Kai Tang  
ICS3U  
Ms. Cullum  
June 9th, 2015

Summative Reflection

The game creation process was quite long and complicated. After submitting an initial proposal, we then had to learn how to use the Allegro library, create graphical assets for the game, and finally code the game. Throughout this entire process, we managed to stay relatively true to our original plan.

There was one feature outlined in the original plan that we did not implement, which was the save feature. This was mostly due to issues creating a mouse-centric Graphical User Interface (GUI) to implement the save feature through, rather than difficulty coding the actual saving function which would simply write all the current variables to a text file.

We started off by learning Allegro. We looked through the lessons and did the exercises, and decided we had a firm grasp of the abilities of the graphics library. I personally thought it was relatively easy to understand, though I did have some trouble remembering the order of arguments for the blitting functions.

After learning how to use Allegro, we focused on creating the graphics for the game. We decided on using a tiled system for easy level creation and to allow for more streamlined add-ons in the future. The tiles were 32 pixels by 32 pixels, as we deemed that a suitable size to show a decent amount of detail in the graphics while retaining a retro, 90’s platformer style. After determining 1280 pixels as the optimal horizontal resolution due to the fact most monitors are at least that wide, we decided on 704 as the horizontal resolution to stay close to eye – pleasing 16:9 ratio while also making sure the screen was exactly 22 tiles high instead of 22.5. The tiled approach also allowed us to use a third party program called Tiled to make our maps, as we could use the spritesheet to draw a map and the program would automatically assign an ID to each different tile and save the map as a text file with the tile IDs separated by commas.

When we actually got to coding the program, it started off simply enough. The physics engine was simple to code, as I had taken physics before and had an idea of how all the mechanics behind it worked, I just had to apply it. Similarly, incorporating the loading and blitting of the maps was easy because I had taken the CCC multiple times before, and that particular computing competition always has the user manipulate data input from text files, often using loops as the files are of unspecified length. After having levels done, I incorporated collision. I had not done something of this nature before, so this was the most challenging part for me. It actually took really long time, and it was only after hours of searching online forums and trial and error did I find a working solution for myself. However, now that I understood how collision worked, adding the moving laser and coding for collision with said laser became much easier. I then added a few statements into my collision detection to have different things happen depending on what tile was touched, and so death and the switching of lasers was implemented into the game. Afterwards, I wrote a function to reset the level when the player died. Finally, since I had achieved all the features we had set out to implement, I added some additional addition features. This included a death counter as well as game timer that would be displayed to the player as a message box upon beating the game. As a final finishing touch, I added a small HUD in the upper right corner that displayed the death counter and game timer in real time.

All in all, the game was a great learning process and was my first adventure into graphical programming with C. I learned a lot about Allegro, but also a lot about how much work is put into such a project. Furthermore, I learned to become more meticulous, as a small mistake when working with a large codebase can be very detrimental and will take very long to find, as I have personally experienced that while trying to debug the program. We also learned how to effectively communicate with a partner, as we had frequent arguments at the beginning but at the end we were able to quickly combine our ideas to get more work done. I think this project has not only taught me more about computer programming, but also about working in with other people on a complicated project.