IT Fluency from a Project-Based Program for Middle School Students

Linda L. Werner
University of California, Santa Cruz

Jill Denner
ETR Associates, Scotts Valley CA

Shannon Campe
ETR Associates, Scotts Valley CA

Abstract

Information technology (IT) fluency is not well defined for K-12 students and this work contributes to filling this gap. In this paper we describe an after-school and summer program that aims to develop IT fluency by teaching computer game construction to middle school girls. Aspects of this program important for IT fluency acquisition are described. The game development software offered both challenges and opportunities for girls to develop IT fluency. We make suggestions for program modifications to achieve similar outcomes in a classroom setting. Our efforts to map project-based activities to IT fluency are important to the use of programs of this type in public classroom settings and for defining IT fluency in K-12.

Keywords

IT fluency, pair programming, middle school girls, project-based program, K-12 education.

1 Introduction

The US National Research Council (NRC) Committee on Information Technology Literacy (CITL) (1999) proposes information technology (IT) fluency as the minimum standard for all college graduates. The committee states that IT fluency is acquired over many years of exposure and describes elements that can be developed during the K-12 school years. According to the NRC CITL's report, IT fluency requires the acquisition of three kinds of knowledge through project-based work: contemporary IT skills, fundamental IT concepts, and intellectual capabilities.

Even though many colleges and universities have adopted IT fluency curricula, Snyder (2005) writes that IT education should be placed within K-12 curricula because "the general consensus is that the fluency content is better suited as an entrance requirement to college rather than an exit outcome" (p. 4). A recent NRC sponsored workshop committee issued a report suggesting the importance of assessing IT fluency in K-12 (Garmire & Pearson, 2006). Even though the International Technology Education Association and its Technology for all Americans Project has derived standards for technology education in the United States (2000), these standards have not been widely adopted. As of this time there are no clear standards for IT fluency in middle school, and thus no clear way to determine if children have been exposed to or acquired IT fluency. Therefore, little is known about how these types of knowledge develop during middle school. This paper moves the field forward in our efforts to map project-based activities to the development of IT fluency in this age group.

We examine IT fluency in the context of an after-school and summer program called Girls Creating Games (GCG). This program was designed to 1) increase girls' interest, abilities, and confidence with IT, and 2) help girls develop resiliency toward gender barriers (Denner, Werner, Bean, & Campe, 2005). GCG provided many opportunities for exposure to IT fluency knowledge. Descriptions of aspects of the program critical to IT fluency knowledge exposure are published elsewhere (Campe, Werner, & Denner, 2005; Werner, Campe, & Denner, 2005). In this paper we describe the areas of IT fluency knowledge identified by the NRC CITL, give an overview of the GCG program, and analyze the GCG program's activities in comparison to our interpretation of the NRC CITL report. We do this by mapping the GCG program's activities to aspects of IT fluency. One visible aspect of GCG was the use of Macromedia's Flash MX (Macromedia, n.d.) as the game development tool. However, nearly all of the IT fluency knowledge exposure derived from participation in this program could have been achieved by substituting any one of a number of different game development tools in place of Flash. We detail this argument and give a description of how using a project-based program such as GCG can achieve similar outcomes in a classroom setting.

2 IT Fluency and Middle School

The NRC CITL (1999) states that IT **fluency** is the minimum standard needed to prepare people for work and their personal lives in our information technology-rich society. It states that IT "literacy", or knowledge of the skills needed to use today's computer applications, is insufficient to prepare students for using future computer applications. Because information technology is rapidly changing, the skills necessary to use today's computer software will probably be different from the skills needed for using future computer software. Students will need more than just the skills they possess for today's popular software such as web browsers, instant messaging, and word processors. Instead they need to learn to adapt those skills to use as yet undeveloped software. The NRC CITL's argument for IT fluency starts with IT literacy and proceeds to the next step where the person learns to adapt as IT changes. It is important that IT fluency

> should not be regarded as an end state that is independent of domain, but rather as something that develops over a lifetime in particular domains of interest and that has a different character and tone depending on which domains are involved. Accordingly, the pedagogic goal is to provide students with a sufficiently complete foundation of the three types of knowledge that they can "learn the rest of it" on their own as the need arises throughout life (NRC CITL, 1999, p. 3).

This "sufficiently complete foundation" is a proposed standard for college graduates. The foundation consists of three aspects of knowledge: contemporary IT skills, fundamental IT concepts, and intellectual capabilities. The NRC CITL (1999) proposes the ten most important items for each of these three IT knowledge areas but does not prioritize within each list.

