio* projects. One example is the work of Rebecca Sinker [30], which studies youth ages 4-7 from culturally diverse communities, whom create multimedia memoirs and interactive digital journeys to record their family history using *Hyperstudio*. Students interviewed family members, working on their interview/storytelling skills and allowing the students to explore their own history and value one another's. The author found that the project gave the children a sense of authorship as well as ownership. However, many of these projects do not necessarily involve the students in programming—the skill that we argue is fundamental for expression in a digital medium.

Building on this work, the remaining portion of this paper will delve more deeply into one design studio, providing an illustrative example of student work that creatively makes use of code—allowing the artist to make connections to art that would otherwise only be played or viewed. It is important to note that this is only one example of the diverse body of work that was created at this studio, as there are countless other works that typify videogame art, animation, and other art works made by students at this same site.

DIGITAL DESIGN STUDIO

Despite a broad definition of media arts, very few formal programs exist in the educational context that attempt to engage youth in this type of work. One exception is the media arts studio found at a community technology center in South Central Los Angeles, California situated at a storefront location in one of the city's poorest areas. As you enter the design studio, students can be found at the clusters of desktop computers grouped along the left wall. Young artists at this studio range in age from 8-18 years of age but most are between the ages of 10-14. At the center of the room sits a gigantic green table, serving as the common area for youth to work away from the computers with paper, pencils, markers, or electronic parts. In the back, sits a separate music studio equipped with a real piano and recording software for youth to create, record, and edit original music. All of the computers are networked to a central server, where youth have a personal folder that serves as a digital sketchbook or image archive, as well as a repository for finished work. Students work individually and in small groups, moving fluidly between the web and the unique software applications to create integrated projects. However, very few of these applications allow young designers to use and learn essential programming skills. *Scratch*, a media-rich programming environment, offers one strong exception [15].

Figure 1: Student discussing his *Scratch* project during a gallery display at the digital design studio.

SCRATCH PROGRAMMING ENVIRONMENT

Scratch differs from other novice-friendly visual programming environments [5] by using a more familiar building block command structure [16, 27], eliminating thorny debugging processes and the risk of syntax errors (see Figure 2). Moreover, designers don't need to learn or memorize bits of code to program when using *Scratch*. Instead, users have several pages of commands that they can drag to a central screen to control objects or characters they have created. Objects can be any imported graphic image, uniquely created or drawn, or chosen from a personal archive. Designers can create or incorporate existing sound files, video, and other input/output devices can also be integrated into new design projects — truly making them media-rich. We found that designers used this software to create videogames, art objects, and animated stories among other projects. Here we present an animated music video created by one artist.

Figure 2: Full screen shot of the networked, media-rich programming environment, *Scratch*.

BEGINNING CREATIVE CODINGS

The studio, as well as *Scratch*, was purposely designed with a constructionist theory of learning in mind, which poses that people learn best when they are active participants in design activities [10, 22]. The examples outlined here build on one of the main contributions of constructionism, appropriation, which posits that learners construct knowledge by making it their own. Resnick et al. adds to this discussion by pointing out that successful constructional design allows the creator to make both personal and epistemological connections as an integral part of the design process [26]. We argue that it's through these two types of connections that users make a more personally meaningful bond to art itself. Through these projects, students make connections to larger bodies of knowledge, better understanding artistic production and expression in a digital medium. Here we will examine how one young designer has appropriated the *Scratch* environment, particularly through the making of personal and epistemological connections. Over the past year, we have documented design activities at the digital design studio through extensive ethnographic field note taking and the collection of over 300 design projects [3, 18]. Young artists in this study worked on creating personally meaningful *Scratch* projects such as animated stories, Videogame art, and interactive or playable art using pop culture images and sounds pulled from the web or created using other software.

To better illustrate how programming can establish deeper connections to art, we'll more closely examine a single project created by a 13-year-old African-American artist, called Kaylee. Although she has over four years of experience at the studio, this is her first project using *Scratch* as an expressive medium. Choosing to create a "dance video" as her first design project, called "k2b," she spent six to seven hours in a period of a three weeks making design decisions and learning the basics of programming. To briefly summarize her project, when the user presses the start button, music begins to play and characters dance on screen. Intermittently, the background changes between eight different images, positioning the characters in a new ways within the space as they dance (see Figure 3).

Figure 3: *k2b* (2005). Partial screen shot, including display window and character library.

Personal Connections & Expressions

There were at least two ways in which Kaylee made a personal connection to the project. At the start of the project, Kaylee had an existing interest in pop music and chose to use Gwen Stefani's "Hollaback Girl" as the basis for her music video. As a result, Kaylee chose media and music that reflected her perception of pop culture and music videos, as well as known web resources. Secondly, Kaylee chose to insert both a picture of herself and her younger brother as two of the dancers in her music video, augmenting the personal expressive quality of this piece (see Figure 3). However, in the search for these images, Kaylee was very particular about the "look" that she was trying to achieve, stopping frequently to say things like, "no...I look too serious there. Besides my clothes just aren't right...it's just not cool." It was less important to her to insert an image of herself, as it was to find one that fitted her perception of how a music video should appear. Ultimately, she found a photo of herself that she liked and, standing up out of her seat, declared, "yeah, that's it...Let's put it in."

