Title
High tech programmers in low-income communities: Seeding reform in a community technology center

ISBN
978-1846289040

Authors
Peppler, Kylie
Chiu, Grace
Kafai, Yasmin

Publication Date
2007-06-21

Peer reviewed

sis on the role such access can have in socio-economic upliftment. This in turn requires a comprehensive approach addressing real social needs, including capacity of users as well as those who must implement the project on the ground, and the provision of relevant information in an understandable format. The key argument of this paper, emphasising the human elements of social inclusion and real access in bridging the digital divide, is most succinctly summarised by Bridges.org (2002:12), noting that "*[i]t's not about the technology, it's about the people*".

5. References

Benjamin P (2000) African Experiences with the Internet. J OnTheInternet, Oct 2000. http://www.isoc.org/oti/articles/1100/benjamin.html

Benjamin P (2001) The Gaseleka Telecentre, Northern Province, South Africa. in Colin Latchem and David Walker (eds). *Telecentres: Case Studies and Key Issues. Vancouver: The Commonwealth of Learning.* http://www.col.org/telecentres/Telecentres_complete.pdf

Bridges.org (2002) Spanning the Digital Divide. Understanding and tackling the issues. Bridges.org. Durbanville

Bridges.org. (2005) Comparison Study of Free/Open Source and Proprietary Softwae in an African Context. Implementation and Policy-Making to Optimise Public Access to ICT. Bridges.org. Durbanville

Infonomics South Africa (2003) Evaluation of the Smart Cape Access Pilot Project A City of Cape Town digital divide initiative. Directorate: Information Technology. City of Cape Town

Momsen N (2006) Personal communication 17 August 2006

Sooful N, Ismail M and Neville M (2002) Project In Progress Report #1: e Powering the People - The First Year of the Smart Cape Access Project October 2001 – September 2002. City of Cape Town IT Directorate: Smart City Initiative

Thomson G (Thomson Research Services) (2006) Directory of Public access points in the Western Cape. Centre for e-Innovation Provincial Government of the Western Cape

Trend D (ed) (2001) Reading Digital Culture. Blackwell Publishers. Oxford, UK; Massachusets, USA

Valentine S (2004) E-Powering the People: South Africa's Smart Cape Access Project. Council on Library and Information Resources. Washington, DC. USA

Warschauer M (2003) Technology and Social Inclusion: Rethinking the Digital Divide. The MIT Press; Cambridge, Massachusetts; London, England

Wright C (2002) Taking Stock and Looking Ahead: Digital Divide Assessment of the City of Cape Town. City of Cape Town: Economic Development & Tourism Directorate

High Tech Programmers in Low-Income Communities: Creating a Computer Culture in a Community Technology Center

Yasmin B. Kafai, Kylie A. Peppler and Grace M. Chiu

University of California, Los Angeles

1. Introduction

For the last twenty years, issues of the digital divide have driven efforts around the world to address the lack of access to computers and the Internet, pertinent and language appropriate content, and technical skills in low-income communities (Schuler & Day, 2004a and b). The title of our paper makes reference to a milestone publication (Schon, Sanyal, & Mitchell, 1998) that showcased some of the early work and thinking in this area. Schon, Sanyal and Mitchell's book edition included an article outlining the Computer Clubhouse, a type of community technology center model, which was developed to create opportunities for youth in low-income communities to become creators and designers of technologies by Resnick, Rusk, and Cooke (1998). The model has been very successful scaling up, with over 110 Computer Clubhouses now in existence worldwide.

Walk into any Computer Clubhouse and you are likely to see youth creating and manipulating graphics, animations, videos, music, and often integrating multiple media. The professional image-processing tool, *Adobe Photoshop*, is particularly popular. Indeed, a "Photoshop culture" has emerged at many Clubhouses, with youth proudly displaying their visual creations on bulletin boards (both physical and online), sharing Photoshop techniques and ideas with one another, and helping Clubhouse newcomers get started with the software. What you don't see very often, if at all, is a culture of programming that was originally part of the Computer Clubhouse vision to promote technological fluency – "the ability to reformulate knowledge, to express oneself creatively and appropriately, and to produce and generate information (rather than simply to comprehend it)" (National Research Council, 2000). Computer programming is integral knowledge across disciplines from the sciences to the arts, yet minorities and low-income students are notably absent in computer science-related fields. The Computer Clubhouse, therefore, potentially represents an important and alternative pathway towards technological fluency for marginalized youth.

In this paper, we will examine why programming, an aspect of technological fluency, despite all good intentions did not become part of the larger Computer Clubhouse culture. Thus, one goal of our investigation is to introduce the issue of change in community technology centers. While discussions about change are prominent in schools, they have not been part of the conversation around communities and technologies. The second goal is to introduce a successful example that illustrates our efforts to extend technological fluency activities in one Computer Clubhouse. We will present findings that examine our efforts from different technical and normative dimensions in line with Oakes' framework (1992; Oakes, Rogers, Lipton, & Morrell, 2002): (1) activities in the Computer Clubhouse before and after the introduction of a programming environment; (2) mentoring practices and technology conceptions of Clubhouse members; and (3) partnerships between community and local institutions. As we will argue, it was not any particular one, but the combination of all three of these factors that was responsible for seeding a programming culture with high tech designers in a low-income community. We intend to contribute to the larger debate on creating equitable technology participation in creative design across all communities.

