**Daniel Mendelsohn**

6.115 Final Project Proposal 2

April 28th, 2013 (Late)

**Introduction**

For my 6.115 final project, I will build an implementation of the game Pac-man. For the game display, I will use an oscilloscope by plotting two input signals on orthogonal axes. For user input, I will build a hardware system that allows users to physically indicate commands. I will build two parallel, upright panels; one of which will have LEDs and the other of which will have photo diodes. When a user swipes their hand between the two panels, the system will use software control to determine the direction of the swipe. The game software would then use this direction the same way a “simple” Pac-man implementation would use a simple (‘Up, ‘Down’, ‘Right’, ‘Left’) key press. The game itself will be for only one player, and it will feature multiple levels of varying degree of difficulty.

I’ve always been interested in creating games, and as a Computer Science major, I get to play with a lot of different technologies. Working in a combined hardware/software system, and on the lowest possible level of programming, is a brand new and eye-opening experience for me, and I want to extend my interests to this world of EECS.

**Hardware Description**

There are two main hardware components: the display and the user input mechanism. The only chips needed for the display are two DACs. Each DAC output feeds into an input of the oscilloscope. By varying the DAC outputs independently, I effectively get a 256x256 pixel display on the scope. Without careful calibration and variation of the values, I can draw lines and figures as desired. At this point, I plan to use the DAC from previous labs, but I may change my mind later on and put the DACs on the PSoC.

The user input hardware is more complicated. I will build two physical crosses, one of which has 5 LEDs, the other of which has 5 photo diodes. See the diagram later in this document for a clearer picture. Each LED is lit whenever the system is turned on. Each photo diode feeds into its own ADC. Having each diode have a dedicated ADC eliminates some “switching” overhead and makes it much faster to read since I don’t have to wait for values to latch within a single “poll cycle”. To simply my life (a lot), I’m going to configure the PSoC to have all these ADCs. This is certainly easier than placing 5 ADCs onto my kit. The digital outputs from these ADCs will lead to their own output port on the PSoC, each of which in turn will connect to the 8255. This way, my R31JP program can simply periodically poll the ports of the 8255 for values.

I’m also hoping, if I have extra time left over at the end, to implement sound effects using the audio amplifier chip and a DAC.

**Software Description**

The software will be the lion’s share of this project. The code will have to be very well organized and modular. There will be some “high-level” game logic code to handle new games, changing levels, character locations and movements, and so forth. There will also be a significant amount of display infrastructure that will allow me to display the entire content of the screen at a rate of about 60Hz.

The input handling code will periodically poll the 8255 to get photo diode readings. It will then interpret these readings to figure out of an obstacle is in the way, namely the user’s hand. The software will be configured as a state machine so as to simply handle the logical processes of determining when the user has made an action, and which action the user has made. The periodicity of the polling will be controlled by an external interrupt via the 8254 timer chip.

If I have leftover time for sound effects, I will implement them the same way as in class; namely repeatedly running through a database of values and sending them to the audio amplifier (via a DAC).

**Project scope and management**

My goal, in order to achieve at least a B, is to make a playable Pac-man game with some sort of light-based user input. I believe that even if my fancy “swiping” motions don’t work, and I end up hacking together a simpler system (i.e. the user just places their hand in front of a diode labeled ‘Left’ in order to indicate the ‘Left’ command).

In order to achieve an A, I will have to implement at least two levels of Pac-man of different difficulty, and I will have to create a decent implementation of the “swiping” user input. As a bonus, I hope to implement sound effects, though I consider this an entirely auxiliary goal.

**Special component needs**

I will 5 LEDs and 5 photo diodes, as well as a very simple physical setup of those parts. Other than that, I will need only parts we’ve used in class, namely the R31JP, 8254, 8255, LM138, and DAC, as well as the PSoC.

**Timetable:**

The week of April 15th has passed.

The week of April 22nd has passed.

During the week of April 29th, I will build the display infrastructure for my program. Namely I will be able to easily display a large number of lines, as well as some simple figures, on the scope. By week’s end, I should be able to display a static “screenshot” of my game.

During the week of May 6th, I will build the game logic of Pac-man. By week’s end, I should be able to play the entire game with keyboard input.

During the week of May 13th, I will build the light-based input system. By week’s end I should be able to play the complete game as specified in this proposal. Hopefully there will be sound effects!