Hacker’s Mentality: Integrating Games and Hacking to build IT Fluency among Middle School Students

Ugochi Acholonu
Stanford University
450 Serra Mall, Bldg 160
Stanford, CA 94305
+16507997030
acholonu@stanford.edu

ABSTRACT
The power of technology lies in its ability to enhance our ingenuity to innovate and solve problems. However, this power is mute if an individual cannot bend technology to serve their needs. The author describes a game environment that facilitates middle school students’ abilities to manipulate technology by engaging students in the practice of hacking.

Categories and Subject Descriptors
K.3.1 [Computer and Education]: Computer Uses in Education
K.3.2 [Computers and Education]: Computer and Information Science Education—computer science education, literacy.
K.8.0 [Personal Computing]: General—games.

General Terms
Design, Human Factors, Theory.

Keywords
Middle-school, IT fluency, Digital Games, Task-Based Learning.

1. INTRODUCTION
The power of information technology (IT) lies in its ability to enhance our ingenuity to innovate and solve problems. To harness this power one must be confident and knowledgeable in manipulating digital devices. My research agenda is to understand what knowledge and practices support young people's abilities to use technology flexibly to solve problems and create innovations. By flexible use, I mean the ability to take everyday technologies (i.e., phones, laptops, software) and use them to use technology flexibly to solve problems and create innovations.

The following section will provide further details of the environment.

My dissertation specifically explores how to design learning environments that empower youth, particularly underrepresented youth, to adapt and construct personally relevant technologies. I have proposed a framework for my research that guides the design of learning environments that foster technological fluency. To test my framework, I am co-designing an online learning environment, with students in a local urban middle school. An initial prototype of the environment has been constructed and will be described in the following proposal.

2. DESIGN PRINCIPLE
All interactions in the environment were designed to help students discover the answers to the following three questions: 1) What things can be changed in technology, 2) how to make those modifications, and 3) Why would anyone want to modify technology? By supporting students’ discovery about the power they have over technology, it is my hypothesis that they will be more inclined and skillful in shaping technology to suit their needs, and the needs of their community.

The end result of the collaboration with the students is a learning environment that is a mixture of a role-playing, strategy game and a technology-authoring environment. The environment is designed to motivate a disposition towards using technology flexibly by engaging students in hacking practices.

The setting of game is an alien high school. The player is a spy in training who must get through various training sessions and challenges to reach agent status. To make it through the challenges a player designs and builds technologies to overcome a particular situation. All the technologies necessary to overcome challenges must be thought up, designed, and implemented by the user. All scenarios have a specific outcome (like unlocking a door with a the secret code); however, the scenarios are purposely designed so that multiple solutions can be used to achieve the desired outcome.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 2010 ACM 978-1-60558-951-0/10/06...$10.00.
Not all challenges require the user to construct the device from scratch. Along the way the player finds or purchases devices. However, some of the technologies that a player acquires will be faulty or have functionalities that are similar to what the player needs to solve a challenge, but not exact. In those cases the user will have to fix, adapt, or build upon what has already been created. Adapting, fixing, and building upon are what I call hacking practices. Engaging students in hacking practices is a game play strategy that serves two roles. The first role is to provide example technology designs to the user. These examples help the player see the range of possibilities for the look, use, and purpose of technology. The procured technologies provide the user with a library of examples to pull from when designing their own creations.

The second role of engaging players in hacking practices is that it positions the player in a relationship with technology that is other than creator. Many software programs designed to teach students about the construction of technology focus on one aspect of technology development process, such as programming. Moreover, many of those applications have students learn by having them create new elements from scratch each time. The merit of this approach is that the user has to have a strong grasp of the content to complete a task. The disadvantage is that the player never sees alternative ways of completing the task. Therefore, the individual cannot refine his own ideas by understanding the tradeoffs of different designs. By providing an avenue and encouraging students to study, fix, alter, and hack technology designs made by others, the learning environment allows the student to integrate their own understanding of technology construction to that of another’s. Moreover, the practice of building off another’s work is aligned with the real world work practices found in technology related careers.