2. Background

In 2000, the U.S. Department of Commerce found that Internet access was significantly dependent on household income and minority status. In the attempt to bridge this wide disparity of resources, more than 2,000 community technology centers (CTCs) have opened in the United States in the last decade, specifically to provide better access to technology in economically disadvantaged communities. Fortunately, recent legislation has reappropriated funding to further these efforts, thus establishing CTCs as a fixture in the landscape of technology access (Schon et al., 1998). But most CTCs support only the most basic computer activities (such as word processing, email, and internet browsing), so participants do not acquire the

type of fluency described in the NRC report. Similarly, many after-school centers (which, unlike CTCs, focus exclusively on youth) have begun to introduce computers, but they too tend to offer only introductory computer activities, along with educational games (Vasquez & Duran, 2000; Zhao, Mishra, & Girod, 2000). If members of low-income and minority communities gain access to new technologies, they are introduced in such a way that neglects to take the local context into consideration, and are often presented in such ways that reinforce rote learning activities rather than cognitively demanding activities (Warschauer, 2004).

A small subset of after-school centers and CTCs, such as those in the Computer Clubhouse network, explicitly focus on the development of technological fluency, moving beyond basic computer skills and helping youth learn to design, create, and invent with new technologies (Resnick et al., 1998). However, even at those centers focusing on fluency, youth rarely become engaged in computer programming. There is no "programming culture" analogous to the "*Photoshop* culture" which is so deeply embedded in most Clubhouses. On the one hand, traditional notions of programming see its value in fostering algorithmic and abstract thinking and problem solving skills (National Research Council, 2000). Yet, others might argue that these notions of programming are overly narrow, especially for CTC settings, and would be better placed in schools or technical colleges. Thus it shouldn't come as a surprise that programming did not take hold even in places like the Computer Clubhouse, which are predisposed by its vision and founders.

In understanding the challenges of bringing programming into Computer Clubhouses, we searched for frameworks that would help us understand the complexity of the situation. Some scholars have used models of technology diffusion to understand the successes and failures of how new technologies get adopted and integrated by users (Rogers, 1995). Within educational contexts, others have examined classroom practices of teachers to understand the lack of computer use in schools (Cuban, 1986, 2003). We turned to Oakes' framework because this model of reform recognizes that in order to expand access for low-income and minority students, change must occur in several dimensions. Oakes (1992) argues that equity-minded reform efforts must go beyond the *technical* (curricular and pedagogical) aspects and include changes in the *normative* (longstanding norms and conceptions) and *political* (institutional support within larger community) dimensions of an educational institution. This framework provides directives for those interested in bridging the missing gaps of the digital divide in non-school settings, particularly CTCs like the Computer Clubhouse.

Our program sought to address the technical and normative dimensions of Oakes' (1992) reform model and involved two critical levels of support (i.e., the addition of new media-rich programming software and the increased presence of mentors), as it was clear that we needed to tackle change in the computer culture on multiple fronts. Through our observations at the Computer Clubhouse, we found that youth have an interest in videogames, music videos, cartoon animations, and interactive, design-based art, which are a natural springboard into creating and programming. Thus we started with addressing the overly narrow notion of programming by focusing on the cultural artifacts that it could produce. This led us to recognize the benefits of programming as creative media production, which included a broader range of digital media texts, ranging from video games to "media mixes" of images, video and texts. With that in mind, we set out to create a media-rich programming environment, called *Scratch* (described later), that would provide youth with experiences creating and designing their own interfaces and applications (Resnick, Kafai, & Maeda, 2003). We argue that youth require technological fluency of how to construct new media in order to become critical consumers and producers. We think that such directions in community technology developments are particularly important for urban youth, who are often seen as pushing new adaptations and transformations of media, but are also perceived as standing on the sidelines of technology development and production.

We also realized that we needed to address support systems, in particular mentoring interactions in the Computer Clubhouse, to make learning and creative expressions the primary purpose of programming activities and not just the acquisition of technology skills. While mentors are often characterized as teachers and guides who provide information and advisement, and help identify mentee strengths and areas of improvement, there is in fact a rich literature that suggests mentors often assume additional roles in mentoring interactions. According to Flaxman (1992), mentors can take on various roles as teacher, advisor, supporter and companion. In our model, mentors who were introduced to the Computer Clubhouse were inexperienced programmers, providing an opportunity for mentees to feel more empowered in the learning process and even reinforce their knowledge in programming when called upon to act as a teacher to the mentors. There is little discussion that expands the continuum of mentoring roles from teachers to learners and thus would be more inclusive of a view that sees mentoring as a reciprocal rather than a hierarchical relationship. Such a view of mentoring counteracts the implicit deficit thinking present in mentoring approaches, which oftentimes assume a patronizing undertaking, where urban youth need to be rescued from their self-destructive behavior (Flaxman, 1992; Guetzloe, 1997).

The focus of our research, then, was to document, describe, and analyze the Computer Clubhouse at the different levels of technical and normative changes, highlighting key aspects for others planning to seed a computing culture. Applied to the Computer Clubhouse setting, the technical dimension involved the introduction of new software, the organization of new activities such as workshops and gallery presentations, and the addition of mentors that were inexperienced programmers. Our analyses were focused on the following questions: How widespread was the adoption of Scratch within the Computer Clubhouse? How well did mentors support Scratch activities? Normative dimensions address longstanding norms and conceptions about what programming is and are held by everyone at the site – including coordinators, staff, parents, mentors, and youth. Here, our analyses were focused on the following two questions: What are considered prototypical-programming projects? What types of beliefs do youth hold about their own ability to computer program? Finally, the political dimension involved introducing two partnering universities – U.C.L.A. and M.I.T. – to the local Clubhouse, where professors along with the computer programmer of Scratch personally visited the Computer Clubhouse on multiple occasions to showcase and share Scratch projects. While this political dimension was observed, it was not directly addressed in this round of analyses.

