ABSTRACT
The goal of the One Laptop Per Child program is “to provide children around the world with new opportunities to explore, experiment, and express themselves.” To stay true to this vision, educational software packages must be engaging and interactive. However, due to the XO Laptop’s modest system components, the software must be very efficient in utilizing the laptop’s resources.

This paper describes the One Laptop Per Child program, the XO Laptop, and the steps that need to be taken to successfully develop an educational program using Python and Pygame and port it to the XO.

Categories and Subject Descriptors
K.8.0 [Personal Computing]: General – Games.

General Terms
Design

Keywords
Educational Gaming, One Laptop Per Child (OLPC), PyGame, Currency.

1. INTRODUCTION – THE ONE LAPTOP PER CHILD ORGANIZATION
The goal of the One Laptop Per Child (OLPC) program is to provide children around the world with new opportunities to explore, experiment, and express themselves” [1]. Founded in 2001 by Nicholas Negroponte, the One Laptop Per Child Organization hopes to help educate the nearly two-billion people in the developing world that are inadequately educated by providing the world’s poorest children with a durable, energy-efficient, low-cost laptop with content and software designed to stimulate interactive learning. After all, OLPC believes any nation’s most precious natural resource is its children [1].

2. THE XO LAPTOP
With respect to the OLPC Organization, Negroponte has stated, “It’s an education project, not a laptop project” [2]. That said, the OLPC initiative is centered around the XO Laptop. The idea for the XO Laptop, sometimes referred to as the $100 laptop, was first sketched out by Negroponte in January 2005. Throughout the next two years, concrete design plans formed and in November 2006, the 875 B1-Test (Beta 1) model XO laptops were manufactured. The stronger, sturdier B2-Test model was released in February 2007, followed by the B3 model in May 2007 and the B4 model in July 2007 [3].

2.1. XO Hardware
The XO laptop’s CPU is an x86-compatible processor with a clock speed of 433 Mhz. Its DRAM memory is 256 MiB dynamic RAM. The XO has no rotating media, rather it uses a 1023 MiB SLC NAND flash, high-speed flash controller for mass storage. The laptop’s display is a 7.5” Dual-mode TFT Liquid-crystal display with a 1200 x 900 resolution [4].

2.2. XO Interface and Software
The XO uses the SUGAR interface, which the OLPC describes as “a ‘zoom’ interface that graphically captures [the] world of fellow learners and teachers as collaborators, emphasizing the connections within the community, among people, and their activities” [5].

This colorful, kid-friendly interface is very easy to use. It does not run software applications, but rather, “activities.” OLPC elaborates on the difference between software applications and activities, explaining that activities are more than just a new naming convention. They “are distinct from applications in their foci – collaboration and expression – and their implementation – journaling and iteration” [6]. Later in this paper, I will detail how to transform a program made using Python and PyGame into an XO Activity.

The XO interface also implements a “mesh network” that allows all laptops within a certain range to become interconnected. This opens up the potential for all activities to be experienced collaboratively [6]. By encouraging peers to learn and play with one another, the XO laptop not only makes education more enjoyable and interactive, but it also strengthens social and team-building skills.

As for software, everything on the XO is free and open-source. OLPC believes every child should be given the opportunity to use the laptops on their own terms. While OLPC does not expect every child to become a programmer, they “do not want any ceilings imposed on those children who choose to modify their machine” [7]. This also makes it very easy to develop new software for the XO.

3. EDUCATIONAL GAME DEVELOPMENT
3.1. Coins! My Educational Game
My educational game is entitled “Coins!,” and it is a coin-counting simulation using American Currency. The basis of the game is as follows: At the top of the screen a line of text presents the player with a dollar and cents amount, i.e. “1 dollar and 26 cents.” Along the left edge of the screen are images of a penny, a
nickel, a dime, and a quarter. Each time a player clicks on one of these coins, it is added to a box on the right side of the screen. The player must keep adding coins to the box on the right side so that the total value of the coins in the box is equal to the amount presented in the line of text at the top of the screen. Once the player thinks he/she has it correct, he/she clicks a button on the screen to check. If it is too high or too low, a message informs the player and asks him/her to try again. If it is correct, a new amount is presented.

3.2. Important Features of a Learning Game
In “Toward a Theory of Intrinsically Motivating Instruction,” Thomas W. Malone argues the five concepts all educational programs should have are [14]:

- Clear goals that students find meaningful;
- Multiple goal structures and scoring to give students feedback on their progress;
- Multiple difficulty levels to adjust the game difficulty to learners’ skills;
- Random elements of skill;
- An emotionally appealing fantasy and metaphor that is related to game skills.

Although Malone first created these concepts in 1981, they are still very relevant and should be considered when brainstorming an idea for an educational XO game.

Another concept to consider when brainstorming game ideas is that the XO’s target market is children in the kindergarten to fifth grade range. The target audience needs to be taken into account while choosing the educational content of your game. This is the reason for choosing a coin counting simulation game. It allows children to practice basic arithmetic through a real world application.