3.2 Authoring Environment

Players will use the authoring environment to construct and alter technologies. The authoring environment consists of three different screens: The User Interface Screen, the Hardware Screen, and the Software Screen. Richer details of each of the three screens are provided below.

3.2.1 User Interface Screen

The User Interface screen is used to create the overall appearance of the device and construct the ways in which a person operates the technology. For example, the player can decide to make his technology look like a bear. He can also decide to use a drawstring as his input method to trigger the bear to play a sound.

As Figure 1 shows, the screen consists of a stage, an information box, and two menus—the Skins Menu and Parts Menu. The stage shows the current outer appearance of the device, called a Skin. The information box provides some general statistics on the technology, such as weight, size, and power levels. The Skin menu allows the user to quickly change the appearance of the device. A player can make his technology look like a bear, a cell phone, a flower, and other skins. The speed that a person can create a device is important for the progression of the game. Platform games usually have a fast pace cadence (about 2-5 minutes to complete a challenge). Thus, to maintain the familiar motif of this genre, the construction of the devices had to be quick and meaningful. Having a library of skins facilitates this goal.

Finally, the Parts menu allows students to add inputs (i.e., buttons, voice recorders, sliders) and outputs (speakers, monitors, etc.) to the gadget. Players drag and drop the different components into the appropriate hot spot. To coordinate the functionality between the input, outputs, and hardware, students will program their devices in the Software Screen.

3.2.2 Hardware Screen

To many youth, technology is a black box [2]. They understand how to use different gadgets; however, they know little about the construction of information technology and the underlying mechanisms that caused them to work. This lack of understanding has implications on students’ abilities to envision and transform technology for their own goals. To provide youth with a notion of what items they will find inside information technology and the role they serve, the Hardware Screen, shown in Figure 2, was added to the environment.
Using the hardware screen, players can add components such as memory, batteries, and transmitters into their gadgets. The hardware components the user selects influences what programming commands the player can use while programming in the Software screen. For instance, if no memory is added to a device in the Hardware Screen, an error will come up when the user tries to use the Set Variable Memory command in the Software Screen.

3.2.3 Software Screen

The Software Screen is used to add functionality to the device. It allows players to link input actions (i.e., a button is pressed, a signal is picked up by the receiver) to a corresponding output or hardware action (i.e., display an image on the screen, save data to memory). The software screen has an interpreter built in that executes commands. Figure 3 is a screenshot of the Software Screen.

The purpose of the Software Screen is not to teach programming per se, but to build student intuition on the purpose of programming and how it applies to more technologies than their computers. Through pilot studies I have found students have a limited notion of programming. In an interview with one young student, the student told me that he had programmed before because he had used the program Microsoft Word. The student’s reasoning was because he had used the program Word, he had programmed.

4. CURRENT RESEARCH

Currently I have completed an initial digital prototype of the environment. In order to test the application, I have conducted four user studies at two different middle schools. These tests were done in two phases. The first phase was used to understand student conception of technology and to improve upon the initial prototype. Activities included interviews, user tests with paper and interactive prototypes, and a card sort to see how students naturally classified technology. The second phase was used to evaluate the digital prototype outline in the Learning Environment section in terms of ease of use, learning, and likability. Currently, I am in the process of analyzing student feedback, and will use the feedback to further develop the application.

5. FUTURE STEPS

To investigate how to support student’s ability to flexibly use technology, I plan to convene a focus group of middle school students to participate in a study. The study will last for two weeks. In those two weeks students will use the application for a minimum of 1 hour at home through an online website. Pre and post-test will be given to assess their technical knowledge, confidence, and creativity with technology. Additionally, a transfer task of decomposing and debugging two different technologies will also be given. A control group will be given the post-test and transfer tasks for purpose of comparison. Furthermore, analysis of the different designs students created and hacked over the two-week study will also be done. I will be monitoring, through a logger, the various choices the user makes when implementing and adapting designs. Through the examination of the user’s choices and the posttest, I will assess the student’s progress and understanding of the technology design process and how the application influenced this growth.

6. REFERENCES