3. Context and Approach

The Computer Clubhouse where we conducted our research is located in South Los Angeles and serves primarily African-American and Latino/a youth, ages 8-18. Two full-time coordinators run the day-to-day operations and facilitate activities at the Computer Clubhouse, where adults play an important role in providing technical, intellectual, and emotional support for Clubhouse members. The volunteer mentors were college undergraduates, who were enrolled in an Education Minor course that focused on gender, culture and technology. As part the course requirement, these Undergraduates became mentors at the Computer Clubhouse, where they helped Clubhouse members in planning, developing and completing various design projects, while simultaneously learning various aspects of programming. We had a total of 38 Undergraduates enrolled over the course of four quarters (Kafai, Desai, Peppler, Chiu, & Moya, in press). The Undergraduates were all third and fourth year Liberal Arts Majors with little to no prior computer programming experience.

Over the course of the last three years, we conducted extensive field work and collected a total of 213 field ethnographic field notes at the Computer Clubhouse, capturing Clubhouse members' various design activities – before, during and after the intervention was introduced. In addition, we coded each sustained mentoring interaction for its content, distinguishing between design, games, web, homework, and social activities. We defined sustained mentoring as any activity where a mentor was interacting with a mentee over an extended period of time (a minimum of 15-20 minutes). In the field notes, either the length of the passage or the description of the amount of time that took place during the activity indicated this. Design activities involved the use of programming, 3D-animation, and graphic software such as *Kai's SuperGoo*, *Bryce5*, *Photoshop*, *KidPix*, game design programs such as *RPGmaker*, and music production software. Game activities included both games on the computer, such as *Roller Coaster (Tycoon)*, *School Tycoon*, video and online games, such as Whyville.net, as well as board and card games, foosball, and air hockey. Web activities involved web surfing with a mentee, while homework involved mentors helping mentees with their homework. We also created a "Personal" category to include all social activities and interactions between the mentor and mentee that establish and build upon the interpersonal relationship outside of the context of the other activities. Examples include a mentee or mentor sharing information about their lives to the other, advising, and/or listening. Four graduate students, in accordance with these three categories, coded all field notes independently. A subset consisting of 64 field notes was coded by all and revealed a reliability of 85-92%. The remaining field notes were then recoded independently.

Throughout the intervention, various design projects – including *Scratch* projects created by both members and Undergraduate mentors – were periodically collected, counted, and coded (Kafai, Peppler, Alavez, & Ruvalcaba, 2006). For the analysis, we took screenshots of program graphics and entered them into a spreadsheet along with short descriptions of content and functionality. Programs were then coded into four categories based on project type (animation, game, story, graphics, and other).

We also conducted interviews with members and undergraduate mentors, asking about their Clubhouse experience and the development of their programming skills (Peppler, in preparation). Each interview lasted about 15-20 minutes and questions included the following: What is computer programming to you? Does Scratch remind you of anything that you do at school or at home? And, how does Scratch differ from other computer software programs? All of the interviews were transcribed in preparation for later analyses. Researchers coded for themes rather than individual statements because these were group interviews and participants often expressed agreements with statements voiced by others; thus we did not expect every participant to repeat impressions.

4. Findings

In the following sections, we will illustrate the multiple levels of support needed for introducing programming into the Computer Clubhouse setting. We will start with an analysis of Clubhouse activities before and after the introduction of Scratch to illustrate the changes we witnessed on the technical level. Included in this documentation is a perspective on the range of mentoring activities that took place and the range of programming projects created at the Computer Clubhouse. From the normative level, we will review the interviews with youth for how they conceptualized their activities and showcase projects that became part of the programming culture in the Computer Clubhouse.

4.1 Technical Changes: Integrating Programming into the Clubhouse Design Portfolio

From our analyses of the field notes 2003–2004, we know that prior to the introduction of Scratch, programming activities did not occur in the Computer Clubhouse in South Los Angeles. Although *Microworlds* software, a visual Logo computer programming system, was available as part of the Computer Clubhouse's broad suite of software, neither adult coordinators nor members used it. While the Computer Clubhouse's most popular software titles enabled multiple media integration and manipulation, programming was considered a "stand alone" task and was therefore perceived as incompatible and irrelevant to popular design activities.

We developed Scratch, a programming environment with the ability to import and manipulate various media files that could be integrated with existing creative software. Arguably a full fledged programming language, Scratch (see Figure 1) vastly differs from other novice-friendly visual programming environments in that it utilizes a user-friendly building block command structure, eliminating debugging processes and the risk of syntax errors (Resnick et al. 2003; Maloney et al. 2004). Figure 1 is a screen shot of the Scratch user interface. The left most portion of the screen lists the palette of available commands. The middle panel lists the commands that the user has chosen to control the objects or sprites listed in the bottom right panel. The top right panel is the design screen.

Analysis of a large body of field notes has revealed that several pathways into the programming culture evolved over time at the Computer Clubhouse. The Clubhouse Coordinator introduced Scratch in Fall 2004. Although Scratch was loaded on several of the computers at this time, less than 10 members took advantage and created anything using the new software. Beginning in Winter 2005, a steady stream of undergraduate mentors joined the Clubhouse and the first explosion of Scratch activity was seen starting in early January 2005. Youth were encouraging one another to try out the program, and mentors worked with youth to create the first Scratch projects. Commonly, mentors would engage youth that had never worked in Scratch before by suggesting to import some of the pictures that they had stored in their folders on the Clubhouse server. At this point in time, the archive of projects represented a predominance of graphics-only projects that lacked any computer programming, which was due in part to the high volume of youth opening the program without any official orientation. Print outs of projects quickly began to cover the walls and Scratch slowly became the leading design activity within a few months of its introduction.