Epistemological Connections & Expressions

As we look more closely at the epistemological connections and expressions that Kaylee made during the design process, we find that there was an iterative cycle, starting with a design-oriented goal (i.e., how the characters should look and dance) and achieved by utilizing complex command structures. To illustrate this design cycle, we will use the distinctive dance patterns that Kaylee created. Each of the characters had a unique set of commands, creating a unique dance pattern, or "behavior." Kaylee uses programming and math concepts such as the x and y coordinate axes, the unit circle, and a repeat loop to create her desired effect (see Figure 4). For example, the ballerina moved as if she were dancing in a pirouette on her toes by flipping the image horizontally and moving it in small circular steps. This contrasts with the bananas that appeared to be moving in a diagonal motion across the screen (see figure 5).

In essence, Kaylee created a palate of movements and would borrow or group them depending on how she felt the image should behave. Kaylee programmed another character—a digital photo of her brother in a toy car--by selecting the scripts that she wanted from other characters and compositing them in order to imitate driving. In the last figure (Figure 6), we see an example of how Sprite15 was programmed by creating a unique composite of two separate program codes. Within the repeat loop, the first section of the code was pulled from the ballerina and the latter half was pulled from the bananas. The field researcher reports, "What was kind of neat about the commands that she serendipitously chose was that the image flipped horizontally and since the car was nearly symmetrical, it looked as if her brother was turning the wheel on the inside of the car and turning his head slightly." When Kaylee discovered and was happy with this effect, she finished programming this object and began working on something else. Without any prior programming experience, Kaylee was using looping constructs, repurposing scripts, and, moreover, coding creatively.

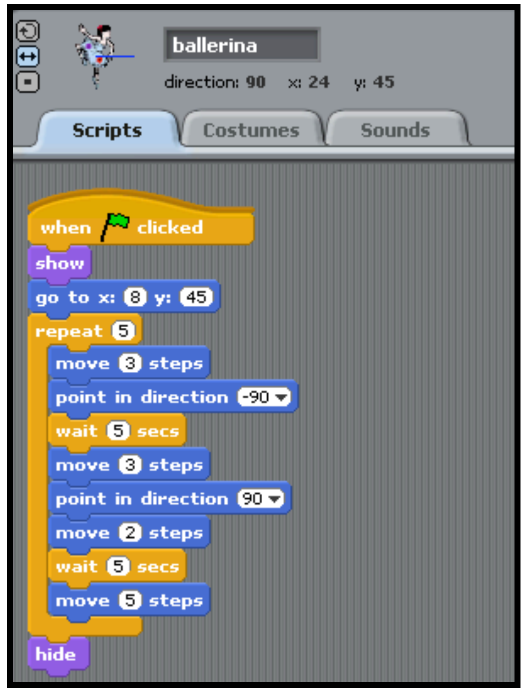

Figure 4: *k2b* (2005). Partial screen shot, including scripts or codes controlling the ballerina.

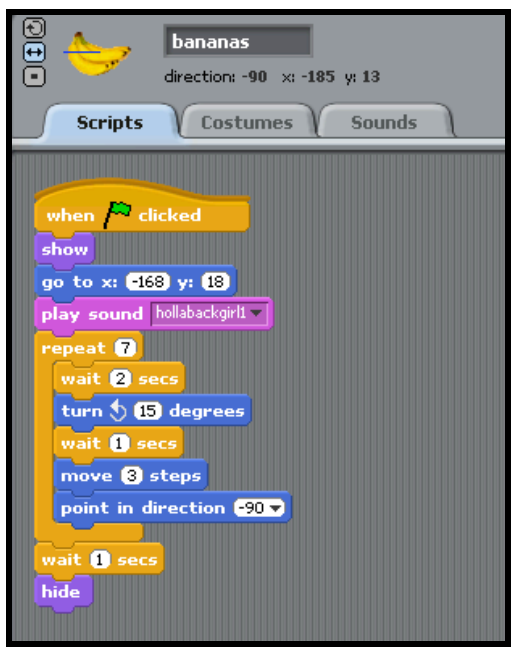

Figure 5: *k2b* (2005). Partial screen shot, including scripts or codes controlling the bananas.