3.3. XO Limitations
When brainstorming ideas for an educational program, it is important to consider the limitations of the XO’s hardware. While modest system requirements allow the XO to have low manufacturing costs and help push the laptop closer to the $100 price objective, they do create limitations on what kinds of programs can be run on the machine. However it is hard to say exactly how much or how little the XO can handle; a highly interactive game with intensive graphics and animation will certainly not run efficiently.

3.4. Python and PyGame Limitations
Python and the PyGame modules make developing a game straightforward. Python’s website has the language’s complete documentation, as well as numerous tutorials on how to program in Python [9]. Mark Lutz’ Learning Python is an especially valuable resource for learning the fundamental concepts of the language [10].

PyGame is a relatively new Graphical User Interface (GUI) component that enables the creation of fully featured games and multimedia programs in the Python language. It is highly portable and runs on practically every platform and operating system [11]. PyGame is an effective, easy to learn tool for making games. The games are most easily developed on a standard PC and ported to the XO then developed directly on the XO.

4. Porting a PyGame to the XO Laptop
4.1. Making the Program XO Ready
Once the PyGame is completed and running on your computer it is ready to be ported to the XO. However, before porting it must meet two pre-conditions.

The first pre-condition is that the program must have a main() method. While Python programs don’t always require a main method, a SUGAR activity does. Guido van Rossum’s weblog at Artima Developer has a great tutorial on different ways to add main methods to a Python program [12].

Also remember what module in your program contains the main() method. This information is necessary to complete the porting process.

The second pre-condition is to check that the PyGame runs as desired on the XO laptop. This can be done by copying all the program files to a USB drive and running the program from the XO’s terminal.

4.2. Building The Activity Wrapper
Now that the program is XO ready, you need to build the game’s activity wrapper. The activity wrapper allows a PyGame program to be open and used as an XO Activity, like the ones already included on the XO Laptop. These instructions are a more streamlined version of those found on the OLPC wiki website [13].

To build a wrapper, download the latest version of the OLPCGames wrapper from http://dev.laptop.org/~mcfletch/OLPCGames/ using the Terminal’s ‘wget’ command. Enter the command lines shown in Figure 1 into a new terminal window, and you will change directories to the home/olpc directory, where the latest OLPCGames wrapper can be downloaded.

```
        cd ~

        wget http://dev.laptop.org/~mcfletch/OLPCGames/OLPCGames-1.6.tar.gz

        tar -xvf OLPCGames-1.6.tar.gz
```

Figure 1. Command Lines used to Download Activity Wrapper

Now that the wrapper is downloaded, it is necessary to run the buildskel.py script which is located in the skeleton directory. The script takes the following four arguments:

- The name of your game.
- A human-readable name for your game.
- The name of your main module. This is the module that contains your main() method.
- The name of the main function within the main module.

An example command line with the above arguments is given in Figure 2.
python buildskel.py –n GAME_NAME –t HUMAN_READABLE_NAME –m MAIN_MODULE_NAME:MAIN_FUNCTION_NAME

Figure 2. Example Command line used to run buildskel.py script.

The skeleton directory should now contain an activity folder named “GAME_NAME.activity”. Once the activity folder is created, all of the files that are stored on your USB should be copied to this folder. The game is now ready to be wrapped into an activity bundle.

4.3. Wrapping The Game
Wrapping the game, the process of applying an Activity Wrapper to a PyGame program, is a simple process. To apply the Activity Wrapper you made in the previous step to your PyGame program, you need to register your game as an activity. To do so, go to the game’s activity folder and enter the command line shown in Figure 3.

python setup.py dev

Figure 3. Command Line used to Wrap the Game
A restart of the XO is required after completing the above.

4.4. Testing The Activity
Testing the activity thoroughly is recommended. This is done by clicking on the hammer icon in the activity icons. If while playing your activity, you discover errors, the “Log Viewer” activity can be checked. The “Log Viewer” activity icon is accessed by clicking the icon that looks like a box that contains a list of three bullet point list items. If you find changes that need to be made, change them on your USB and then copy the files over to the XO again as previously explained.

5. Conclusions and Future Research
Coins is a relatively simple game that was conceived, developed in Python and PyGame, and ported to the XO laptop as an 8-week research project.

It would be greatly enhanced and made more entertaining by adding sounds, multiple levels of difficulty, a scoring system, and smoother animations. Other possible extensions would be to add additional game modes that will ask the player to compare the values of two different collections of coins or allow the players to make change by subtracting coins from a collection to reach a certain value. These would not only help make the game more fun, appealing, and educational for children, but would also help the game to become better aligned with Malone’s characteristics of a good learning game [14].

6. Acknowledgements
This work was done at the University of South Carolina as part of the Research Experiences for Undergraduates in Multidisciplinary Computing project and supported in part by the National Science Foundation (Award #0649105). I would like to thank Dr. Duncan Buell, Dr. John Bowles, Dr. Caroline Eastman, and Kimberly Yonce for their help and guidance with this project.

7. References