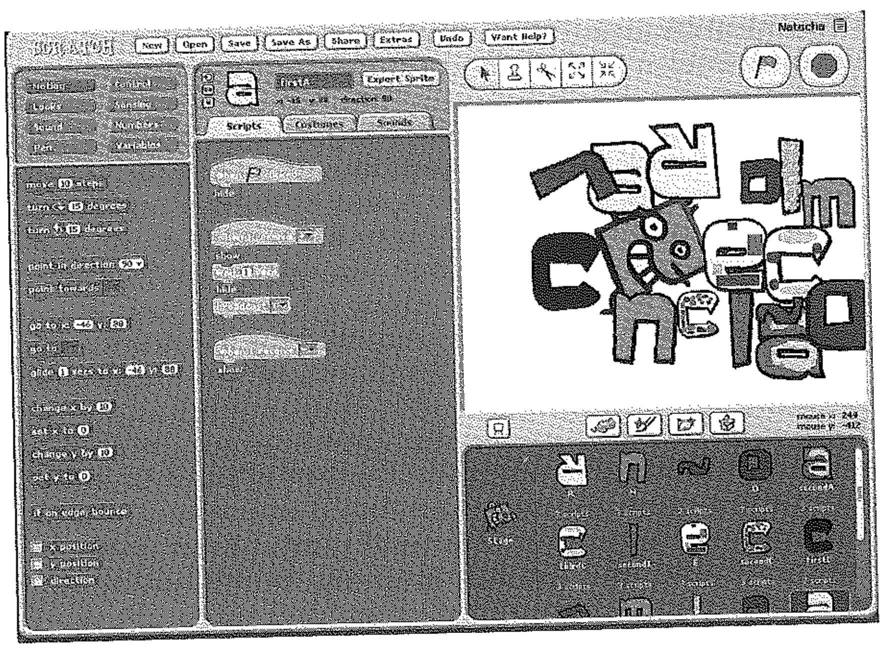


Fig. 1. Screenshot of the *Scratch* user interface.

In Winter 2006, there was an even greater interest in *Scratch* and some new things began happening within the computing culture. *Scratch* was used among the youth as a measure of membership in the local culture: new members wanting to establish clear membership in the community had to first create at least one Scratch project and store it for others to play on the central server. For the first time, more expert youth were seen mentoring other youth in Scratch. Scratch experts had a high-status position within the local culture and some youth had emerged as general experts that mentors, coordinators, and other youth consulted for help with Scratch, while other youth had specialized in certain genres or tricks within Scratch. In addition, groups of youth had begun working collaboratively together to create projects with a group name, such as "DGMM," for the Dang Good Money Makers. Youth also began to work independently of mentoring support, reflective of the high volume of projects beginning in June 2006, on complex projects and problems that they encountered in Scratch.

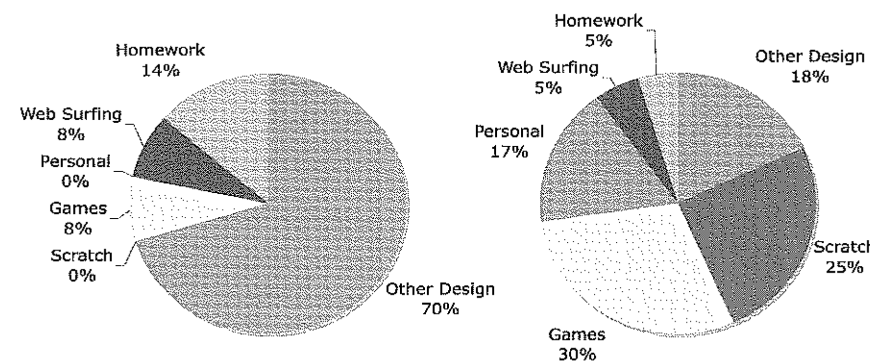
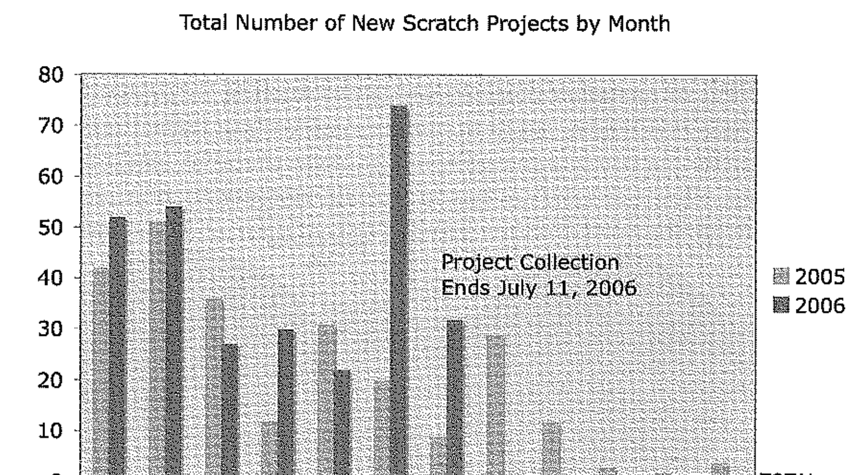


Fig. 2. Portfolio of Computer Clubhouse Activities (a) before [left] and (b) after the Scratch Introduction [right]

To further understand the impact of introducing new design software into the Clubhouse environment, we examined the field notes as records of sustained mentoring activities during winter and spring of 2004, 2005, and 2006. The "Clubhouse Design Portfolio" is therefore the average of sustained mentoring activities during these different time points. We interpret these findings as being a proxy for Clubhouse activities, of which we would otherwise have no other indication. Figures 2a and 2b summarize the portfolio of Clubhouse design activities before and after the introduction of Scratch. One finding is that programming activities increased as Scratch became embedded in the popular suite of design tools that Clubhouse members utilized on a daily basis.