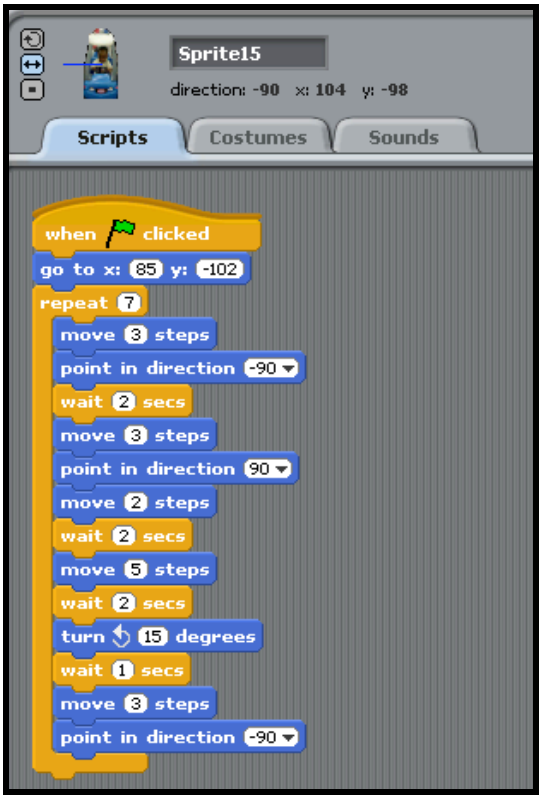

Figure 6: *k2b* (2005). Partial screen shot, including scripts or codes controlling the bananas.

DISCUSSION

We feel that young designers involved in *Scratch* make specific personal and epistemological connections to their work. It is through these connections that young designers, like Kaylee, establish a greater connection to the arts in general. In addition, to the competency that Kaylee already had in viewing and playing of digital medium, she developed through this project a greater understanding of how such mediums are created. This was evident through her utilization of programming concepts like looping constructs, conditionals, and assembling programs out of base components — concepts that are difficult for novice computer science majors [5, 32]. Moreover, as found in the example here, learning to creatively code does not necessarily involve an extensive and time-consuming introduction. It's important to note that our efforts in introducing creative coding into community technology centers are not geared towards replacing traditional arts or turning all youth into code-obsessed hackers. Rather, we argue that learning the language of creative coding is essential to expression in a digital medium—a medium that has an increasing importance for youth and society at large. In short, media arts expression requires a separate set of competencies and tools not usually cultivated in an educational context.

Some may question whether these creative codings fall into media arts. To answer this question, we turn to two recent papers, which discuss this issue more specifically within the context of video games. These papers point out the different directions into which video games become art — from the viewing and the playability of expressive adaptations of video games. Darshana Jayemanne, in her paper titled "Spielraum: Games, art and cyperspace," wrestled with the issue of whether videogames can be considered art [9]. Jayemanne situates video games as ready-made worlds that invite interaction and participation between the viewer (i.e., player) and the object (i.e., the game). One tension is that game worlds are both simultaneously hardscape and imagescape (according to Peter Lunenfeld's distinction) as they are embodied in certain hardware architectures but also in the mind of the player. In her mind, these connections provide the qualification to consider videogames as art. In other words, despite the mass production and wide availability of video games, in certain respects video games can be considered artistic because of the ways in which the viewer in engaged in the piece. In a similar manner, creative coding projects like the ones mentioned here are artistic objects for the viewer.

Accordingly, work like that of Tiffany Holmes explores videogames as playable art [8]. She examines how some video games adopt or incorporate the interface and interaction style of first generation video games, like Pong, and are exhibited at museum spaces. There are a range of video game adaptations for feminist and cultural purposes, demonstrating the variability of the genre and how it has been adopted for expressive means. Her paper focuses entirely on the game playing experience and all of the games discussed in the article were designed by others for the player or viewer to engage in certain ideas. Together, these two papers build an argument for valuing the artistic contribution in the viewing and playing of digital media. Our work adds another, equally important perspective -- that of creating your own video game or related artistic project. We add that it is not just in the viewing or playing but also in the constructive – or coding – experience that connections to art can be established. In other words, coding allows the creator to simultaneously appreciate and create art.

CONCLUSION

Undoubtedly, more work needs to be done in this area. Not only is there a limited amount of research in the field of primary and secondary media arts programs, an even smaller amount of work has been aimed at giving youth access and knowledge to programming skills. Without such competencies and skills, youth will not be able to manipulate digital mediums in an expressive manner. The potential of getting youth interested in technology has not been realized in computer science nor in arts classes, representing a missed opportunity for arts education especially since many involved in changing the absence of women and minorities in information technologies have argued for alternative approaches in computer classes to broaden participation [17]. Given this need to become fluent in both the arts and information technologies, it makes sense to offer interdisciplinary experiences for young children, similar to those presented by *Scratch*.

REFERENCES

[1] Abbott, C. (1999). Web publishing by young people. In J. Sefton-Green (Ed.), Young People, Creativity and New Technologies. London: Routledge.