Contemporary IT skills are seen as today's IT literacy and include both contemporary hardware and software knowledge. These combined skills are commonly referred to as "knowing how to use a computer" and include:

- Set up a personal computer
- Use basic operating system features
- Use a word processor to create a text document
- Use a graphics and/or artwork package to create illustrations, slides, or other image-based expressions of ideas
- Connect a computer to a network
- Use the Internet to find information and resources
- Use a computer to communicate with others
- Use a spreadsheet to model simple processes or financial tables
- Use a database system to set up and access useful information
- Use instructional materials to learn how to use new applications or features

What was known as IT literacy 10 years ago is different from what is known as IT literacy today. For example, 10 years ago the use of the Internet would not have been included in the list. Based on an historical perspective, 10 years from now IT literacy will mean a different set of contemporary IT skills. Today, word processing skills are on the list, however, the need for these skills may disappear or dramatically change if there is widespread voice-activated access to computer resources.

The second IT fluency knowledge area, fundamental concepts, is the conceptual basis of IT and is independent of current technology and current application areas. This is the "book learning" part of IT fluency, but the NRC CITL is doubtful IT fluency fundamental concepts can be effectively learned solely by the use of textbooks. The NRC CITL believes new concepts may be developed upon which future technologies are based, however, these new concepts will add to the current conceptual basis of IT. The suggested set of IT concepts includes:

- Computers
- Information systems
- Networks
- Digital representation of information
- Information organization
- Modeling and abstraction
- Algorithmic thinking and programming
- Universality
- Limitations of information technology
- Societal impact of information and information technology

An understanding of the first of these items, computers, means one understands a computer program is composed of a sequence of steps, the program and program data are stored in memory, and the computer with associated peripheral devices interprets the program. Knowledge of information systems includes awareness that these systems include hardware, software, people, interfaces, databases, transactions, and issues of persistency of data, security, and privacy. One understands networks from an IT fluency perspective if one understands computers can be connected to networks and concepts of latency and bandwidth affect the transfer speed of information on these networks. The digital representation of information means all information stored and processed by computers is stored as a sequence of binary digits. Modeling and abstraction are crucial in order to understand technological representations of reality. When using computer systems, irrelevant details are abstracted away and the resulting simpler model of the problem domain is then manipulated. The concepts of iteration, sequence, and alternation form the basis for the functional composition of algorithms. For IT fluency, one needs to be able to program or take an algorithm and express it in a specific format readable by computers. Universality means any computer can execute any computational task. Because computers are designed and built by humans, computers are not magic and are subject to limitations of the knowledge of their creators. IT issues such as privacy, security, encryption, copyright, and free speech have an impact on all citizens; to participate in policy decisions in a democracy, citizens need to be informed. These concepts form the basis for IT and computer science and will continue to do so in the future.

The third IT fluency knowledge area, intellectual capabilities, refers to

> one's ability to apply information technology in complex and sustained situations and to understand the consequences of doing so. These capabilities transcend particular hardware or software applications (NRC CITL, 1999, p. 17).

This knowledge area states that people fluent in IT can:

- Engage in sustained reasoning
- Manage complexity
- Test a solution
- Manage problems in faulty solutions
- Organize and navigate information structures and evaluate information
- Collaborate
- Communicate to other audiences
- Expect the unexpected
- Anticipate changing technologies
- Think about information technology abstractly

Engagement in sustained reasoning involves problem definition and solution planning, executing, and evaluation over periods of weeks or months. A complex IT task can involve multiple steps where some of the steps may be interdependent, and the work may involve managing many different IT resources. Testing an IT solution has two components. First, if the proposed solution is developed, will it meet the user's needs? Second, the developed solution needs to be verified to determine if it is a correct implementation of the proposed solution. IT doesn't always operate as expected. A fluent person is able to determine the cause of the unexpected action and remove the cause of the problem and/or repair the faulty result. Organizing and navigating information structures and evaluating information involves online activities such as those using the World Wide Web and electronic databases and low-technology activities such as reading and understanding paper manuals. When a problem is divided between two or more people, collaboration is required to manage the division of the problem and reintegration of the solution pieces. IT fluency requires communication using IT and knowledge of the pros and cons of the different methods of communication. Also, information often needs to be communicated to others whose IT knowledge background is different. When IT is used in environments different than the intended environment (i.e., higher intensity of use, larger scale use, or purpose other than the intended), unexpected results can occur. Risk analysis is a way to deal with unexpected results. As new versions and forms of IT become available, IT fluent people need to evaluate the benefits and costs of the new technology. Abstract thinking about IT allows a person to use technological analogies to understand the use of IT in new environments or the use of completely new forms of IT. These intellectual capabilities are also relevant to many other domains such as science, engineering, business administration, and general education; however, the NRC CITL believes these intellectual capabilities do not easily transfer among different domains (NRC CITL, 1999).