4.2 Proliferation of Programming Activities in the Clubhouse

The design portfolio illustrates how programming had become part of the Computer Clubhouse activities. Over the course of the first 18 months we tracked Scratch development and collected all projects created by Clubhouse members (see Figure 3).



	Jan	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov	Dec	TOTAL
2005	42	51	36	12	31	20	9	29	12	3	2	4	250
2006	52	54	27	30	22	74	32						291

Fig. 3: *Scratch* Project Creation 2005-2006

There were several reasons for this approach, but important to the purposes of this paper is that it allowed us to peak at the computing culture when even mentors and researchers were not present at the site to answer questions about the sustainability of the programming culture in the absence of mentors. The number of new Scratch projects is also a good indication of general interest in computer programming over time. Figure 3 is a graph of the first 18 months of new Scratch projects arranged by the creation date and grouped by month. There are various peaks and valleys to the bar graph, indicating that the majority of interest in Scratch occurs

from January through August and there is less interest in the fall months between September and December. This is probably due to several reasons but can be somewhat explained by the presence of Undergraduate mentors from January thru March. Although further analyses are underway, it is difficult to explain the relative peaks and lows within this period (Peppler, in preparation). In addition, there is also a high volume of projects being created over the summer months (especially in June and July of 2006) in the absence of extensive mentoring support. We interpret this as an indication of the extended and prolonged impact that mentoring support can have on a programming culture beyond (or at least temporarily beyond) the weekly visits of the mentors.

The total number of Scratch projects paints a picture of an active computing culture, but what exactly are youth creating in Scratch? Because Scratch was designed to flexibly promote self-expression, youth have appropriated the software in a number of ways. Over the course of eighteen months, we collected over 500 programming projects created by members of the Clubhouse, some designed alone and others with mentors. We found that 44% of these projects fell into the category of animations with and without user manipulation, followed by 23% of graphics-only projects, and 15% of game projects focusing on fighting, sports and adventure; 14% of projects escaped a clear categorization because they did not provide enough detail.

We realize that this archival analysis of programming artifacts provides us only with a partial representation of a computer culture for multiple reasons: to begin with, our archive, while extensive, did not capture all Scratch programs designed but only those saved. The archive does not tell us what motivated Clubhouse members to create their programs, what they value in their designs, and how they compare them to their other design projects. We also could not address the equally important social and local influences at work that contributed to the design of the programs. Notwithstanding these limitations, the large number of Scratch programs provides a compelling example that members were active in creating numerous programs over an extended period of time and even without explicit curricular goals, grades, or instruction.

5.3 Social Support: Mentoring Activities in the Clubhouse

We also understood that access to relevant programming was only one of the technical aspects necessary to develop a culture of programming. Social support structures were equally important. Before Scratch was introduced, we observed that programming was a term that was rarely used in the Computer Clubhouse. Realizing that simply providing access to programming software would be insufficient, we created opportunities for Clubhouse members to interact with adult mentors who were learning to program as well (Kafai et al., in press). By introducing Undergraduate mentors and hosting Scratch workshops and showcasing events, we sought to establish new norms around programming. With daily support and exposure to Scratch, programming developed into a regular, socially accepted practice at the Computer Clubhouse. Throughout the showcasing events of Scratch projects, both mentor and member works were regarded as valuable.

It is also important to point out that Undergraduate mentors were not introduced to Clubhouse members as experts or teachers. In fact, the Undergraduate mentors were presented as fellow novices and collaborators, thus supporting one of the existing norms of the Computer Clubhouse learning model. As a result, many Clubhouse members emerged as resident experts of Scratch, thereby challenging the notion that programming is strictly for adults as demonstrated in the following field note excerpt:

> As we were both Scratch novice[s], Kathy went to ask an African American girl, whose name was Chenille, to help us ... she showed us *Scratch* skills such as how to use the glide and coordinates function ... When she gave us instructions, she looked very confident with her instructor-like tone.

While the traditional role of *teacher* surfaced—as some mentors attempted to dictate or control their situation as they would in a classroom—it became evident that Scratch provided additional opportunities for mentors to engage as learners. The role as *learner* occurred when the mentee led with an intention to teach, and there was evidence that the mentor was learning from the interaction. The mentee would be actively leading and explaining an activity with the mentor as exemplified in the following field note excerpt:

> After forming the basic animations and narration, we still had to figure out how to animate the soldier's beheading. Amanda became our best source as she came over and offered to help. She showed us some of her project so then we could understand how she switched head graphics. We learned from looking at Amanda's animation grid that in order to switch graphics, we had to apply a switch costumes function at the end of the previous animations for that costume...

Our analyses revealed that while the Undergraduate mentors sustained various mentoring interactions ranging from teaching to learning, the

prevalence of mentoring interactions that placed the mentor in the role of learner, observer or co-constructor – all roles which imply a more reciprocal and equitable relationship between mentor and mentees.

