EE 445L – Lab 11 (Final Project)
Justin Nguyen and Trevor Murdock
December 2nd, 2016

1.0 OBJECTIVES

See the requirements document

2.0 HARDWARE DESIGN

See the schematic file

3.0 SOFTWARE DESIGN

Our game engine is actually really nice. Every game tick we simply read in the thumbstick inputs, normalize those values, erase the existing entities, update their position based on the game tick, and then redraw them. The foundation for all of the code is based around our classes since we were able to get C++ code running. Since our code is really modular, adding new features to the game (such as a little "fire trail" that follows the ship, adding sound, or new power-ups) is as simple as defining that entity and then pushing it onto the entity list for the game engine to take care of later down the road. In addition, we hunted for memory leaks to make sure that our quadtree was still operational and be used to efficiently detect collisions. Though we didn't write enough features for this to be all displayed in our final product (since we spent most of our time working on getting the ESP to work), the foundation for our game is all there and easily changeable.

4.0 MEASUREMENT DATA

Overall current drain (seen by plugging the system into the power supply): ~130 mA
Approximate time the batteries lasted:

One side of our crystal runs at 16 MHz (seen through the oscilloscope) but the other side doesn't show any kind of oscillations.

The digital range of our accelerometer: 1650 – 2450
our thumbsticks: 0 – 4050

The digital output of our accelerometer when stationary
- x-direction: 2050
- y-direction: 2000
- z-direction: 1650

The digital output of our thumbstick when stationary
- x-direction: 2100
- y-direction: 1950

We tested our quadtree by limiting the size of our entity list array and creating enough entities to hit that limit (at which point no more entities were created). We originally had memory leaks that would cause our system to crash upon producing too many entities, but we fixed that. We also drew the gridlines of the quadtree when debugging to make sure that collision detection was working.

We visually tested for lag by moving the ship around, firing bullets, moving the bullets with the accelerometer, and having sound output upon each bullet fire. We saw unnoticeable lag in the game.

We tested our ESP by outputting commands and seeing the server/access point response on the screen through UART and PuTTy. We looked at online videos but they were never thorough enough. We read through the ESP AT command manual and many forums but never found very good documentation. We also saw a lot of webpages describing firmware issues with specific ESPs and never conclusively found out if ours could have been one of those.

5.0 ANALYSIS AND DISCUSSION

Here is the link to our YouTube video: https://yout.be/4hkyUApRykw