The NRC CITL's implementation plan for IT fluency education focuses on the four-year college or university graduate's preparation but includes comments about K-12 applicability. Their focus is college-level because school districts do not usually enjoy the kind of curricula changing autonomy possessed by colleges and universities. Comments aimed at the K-12 arena are based on the NRC CITL's belief that IT fluency knowledge is acquired over many years in which a student "learns or covers the 'same' material repeatedly, but from increasingly sophisticated perspectives that build on what has previously been learned" (1999, p. 52). They state that K-12 programs that focus on specific IT skills in isolation of the use of the skills are inappropriate. Therefore, we believe a project-based approach for middle school students should significantly increase participants' level of IT fluency knowledge acquisition. While the NRC's report suggests IT fluency education should be part of a K-12 education, it does not contain guidelines. This paper contributes to the discussion necessary for development of guidelines for the K-12 level.

3 GCG Program Description

Girls Creating Games is a voluntary after-school and summer program for 6th, 7th, and 8th grade girls in a central California community. Four after-school and two summer implementations of this program were completed over the course of two years. In the after-school program, participants met on the campus of a public middle school twice a week for twelve weeks. Each of the 23 sessions was two hours long. The project-based program focused on the design and construction of interactive, narrative computer games using the Macromedia Flash MX development environment—software used to create web-based multimedia software.

Most computer classes and programs for middle and high school students focus on computer or information technology literacy-related tasks rather than on programming and design (Barker & Aspray, 2006, Commission on Technology Gender and Teacher Education of the American Association of University Women, 2000; Clewell & Campbell, 2002). The GCG program makes a unique contribution because it is designed to teach IT fluency rather than just IT literacy. The GCG program teaches IT fluency by having students create IT via project work. In GCG, knowledge of many of the NRC CITL's IT fluency skills, concepts, and capabilities was useful for creating IT technology.

In GCG, girls participated in activities on and off the computer. For example, the girls first played other computer games to generate ideas for their own designs. Then they participated in activities similar to those used by professional software engineers when designing a new computer game. They brainstormed story ideas, took notes on ideas, and wrote their game narratives on paper with multiple path options that lead to different outcomes.

In addition, the girls received instruction on using Flash. Instruction typically included an example of a Flash construct, a step-by-step demonstration of how that example was created, and a walk-through with each pair of girls creating a simple use of the new Flash construct. Interspersed within the Flash instruction sessions, the girls constructed their games. Later in the program they often modified previously designed parts of their games to incorporate newly learned Flash skills. Approximately half of the 46 hours of the GCG program time focused on learning the Flash development environment and on using Flash to design, program, and debug games.

Reports about the successful use of pair programming (see Williams & Kessler, 2000) in universities motivated our use of pair programming for the GCG program. McDowell, Werner, Bullock, and Fernald (2002) found that university students who pair programmed had greater confidence in solutions and enjoyed completing their assignments more than students who worked alone. Partners who pair programmed in entry-level courses performed similarly on tests taken independently, and pair programming reduced the gender gap for those who pursue computer science related university degrees (Werner, Hanks, & McDowell, 2005). The GCG program participants worked as pair programmers through all stages of game development. Additionally, the participants worked in these pairs for other collaborative program exercises. Just as in industry and university settings, the girls worked together side-by-side at a single computer and gave each other support. One girl was the driver who operated the keyboard and mouse, and the other was the navigator who provided guidance by negotiating decisions with her driver and monitoring the actions of the driver looking for potential problems. Partners switched roles frequently; an average driving session lasted 15 minutes.

Several new program activities were created to support the pairs. For example, effective and ineffective pair programming behaviors were identified for the target age/gender group using Williams and Kessler's (2000) paper as a starting point and following with observations both from our early implementations of the GCG program and of university students pair programming,. We built role-modeling scripts to demonstrate effective and ineffective pair programming (Werner, Denner, & Bean, 2004). From demonstrations, participants identified rules and placed them on a poster displayed in the GCG computer laboratory. Pair programmers of the week were recognized by the program leaders. Most of the participants worked with the same partner the entire time (85%). The remaining participants either worked alone (6%) or started with a partner and finished alone (9%).

The remaining half of the GCG program time focused on self-identity activities including career exploration and countering stereotypes, group-building activities, and pair-building activities. An example of an activity designed to both enhance self-identity and build the group is the Affirmation Activity. Girls wrote an affirmation or positive statement about some technology-related item they saw in someone else's work and these statements were shared and a few were read out loud by the receivers. These activities made the girls feel supported and helped build a community-of-learners, (Wenger, 1998) instead of solitary, competitive individuals.