5.4 Idea Diffusion of Media-Rich Programming

The quantitative changes in design and mentoring activities were accompanied by qualitative changes in Scratch program genres and Clubhouse members' conceptions of programming. Some youth emerged that took on strong leadership roles. These leaders began to work with groups of 10-12 other youth to seemingly manufacture certain genres of projects; one example of this is the "Low-Rida" movement that began in January 2006. Within urban youth cultures there is a lot of interest in customizing cars. Television shows, like MTV's *Pimp My Ride,* have popularized this trend within mainstream American culture. Previously in the Clubhouse, a popular activity was to manipulate digital pictures of expensive cars, inserting a picture of yourself next to "your" car. Made popular by a young bi-racial African-American and Latino youth named Dwight, a culture of "Low Rida" interactive art projects has emerged (see Figure 4).

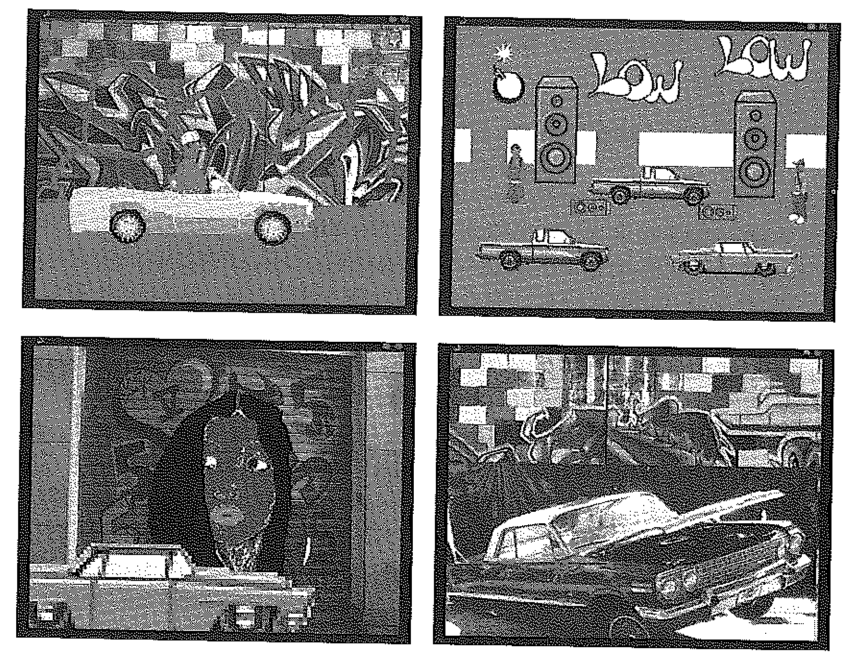


Figure 4: Screenshots of Dwight's "Low Rida" projects are in the upper and lower right corners. Other members of the Clubhouse created the two other

"Low Rida" Scratch projects. In the upper left, Dwight's brother customized his ride by painting it gold and drawing in gold hubcaps. In the lower left, an 8 year-old girl creates her own version of the Low Rida project, inserting a portrait that she created of herself using Painter7 software.

In one of Dwight's first projects, "Low Low," the viewer controls the hydraulics on two cars using arrow and letter keys. According to Dwight, the essential parts to his "Low Rida" project are the cars, the urban background, the graffiti-like lettering, and the speakers. It is important to note that the Low Rida movement emerged in the absence of mentoring support. The members conceptualized the idea and executed the projects almost entirely by themselves.

Several new Low Rida projects have emerged based on Dwight's earlier work, resulting in a widespread use of Scratch. In these projects, the creators have used Photoshop, Painter7, Image editors, and computer programming for creative production. By participating in the Low Rida movement, youth gain access to skills, empowering them to become designers of digital media. This is an important aspect of participation in an informal learning culture where contribution is valued. Projects like these eliminate barriers between high and low pop cultures (Sefton-Green & Reiss, 1999) by taking an urban youth culture theme and reinventing it using high status knowledge, such as software design.

5.5 Concepts of Media-Rich Programming

We also interviewed a large number of youth to better understand how they are making sense and appropriating Scratch. General conceptions of Scratch were overwhelmingly positive with youth proclaiming that it's their "favoritest thing ever." According to youth, Scratch is extremely flexible and has no or few limitations. Having trouble defining what Scratch was exactly most youth described it as "something that allows you to use your imagination" or as "a system that will allow you to do whatever you want." Most youth cited at least 4-5 different applications, which *Scratch* could be used for including making games, Low Ridas, comics, animations, music videos, short movies, and digital art. Although youth could recall a great deal about how to create projects in Scratch, citing specific commands and naming specific parts of the screen, most youth were unaware that creating in Scratch would be considered "computer programming." In fact, over half of the youth were unable to define computer programming.

If youth do not recognize that they are learning programming through Scratch, what do youth believe that they are gaining from their experience? Youth report a wide range of connections to traditional subject areas such as math, reading, science, and foreign language learning in addition to strong connections to the arts. The following excerpt is taken from an interview with Arnold, a 14 year-old African-American boy with limited Scratch experience, as he recounts his personal connection to Scratch through his experience as an actor. Notably, he cites how drama could be extended and reinforced in certain ways through Scratch.

Arnold: Well let me see…Well Scratch it really brings out my potential and it actually brings out my acting experience.

Interviewer: How so?

Arnold: Well when you take the microphone, you can create your own voice for your character. Like I love Arnold Schwarzenegger. Yeah it just really brings out your potential...Thinking of what you're doing with acting you can take it out of your mind and say like "in this picture we want to like do action stunts like flips and stuff", and if you're at school you're like doing Romeo and Juliet. You can make it more funny [in Scratch] by putting in some dragons. You can make a dragon go up to a castle and say "I came to rescue you." … Then you put them all in their places [in Scratch] and then once we do "Action!" We all come in with our parts.