[2] AEP. (2004). The Arts and Education: New Opportunities for Research. Washington, DC: Arts Education Partnership (AEP).

[3] Creswell, J. W. (2003). Research Design: Qualitative, Quantitative, and Mixed Methods. Thousand Oaks, CA: Sage Publications, Inc.

[4] Eisenberg, M., & Nishioka, A. (1996, April). *Polyhedral Sculpture: the Path from Computational Artifact to Real-World Mathematical Object.* Paper presented at the American Educational Research Association (AERA), New York, NY.

[5] Guzdial, M. (2003). Programming Environments for Novices.Unpublished manuscript, Atlanta, GA.

[6] Harel, I. (1990). *Children designers*. Norwood, NJ: Ablex.

[7] Harel, I. & Papert, S. (1991). Software design as a learning environment. *Interactive Learning Environments, 1*(1), 1-30.

[8] Holmes, T. (2003). *Arcade Classics Spawn Art? Current Trends in the Art Game Genre.* Paper presented at the MelbourneDAC2003, Melbourne, Australia.

[9] Jayemanne, D. (2003). *Spielraum: Games, Art, and Cyberspace.* Paper presented at the MelbourneDAC2003, Melbourne, Australia.

[10] Kafai, Y. B. (in press). Constructionism. In K. Sawyer (Ed.), Cambridge Handbook of the Learning Sciences. Cambridge, MA: Cambridge University Press.

[11] Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning*. Hillsdale, NJ: Lawrence Erlbaum Associates.

[12] Kafai, Y. B. (1998). 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). Boston, MA: MIT Press.

[13] Leopoldseder, H., & Schopf, C. (2003). *Prix Ars Electronica: CyberArts 2003*. Linz, Austria: Ars Electronica.

[14] Maeda, J. (1999). Design By Numbers. Cambridge: The MIT Press.

[15] Maeda, J. (2004). Creative Code. New York: Thames & Hudson Inc.

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

[17] Margolis, J., Fisher, A., & Miller, F. (2000). The Anatomy of Interest: Women in Undergraduate Computer Science. Women's Studies Quarterly, Special Issue on Women in Science, 104-126.

[18] Maxwell, J. A. (2005). Qualitative Research Design: An Interactive Approach (Vol. 41). Thousand Oaks, CA: Sage Publications, Inc.

[19] Mehta, S. (2005, July 28). Summer Camp for Kids, Ninjas. *Los Angeles Times*.

[20] Muchnic, S. (2005, April 24, 2005). Art on the move. Los Angeles Times, pp. E1, E34-36.

[21] Palumbo, D. J., & Calista, D. J. (1990). Opening up the Black Box: Implementation and the Policy Process. In D. J. Palumbo & D. J. Calista (Eds.), Implementation and the Policy Process (pp. 1-17). New York: Groenwood Press.

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

[23] Papert, S., & Resnick, M. (1993). Technological Fluency and the Representation of Knowledge, Proposal to the National Science Foundation: MIT Media Laboratory.

[24] Pinkett, R. D. (2000, April 24-28). *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.

[25] Poissant, L. (2005). New Media Dictionary [Website]. Groupe de Recherche en arts mediatiques (GRAM). Retrieved July 17, 2005, from the World Wide Web: http://mitpress.mit.edu/e-journals/LEA/NMD/nmdhome.html

[26] Resnick, M., Bruckman, A., & Martin, F. (1996). Pianos Not Stereos: Creating Computational Construction Kits. Interactions, 3(6).

[27] 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 submitted to National Science Foundation.

[28] Ryokai, K., Marti, S., & Ishii, H. (2005). Designing the World as your Palette. Paper presented at the CHI 2005, Portland, Oregon.

[29] Sefton-Green, J., & Buckingham, D. (1998). Digital Visions: Children's 'Creative' Uses of Multimedia Technologies. In J. Sefton-Green (Ed.), Digital Diversions: Youth Culture in the Age of Multimedia (pp. 62-83). London: UCL Press.

[30] Sinker, R. (1999). The Rosendale Odyssey: multimedia memoirs and digital journeys. In J. Sefton-Green (Ed.), Young People, Creativity and New Technologies. London: Routledge.

[31] Spohrer, J. C., & Soloway, E. (1985). Putting it all together is hard for novice programmers. Paper presented at the IEEE International Conference on Systems, Man, and Cybernetics, Tucson, AZ.

[32] Turbak, F., & Berg, R. (2002). Robotic Design Studio: Exploring the Big Ideas of Engineering In a Liberal Arts Environment. *Journal of Science Education and Technology, 11*(3), 237-253.

[33] Williams, S. (1999). Roath Village Web: the Marlborough Road Online School Scrapbook. In J. Sefton-Green (Ed.), Young People, Creativity, and New Technologies (pp. 70-82). London: Routledge.