4 Mapping Approaches

4.1 Program Examples

In our discussion of the GCG program activities and the mapping of these activities to IT fluency area items, we include data and program examples from three after-school implementation cycles in which a total of 33 choose-your-own-adventure games were completed. We include only the third (with 22 girls), fourth (with 23 girls), and sixth (with 19 girls) cycles because these had more consistent attendance, program activities, and learning environments. Ninety-eight percent of the 64 participants attended 70% or more of these sessions. This participation rate excludes girls who dropped out of the program. We also describe data from electronic notebook entries by participants of all six cycles in which a total of 126 girls participated. We use notebook entries in our discussions of how to modify the program to support IT fluency.

4.2 Details of the Mapping Approaches

As stated earlier, one of the GCG program goals is to increase girls' abilities in IT. Since increasing abilities in IT requires repeated exposure to the knowledge areas (NRC CITL, 1999); we developed two approaches to map program activities to exposure to IT fluency knowledge. These approaches are described in two earlier papers (Campe, Werner, & Denner, 2005; Werner, Campe, & Denner, 2005) and the details are repeated here to support the argument that much of the exposure is independent of the use of Macromedia's Flash as the game development software tool. For the first approach, we studied the computer games created by the program participants. We mapped many of the Macromedia's Flash features that participants integrated into their computer games to IT fluency knowledge area items. For example, the use of Flash's ActionScripting was mapped to the IT fluency fundamental concept of "algorithmic thinking and programming". When participants practice pair programming as they design and create their game, both members gain similar IT fluency exposure. We base this belief on research at the university level that found both members of a pair programming partnership benefited from the practice (Werner, Hanks, & McDowell, 2005). We include in the mapping a percentage of GCG program participants whose games demonstrate that IT fluency area item.

Our criteria for fluency is that if one of the Flash features is used in a game three or more times, the students who created that game have achieved a minimum state of competency in the corresponding IT fluency knowledge area item. We use a threshold level of three because the first time a pair of students use a specific Flash feature in their game, the instructor could have walked them through the steps; the second time it is used suggests the pair may have understood its use without the assistance of an instructor; the third time we assume they did not need the assistance of an instructor and at that point have repeated it a sufficient number of times to show a minimum level of competency in how to use the feature. We used this criteria to identify whether the pair has demonstrated minimum competency in the corresponding contemporary IT skill, fundamental IT concept, or intellectual capability (Campe, Werner, & Denner, 2005).

Next, we mapped specific GCG program activities to IT fluency knowledge areas. These activities cannot be quantified like the computer game features described above and differences exist in the extent to which the knowledge area items are addressed by GCG program activities. Some are fairly completely addressed, while others are addressed much less obviously and to a lesser degree of completeness. Additional details regarding the degree an area was addressed or the strength or weakness of certain claims including specific percentages for areas that are Flash-related and exhibited in the games can be found in (Campe, Werner, & Denner, 2005). These differences in exposure to IT fluency are described in the next section.

5 Results of the Mapping

GCG program activities are mapped to 22 IT fluency knowledge area items. Nine of these 22 items are labeled with an asterisk meaning students were exposed to these aspects of IT fluency only while working with Flash. This labeling is important for our discussion in Section 6.

5.1 Contemporary IT Skills or IT "Literacy"

We determined the girls were exposed to seven of the ten IT skills elements of IT fluency:

- Use basic operating system features
- Use a word processor to create a text document *
- Use a graphics and/or artwork package to create illustrations, slides, or other image-based expressions of ideas *
- Use the Internet to find information and resources
- Use a computer to communicate with others
- Use a database system to set up and access useful information *
- Use instructional materials to learn how to use new applications or features *

While students did not directly use word processing or database applications, our analysis suggest the use of Flash provides quite similar experience. Flash is designed with a full-featured artwork package of drawing and text tools for manipulation of selected screen elements. Flash's text processing tools are similar to a small subset of those seen with word processing systems and require similar skills to use. Users of Flash can control the look of their text by manipulating elements such as text type, width and height, character spacing, font, color, etc. Flash has built-in libraries which provide features which are similar to some found in simple database systems. Each object in a game such as a simple graphic or a much more complicated movie segment, called a symbol in Flash, is stored in either the built-in library or a user-defined library. When multiple instances of a symbol are used in a game, less computer memory is required because the symbols are each stored only once in the library. Students create folders in their user-defined libraries for the storage of similar symbols and these symbols use counts, which creates a hierarchical storage structure. As students become comfortable with Flash, their use and understanding of the library structure increases because it is believed to be the "single most important mechanism in Flash" (Blake, 2002, p. 43). All of the programming pairs used the built-in and/or user-defined libraries. Also, 64% of the games contained images and sound that were downloaded from the Internet after locating using a search engine.