Although we don't intend for all youth to become hacker-types as a result of their experience in Scratch, the involvement in the design process has awakened new possible career opportunities for some of the youth – notably the teenage boys. As one member puts it, "…it teaches how to play games and make games and it helps us figure out our future." This particular youth would now like to be a professional videogame designer, to attend college at M.I.T., and perhaps someday design a program like Scratch. He revels in his conversations with the professional programmers of Scratch and thoughtfully comes up with suggestions for how to further revise Scratch. It's clear that experiences like the ones at the Computer Clubhouse can have a considerable impact on the outlook and career aspirations of young people. Clearly, this is an area worthy of further exploration if we intend for youth to enter the computer science pipeline through informal avenues of education.

6. Discussion

A simple story of our efforts to seed a programming culture in the Computer Clubhouse would focus on the Scratch technology, alone. But as studies of technology change and innovation in organizations have shown, the introduction of new technologies is a much more complex enterprise. Researchers like Rogers (1995) have distinguished different phases from adoption which describes the selection of a technology to diffusion that refers to more wide-spread use and, finally, integration that illustrates acceptance in the community of practice. We are cognizant that our research partnership with the original founders of the Computer Clubhouse model gave us additional leverage in promoting new technology use not available to others. Our results indicate that Scratch indeed was integrated into the portfolio of design activities in this particular Computer Clubhouse, yet the true test of diffusion and integration will come as we are releasing the software to other Computer Clubhouses within the network.

The use of Oakes's reform model, previously only applied to schools, provided us with insights of the multiple dimensions at play in getting Scratch integrated into Computer Clubhouse activities. As part of our intervention, Scratch was never intended to be a shrink-wrapped package that was simply handed to members; rather, it was introduced in tandem with normative and political changes at the Computer Clubhouse. The introduction of both Scratch and undergraduate mentorship would not have been possible without a change in the political realm at the Computer Clubhouse. A formal partnership was forged between the university and the Computer Clubhouse's community host organization in order to gain support from the organization's infrastructure for these changes. By establishing goals, expectations, and communication protocols with the community organization, we were able to gain crucial buy-in on multiple levels, from the director to the coordinators. Through these various changes, a culture of programming began to emerge more in line with the initial vision of technology fluency aspect of the Computer Clubhouse model.

Meanwhile, we acknowledge the limitations to applying a school framework to a non-school reform model, which differs on many levels. For instance, normative and political structures in public schools are much more institutionalized than in most CTCs. Also, in our current era of increased accountability, pedagogy is strictly monitored in today's schools via national and state standards, while CTCs are usually left to their own devices to determine their respective learning approaches. These glaring differences may actually shed light on the unique advantages, challenges, and opportunities CTCs face in promoting technological fluency. Perhaps CTCs may serve as more fertile ground for promoting technological fluency than schools.

As illustrated in the examples of Clubhouse work, multiple aspects of media-rich production in informal settings provide youth access to technological fluency that empower them as designers in a setting where their contributions are valued. Our approach to technological fluency in the media rich Scratch software and in the programming projects in the Computer Clubhouse was grounded in youth practices. Previous discussions have cast this issue mostly in terms of access to digital equipment, talking about the digital divide when, in fact, the focus should be on the participation gap (Jenkins, 2006) that exists in today's society. It is here that our work with Scratch production gathers particular relevance in light of the inequitable access and participation of minority and low-income youth in digital technologies. Technological fluency is not just about knowing how to code, but also involves the personal expression as illustrated in the previous examples. These projects emphasize graphic, music, and video — media that have been found to be at the core of technology interests for youth. As we have argued, in the digital age, media literacy education needs to foster both critical understanding and creative productions of new media to encourage urban youth to be consumers, designers, and inventors with new technologies (Peppler & Kafai, in press). Places like the Computer Clubhouse can provide access to creative and critical media production skills such as programming in low-income communities and fill a gap not covered elsewhere.

7. Next Steps

As we move forward in introducing Scratch to other Computer Clubhouses in the world, we acknowledge that the structures we have put into place are unique to our location. Meanwhile, we contend that Scratch can flourish in other Computer Clubhouses as well, given that normative and political aspects are leveraged alongside this new programming environment. Currently, we are in the process of debuting Scratch to the entire network of Computer Clubhouses through three approaches: presenting workshops at training events for coordinators across the network; presenting workshops and showcase events for Clubhouse members across the network; and establishing a presence on the network's intranet project website. Through these efforts, we expect to develop new norms around programming and supportive political structures for sustained collaboration among Clubhouse members.

8. Acknowledgements

The work reported in this paper was supported by dissertation grant from the Spencer Foundation to the second author and by a grant of the National Science Foundation (NSF-0325828) to the first author in collaboration with Mitchel Resnick's research group at the MIT Media Lab.

