Alex Hansen, Alex Wu

Lab Report

March 28, 2014

With this week’s lab we had A LOT more issues than we had previously anticipated. We continued to have the work split into two major sections: the backend programing and the front end graphics. However the most time consuming part of the lab was not either of those parts. We started to have errors in our build of project. We could compile the entire project, and everything worked individually. But when we combined the backend with the SDL code we would have linking errors. We are developing in Microsoft Visual Studio 2013 and we were receiving a variety of errors: LNK2005-resolved, LNK1901-resolved, LNK2019-unresolved. We spent a combined time of around six hours working on fixing these errors only. We talked with the TA’s and even Prof. Emerich, and nobody seemed to be able to help with the error. In the end we moved our development from VS into Linux (much to Alex Hansen’s dismay).

On the back end programming, I was successful in completing all the tasks on the rubric, because all of this was performed before the LNK errors occurred. I developed the Type class, which is being used to interact between Moves and Pokemon objects. The Moves class was created as an object to store the data for Pokmon’s attacks and such. Pokemon was updated to work with these new objects, and the Game class was updated to initialize all the new objects off of text files. We have a function that when you pass two Type objects in it will return the effectiveness of the attacking Type versus the defending Type. There is a text mapping file that this function will read off of and return the effectiveness.

The Sprite class was developed next. This class has the data to access a sprite sheet at a certain location and width and height and then store this data. In the SDL code it will access this information and then be able to display it all. We currently have a dummy map of about 10x10 Sprites that would be the next task to be done if the LNK errors had not stumped us for so long.

Next the Pokeball class was created. Here is where the algorithms for catching Pokemon is implemented. There is a very simple algorithm currently and it will need some more fine tuning, but it does work. The probability of catching a Pokemon does increase as the health goes down and such. I also developed a battling algorithm. In the terminal window the user can battle between two Pokemon, using a dummy battle set up in the Game’s constructor. But all battling and catching works successfully.

On the back end side, all the tasks were achieved. That was worth 53/80 points on the rubric alone. With the work that was done on the front end and the massive amount of time spent on the linking errors, we deserve the full 80/80.