In addition to game design, there were other elements of the GCG program that exposed participants to contemporary IT skills. These skills included the use of basic operating system features, use of the computer to communicate with others, and the use of instructional materials. For example, participants logged onto a secure server and invoked applications during every computer laboratory session. They used basic operating system features to do this. Participants used the computer to communicate in over two-thirds of the sessions when they responded to questions about the program's activities using an electronic notebook application. Lastly, students used instructional materials such as the instruction diagrams we referred to as "flowcharts." These flowcharts had step-by-step instructions for each Flash skill taught and were part of the documentation all pair programming navigators used during Flash instruction and game creation sessions.

5.2 Intellectual Capabilities

With respect to the intellectual capabilities component of IT fluency, we found that our GCG program provided activities addressing eight of the ten suggested elements (again, the Flash-based experiences are marked with an asterisk):

- Engage in sustained reasoning *
- Manage complexity *
- Test a solution *
- Manage problems in faulty solutions *
- Organize and navigate information structures and evaluate information
- Collaborate
- Communicate to other audiences
- Expect the unexpected

Because all of the program participants successfully created games using Flash, we believe they all engaged in sustained reasoning and managing complexity. In general, sustained reasoning involves more than a one-time problem solving event. Game design involves all stages of problem solving, including problem definition, clarification, planning, designing, executing, and evaluation and occurs over a period of days or weeks. The girls in GCG worked for twelve weeks designing and creating their games and had many opportunities to manage complexity. For example, the creation of a choose-your-own-adventure type of game requires management of multiple story paths. Each story path needs appropriate text to lead the player of the game along a specific path. We looked at the organization of the story paths to determine if each game met our minimum standards regarding complexity of story and number of endings. All but one game was created with the minimum required number of endings and different scenes. Many games had additional story paths and scenes. The one game that did not meet the minimum requirements was created by two girls who had been re-paired several weeks into the program and thus had less time to work on their shared game.

GCG program activities map directly to several intellectual capabilities, such as managing problems in faulty solutions, testing solutions, learning to expect the unexpected, navigating information structures and assessing the quality of the information, collaborating, and communicating to other audiences. For example, students managed problems in faulty solutions when debugging their Flash games. The debugging activity required each pair to identify and record bugs in another pair's game, review their own game's list of bugs, and decide how to fix the bugs. Sometimes the girls used the flowcharts to retrace steps (another example of organizing and navigating information structures and evaluating information). While programming and debugging their games, the girls also tested solutions involving animations, buttons, and the movie files to see how their games would play and made necessary modifications based on the tests. Additionally, because the program leaders had limited Flash expertise, the participants had ample opportunities to practice testing and debugging. The girls also had to learn to expect the unexpected. For example, they had first-hand experience when computers crashed and work was lost, or when one partner exited the Flash development environment without saving their most recent work.

The use of flowcharts throughout instruction and practice required girls to navigate information structures and assess the quality of information. Throughout game programming, they had to identify the correct flowchart and whether it would assist them in the problem they were trying to solve. The girls also had the opportunity to search the Internet (navigating an information structure) for image and sound files for possible use in their games.

The program offered opportunities to develop the intellectual capabilities to collaborate using IT and communicate to other audiences. As described earlier, pair-programming partnerships were a strong focus of the program. The girls used pair programming during game design and construction; however, they also participated in many other collaborative activities such as: Telephone Architects, Pattern Blocks, and Minefield. For example, in Telephone Architects, one partner, the architect, constructs a structure with plastic building pieces that their partner, the builder, cannot see. After construction is complete, the architect verbally describes how to build the structure to the builder who has an identical set of building pieces. The architect and builder work on terminology and the sequence of steps to successfully complete a replica of the original structure. Communication is critical for successful completion of this type of activity. All of these activities contributed to more peer support for the programming partnerships. Additionally, working in a pair means that communication is essential for problem solving.

The intellectual capability of communication to other audiences was also practiced. The girls had to prepare a presentation about their game and the software to people not familiar with the details of the GCG program or Flash. On the last day of the program, each pair gave a presentation and demonstration of one aspect of their game to a gathering of their families and friends.

5.3 Fundamental IT Concepts

As described by the NRC CITL, fundamental IT concepts

> touch on ideas of computation, communication, and information that are deep and intellectually challenging. Although any of the topics could be the basis of years of graduate study for a specialist, the basic ideas are straightforward and accessible, having been regularly taught to non-specialists for years (NRC CITL, 1999, p. 28).