9. References

Cuban L (1986) Teachers and machines. Teachers College Press, New York

Cuban L (2003) Underused and oversold and underused: Computers in the Classroom. Harvard University Press, Cambridge

Flaxman E (1992) The mentoring relationship in action. The Institute for Urban & Minority Education Briefs, 3. Teachers College, New York

Guetzloe E (1997) The power of positive relationships: Mentoring programs in the school and community. Preventing School Failure, 41: 100-105

Jenkins H (2006) Media literacy—Who needs it? Available online at: http://www.projectnml.org/yoyogi (accessed 9 August, 2006)

Kafai Y, Peppler K, Alavez M, Ruvalcaba O (2006) Seeds of a computer culture: An archival analysis of programming artifacts from a community technology center. In: Barab S, Hay K, Hickey D (eds) Proceedings of the Seventh International Conference of the Learning Sciences , pp 942-943

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

Maloney J, Burd L, Kafai Y, Rusk N, Silverman B, Resnick M (2004) Scratch: A sneak preview. Paper published in Creating, Connecting and Collaborating through Computing, Proceedings for the second International Conference of the Institute of Electrical and Electronics Engineers

National Academy of Engineering (2002) Technically speaking: Why all Americans need to know more about technology. National Academy Press, Washington DC

National Research Council (1999) Being fluent with information technology. A report of the Committee on Information Technology Literacy. National Academy Press, Washington DC

Oakes J (1992) Can tracking research inform practice? Technical, normative, and political considerations. Educational Researcher 21: 12-21

Oakes J, Rogers J, Lipton M, Morrell E (2002) The social construction of college access: Confronting the technical, cultural, and political barriers to low-income students of color. In: Tierney, WG and Haggedorn, LS (eds) Extending our reach: Strategies for increasing access to college. State University of New York Press, Albany

Peppler K (in preparation) Creative bytes: Literacy and learning in the media arts practices of urban youth. Unpublished dissertation. UCLA, Los Angeles

Peppler K, Kafai Y (in press) From SuperGoo to Scratch: Exploring creative digital media production in informal learning. Media, Learning, and Technology

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 for the National Science Foundation

Resnick M, Rusk N, Cooke S (1998) Computer Clubhouse: Technological fluency in the inner city. In: Schon, D, Sanyal, B, and Mitchell, W (eds) High technology and low-income communities. MIT Press, Cambridge

Rogers ME (1995) Diffusion of innovation, 4th edn. The Free Press, New York

Schon DA, Sanyal B, Mitchell WJ (1998) High technology and low-income communities: Prospects for the positive use of advanced information technology. The MIT Press, Cambridge

Schuler D, Day P (2004a) (eds) Shaping the network society: The new role of civil society in cyberspace. The MIT Press, Cambridge

Schuler D, Day P (2004b) (eds) Community practice in the network society: Local action / global interaction. Routledge, London

Sefton-Green J, Reiss V (1999) Multimedia literacies: developing the creative uses of new technology with young people. In: Sefton-Green J (ed) Young people, creativity and new technologies. Routledge, London

Vasquez OA, Duran R (2000) La Clase Magica & El Club Proteo: Multiple literacies in new community institutions. In: Gallegos M, Hollingsworth S (eds) What counts as literacy: Challenging the school standard (pp 173-189). Teacher's College Press, New York

Warschauer M (2004) Technology and social inclusion: Rethinking the digital divide. The MIT Press, Cambridge

Zhao Y, Mishra P, Girod M (2000) A clubhouse is a clubhouse is a clubhouse. Computers in Human Behavior 16: 287-300.

Charles Steinfield, Brian T. Pentland,
Mark Ackerman and Noshir Contractor (Eds)

Communities and Technologies 2007

Proceedings of the Third Communities and Technologies Conference, Michigan State University 2007

Springer

Charles Steinfield
Department of Telecommunication,
Information Studies and Media
Michigan State University
East Lansing, MI 48824, USA

Mark Ackerman
School of Information and Department
of Electrical Engineering and Computer
Science
University of Michigan
Ann Arbor, MI 48109, USA

Brian T. Pentland
Department of Accounting and
Information Systems
The Eli Broad College of Business
Michigan State University
East Lansing, MI 48824, USA

Noshir Contractor
Department of Speech Communication,
Department of Psychology,
and the Coordinated Science Laboratory
University of Illinois,
Urbana-Champaign
Urbana, IL 61801, USA

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2007927752

ISBN-13: 978-1-84628-904-0 e-ISBN-13: 978-1-84628-905-7 Printed on acid-free paper

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springer.com

C&T 2007 Sponsors

Department of Telecommunication, Information Studies, and Media, Michigan State University

College of Communication Arts and Sciences, Michigan State University

Quello Center for Telecommunications Management and Law, Michigan State University

Center for Leadership of the Digital Enterprise, Eli Broad College of Business, Michigan State University

Writing in Digital Environments Research Center, College of Arts and Letters, Michigan State University

Intellectual Property and Communications Law Program, College of Law, Michigan State University

Microsoft Research, Community Technologies Group

Each of the chapters was competitively selected from over 100 submitted works, using a rigorous peer-review process. We hope readers will agree that these papers reflect the high standards that have become associated with Communities and Technologies meetings.

We would like to thank a number of people who have helped to make this event and these proceedings possible. First, we are grateful to all the people who submitted papers and workshop proposals. We hope that future C&T meetings will experience similar strong interest from this emerging research community. Second, we deeply appreciate the work of our outstanding program committee. In addition to reviewing duties, program committee members helped to stimulate interest in the meeting and encourage their peers to view C&T as a premier outlet for community and technology-relevant research. Third, we wish to acknowledge the strong support we received from Michigan State University, which demonstrated its commitment to global scholarly exchange in hosting this meeting. Fourth, each of us acknowledges the support we received from our own institutions, Michigan State University, the University of Michigan, and the University of Illinois, for the time we were able to devote to organizing this meeting. Fifth, many people at MSU helped with the myriad details involved with organizing such an event and producing a volume such as this. In particular, Vanessa Pollok, Kim Croel, Jeff Siarto, and Ying-ju Lai, are due special thanks for their administrative, clerical, and web design work in support C&T 2007. Finally, we would like to acknowledge our representatives at Springer, Beverly Ford, Joanne Cooling, Helen Desmond, and Frank Ganz, for their timely production of a finely crafted volume.

Table of Contents