In this area, our analysis suggests that seven of the ten elements were addressed by our GCG program (one, marked with the asterisk, arose from the use of Flash):

- Computers
- Information systems
- Digital representation of information
- Information organization
- Modeling and abstraction
- Algorithmic thinking and programming *
- Limitations of information technology

It is important to expose students to these concepts often so that IT fluency can be achieved, and as IT evolves students can continue to acquire skills needed to create and use new technologies. We examined the completed games, and concluded that the girls demonstrated the fundamental IT concepts of information organization and algorithmic thinking and programming. All the games contained uses of ActionScripting—the programming feature of Flash. The ActionScript commands of stop and goto were used in all of the games. Some of the pairs added ActionScript commands for manipulating input and output fields so user-provided information would appear later in the text of the story. Additionally, team-building activities (described in Section 5.2) enforced precise language and sequence of instructions and prepared the girls for actual programming activities. The girls also demonstrated knowledge of the concept of information organization, sometimes referred to as information literacy (NRC CITL, 1999). Sixty-four percent of the games used clipart, photos and/or sound the girls found on the Internet. Participants used the Google search engine to locate websites or banks of images and chose graphics to download for their games. Another example of exposure to the fundamental concept of information organization was exhibited by the extensive use of Flash's layering construct in 61% of the games. A program created with Flash can have many layers and each of these layers can have different graphics, sound, and ActionScript actions. Layers are necessary to organize the different pieces of the game, so if a pair needed to change a part of the look-and-feel of their game, they usually only had to manipulate a single layer.

In the NRC CITL's report (1999), the fundamental IT concept of computers includes general knowledge such as the concept that computer programs are composed of precise sequences of instructions. Many GCG activities, such as the team-building activities described in Section 5.2, required the use of language and steps with a precision similar to that required to program a computer. The extensive use of flowcharts also enhanced the girls' concepts of computers and offered a visual description of how they are programmed as a sequence of precise steps. The use of Flash's ActionScript commands, a form of programming, is another example of this concept.

Participants had many opportunities for exposure to the fundamental IT concept of information systems. For example, information systems were part of every day of the GCG program that had a computer laboratory component. The participants gained access to GCG files using a password-protected interface and learned that files were stored in persistent storage because they were still accessible weeks after creation. The girls had exposure to various types of user interfaces including the operating system, network search engines, the Flash program, and many online computer games. Each of these is an example of one aspect of an information system. Each exposure to a different information system helps with adaptation to each new one.

The IT concept of digital representation of information was experienced when the program participants manipulated graphical, textual, and audio symbols both during activities involving search engines and within Flash. Most of the girls searched for and downloaded images from the Internet for their games. They frequently enlarged the digital images. Enlarging these bitmapped graphics often resulted in distortion, which exposed the girls to the limitations of information technology. Many more opportunities of manipulation of digital representations are needed before these two IT concepts of digital representation of information and limitations of information technology would be internalized.

Participants were also exposed to activities to learn modeling and abstraction. Prior to using Flash to create games, the girls designed models of their games on paper with an instructional scaffolding tool called a "story path diagram" and then used Flash to actualize their designs. They used story path diagrams to specify only the control paths and major plot ideas instead of all of the details of their games.

6 Applying the Program Activities to a Classroom Setting.

Most of the activities in the GCG program map directly onto critical aspects of IT fluency. These activities would be relatively easy to implement in a classroom and other settings, except the use of Flash as the game development software. Flash is a professional development tool for interactive media and the readers are freely available, however, a license is required to use the development tool. We found that the Flash development tool was not kid-friendly and we hypothesize that most middle school teachers are not trained in its use. Therefore, it is important to understand whether Flash was an important part of the IT fluency acquisition aspect of our program.

To this end, we asked the girls to respond to the following questions in their electronic notebooks: What was your favorite part about making a game in Flash? What did YOU do best? The 57 responses were coded according to the 30 IT fluency knowledge area items. Each response was associated with zero or more IT fluency knowledge area items, resulting in 103 encodings or entries. Forty-five percent, or 46, of the entries identified the use of graphics as their favorite part of using Flash. Many girls stated they liked to draw, but in only ten of these 46 entries was animation given as their favorite part. Flash, being a commercially available multimedia-authoring tool, has features for two kinds of animation not usually found in kid-friendly game creation software: frame-by-frame animation and tweening[1]. The animation features of Flash were not as important to the GCG participants as was

[1] "Short for *in-betweening,* the process of generating intermediate frames between two images to give the appearance that the first image evolves smoothly into the second image." From http://www.webopedia.com/ TERM/T/tweening.html accessed on December 19, 2005.

the ability to create drawings. Twenty-two percent of the entries contained references to the creation of a project as a whole (encoded as sustained reasoning) whereas only four percent contained references to algorithmic thinking and programming. There were 10 or less entries containing references for any of the other IT fluency knowledge area items.

To what extent was Flash a critical part of IT fluency exposure? The GCG program provided exposure to 22 of the 30 IT fluency knowledge area items. In nine of these items, this exposure was a result of using Flash:

- Using a word processor to create a text document,
- Using a graphics and/or artwork package to create illustrations, slides, or other image-based expressions of ideas,
- Using a database system to set up and access useful information,
- Using instructional materials to learn how to use new applications or features,
- Engage in sustained reasoning,
- Manage complexity,
- Test a solution,
- Manage problems in faulty solutions, and
- Algorithmic thinking and programming

Even though exposure to each of the above items was solely through the use of Flash within the GCG program, we conclude that Flash itself was not required. Many kid-friendly game creation tools come complete with graphics packages and programming and provide features to import graphics from the Internet. Even Microsoft Word or PowerPoint can be used to create hypertext documents that are the basis for many choose-your-own-adventure type games using graphics downloaded from the Internet or drawn using embedded graphics packages. Using Word or PowerPoint would not give students exposure to programming, however, the use of any of these other game creation tools or Word or PowerPoint for the project work can offer exposure to all of the other items related to the design and development, test, and debug of a computer game: sustained reasoning, management of complexity, test a solution, and manage problems in faulty solutions. This leaves only one item: use of a database system to set up and access useful information. Recall from our discussion in Section 5.1, our use of Flash gave the participants exposure to very simple database system features. Therefore, almost the same level of IT fluency knowledge exposure can likely be achieved with the use of a simpler, more kid-friendly development tool. Without the need for exposure to a database system, a classroom teacher can pick from a number of kid-friendly game development tools. Or they can solicit help from a parent volunteer familiar with game development tools. For example, Stagecast Creator (http://www.stagecast.com/) is an award winning point-and-click programming tool that allows kids to create games and other animations using a programming-by-example technique. It has facilities for drawing your own graphics or importing them. Stagecast Creator, Hands, and Visual Basic were used with 10 twelve year olds to create games and animations. Stagecast Creator was reported as enjoyable and usable by children and it was found to be the easiest among these three to learn (Lin, Yen, Yang, & Chen, 2005). Toontalk is another kid-friendly software tool for creating games and other software applications (Kahn, 1999). Alice and Virtual Family are two other kid-friendly programming tools; however, they have been introduced to middle school girls with limited success: technical as well as other problems were identified with their use (Fiebrink & Alcott, 2003). Kelleher (in press, 2007) is researching changes to Alice to improve its usability for middle school girls. Even Microsoft's Word or PowerPoint can be used to provide exposure to most of the above nine IT fluency knowledge items. Both Microsoft Word and Powerpoint are readily accessible and require minimal teacher training or instructional support. Many middle schools already provide students with instruction on how to use PowerPoint and Word for reports. Each of these software tools has features for creating hypertext documents. A student would not be programming if they were using Microsoft Word or PowerPoint, however, they would be exposed to the IT fluency concept of algorithmic thinking while participating in the other GCG program activities, namely the sustained reasoning provided with the project work, and with this exposure, they would be well prepared to program during their next exposure to IT fluency.

7 Conclusions and Future Work

This paper contributes to efforts to define standards and assess IT fluency in K-12. In this paper, we mapped project-based activities and the products of those activities (games) to IT fluency standards for the K-12 arena.. Although IT skills (IT "literacy") are most likely to be taught in classroom settings, we have shown the GCG program to be a unique program for middle school girls that encourages exposure to more than just contemporary IT skills—it also provides exposure to intellectual capabilities and fundamental IT concepts. One major challenge to the adoption of the GCG program in a classroom setting is most teachers lack expertise in using the Flash development tool. We discussed how any one of a number of other kid-friendly software tools could be used in place of Flash and describe how that choice would provide for similar levels of IT fluency knowledge exposure.

Although the findings presented in this paper advance the study of IT fluency standards, they are based on a small number of sessions of the GCG program, and may not be generalizable to similar programs. Next steps could include using these mapping techniques with different programs for both girls and boys and using more kid-friendly software tools. It would also be of interest to investigate methods of how to more accurately map classroom activities to IT fluency for middle school students.

8 Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0217221. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We wish to thank all of the participants, program leaders, teaching assistants, and school collaborators involved with the Girls Creating Games program, as well as Steve Bean, the program developer. The website for the GCG program is http://programservices.etr.org/gcgweb/. We want to thank Philip East and an anonymous reviewer for their help with the editing of this paper.

References

Barker, L.J & Aspray, W. (2006). The state of research on girls and IT. In J.M. Cohoon & W. Aspray (Eds.), *Women and Information Technology: Research on Underrepresentation* (pp. 3-54). Cambridge: MIT Press.

Blake, B. (2002). *How to do Everything with Macromedia Flash MX*. Berkeley, CA: McGraw-Hill/Osborne.

Campe, S., Werner, L.L. & Denner, J. (2005). Information technology fluency for middle school girls. *IASTED International Conference Computers and Advanced Technology in Education*.

Clewell, B.C. & Campbell, P.B. (2002). Taking stock: where we've been, where we are, where we're going. *Journal of Women and Minorities in Science and Engineering*, *8* (3,4).

Commission on Technology Gender and Teacher Education of the American Association of University Women (2000). Tech-Savvy: educating girls in the new computer age, Washington, D.C.

Denner, J., Werner, L., Bean, S. & Campe, S. (2005). The girls creating games program: strategies for engaging middle school girls in information technology. *Frontiers*, *26* (1). 90-98.

Fiebrink, R. & Alcott, T. (2003). Designing a programming workshop for girls. Retrieved January 17, 2006 from http://www.cse.ohio-state.edu/~bbair/WIC/ Designing_A_Programming_Workshop_For_Girls.pdf.

Garmire, E. & Pearson, G. (Eds.) (2006). Tech tally: Approaches to assessing technological literacy. Washington, DC: The National Academies Press.

International Technology Education Association and its Technology for All Americans Project (2000). Standards for technological literacy: content for the study of technology.

Kahn, K. (1999). Helping children learn hard things: Computer programming with familiar objects and activities. In A. Druin (Ed.), The design of children's technology, 223-242. San Francisco: Morgan Kauffman, Publishers.

Kelleher, C. (in press, 2007). Learning computer programming through storytelling. To appear in Y. Kafai, C. Heeter, J. Denner, & J. Sun (Eds.), Beyond barbie and mortal kombat: New perspectives on gender and computer games. Cambridge: MIT Press.

Lin, J.M.-C., Yen, L.-Y., Yang, M.-C. & Chen, C.-F. (2005). Teaching computer programming in elementary school: a pilot study. *National Educational Computing Conference*.

Macromedia Flash (n.d.). Retrieved July 7, 2006 from www.macromedia.com.

McDowell, C., Werner, L, Bullock, H, & Fernald, J. (2002). The effects of pair-programming on performance in an introductory programming course. *Proceedings of the 33rd SIGCSE Technical Symposium on Computer science education*. 38-42.

National Research Council Committee on Information Technology Literacy (1999). *Being fluent with information technology*. Washington, D.C.: National Academy Press.

Snyder, L. (2005). Bringing fluency with information technology to high school. *CSTA Voice*, *1* (3), 3-4.

Wenger, E. (1998). *Communities of practice: learning, meaning, and identity*. Cambridge, England: University Press.

Werner, L.L., Campe, S. & Denner, J. (2005). Middle school girls + games programming = information technology fluency. *Conference on Information Technology Education*, New York, NY: ACM Press, 301-305.

Werner, L., Denner, J. & Bean, S. (2004). Pair programming strategies for middle school girls. *IASTED International Conference Computers and Advanced Technology in Education*, 161-166.

Werner, L., Hanks, B. & McDowell, C. (2005). Pair programming helps female computer science students persist. *ACM Journal of Educational Resources in Computing*, *4* (1). Retrieved January 17, 2006 from http://doi.acm.org/10.1145/1060071.1060075.

Williams, L. & Kessler, R. (2000). All I really need to know about pair programming I learned in kindergarten. *Communications of the ACM*, *43* (5), 108-114.

Author Information

Linda Werner
Baskin School of Engineering
SOE-3
1156 High Street
University of California, Santa Cruz, CA 95064
linda@cs.ucsc.edu
www.cse.ucsc.edu/~linda

Linda enjoys teaching software engineering and working on gender equity issues in computer science. When not doing that she likes to play trumpet, garden, swim, hike in the mountains, and play board games with her family.

Jill Denner
ETR Associates
4 Carbonerro Way
Scotts Valley, CA 95066
jilld@etr.org

Jill is a developmental psychologist who does research on teenage girls because she believes they have a lot to teach adults about how they learn and live. When not working at her computer, she enjoys spending time with her husband, daughter, and dog at the beach.

Shannon Campe
ETR Associates
4 Carbonerro Way
Scotts Valley, CA 95066
shannonc@etr.org

Shannon coordinates and teaches various research-based programs for youth. She examines how students learn and explores facilitation strategies to support that. Outside of work she practices yoga, rides her bike, enjoys live music, and spends lots of time with friends and family.