HW5 Design Document

Karan Rakesh

Part 1 : Implementing scripting was relatively complex, because the documentation on how to set it up is so poorly written that it was nearly impossible until one of our classmates wrote a how-to and gave some clarity. The documentation also doesn't specify how to use the scripting language with enough code snippets so it was hard to learn it quickly. So, I implemented basic tasks just with the aim of satisfying the requirements as I found this scripting software very unintuitive. I must mention that I did the assignment before someone posted about dukglue, which apparently made the scripting slightly easier. I implemented setting of the screen number using a script to handle the event. And for the regular updates, I implemented a button which could be pressed on the client side to change the colour of the sprite. I didn't notice any change in the performance of my program since my scripting component was very minimal and definitely couldn't have much performance impact. This didn't take much time after the succesful setting up of duktape and understanding one of the example programs.

Part 2 : I implemented a Snake game for my second game. Here I basically handled the food on the server side so everytime the snake hit the food the server would send a new position to place the food at. This was done by building a custom random computation by using the time from our timeline class. The snake was stored in a vector and everytime the snake moved, I popped from the end of the vector and then inserted the new position at the front. I basically treated the snake as a bunch of independent characters which simulated movement by getting created and then destroyed. I was able to accomplish this with not too many changes required to the basic 2D platformer, which was very nice to see. I've mentioned other design decisions in the reflection document and why I kept the variable names the same. THe hardest part in making this game was to keep the composition of the program as close to the 2D platformer to ensure maximum similarity which involved rethinking the logical representation of certain variables. This was challenging yet very rewarding.

Part 3 : For the third game, I initially wanted to make pong but since it was not possible to interact on two screens at once, I decided against it and found another similar game which I decided to go for, Breakout. I chose this game because it had some physics to consider regarding the bouncing of the ball along the walls and handling collisions that had to destroy the blocks. So these were both elements that I had not explored yet. Hence I built a basic Breakout level and after lots of calculation and trial and error regarding the angles of the ball trajectory I was able to make a simple breakout level with 2 rows of bricks and able to break all of them without too much problem. I also wanted to experiment using text, so I created a text that said you win or you lose whenever required for which I had to remove my condition of immediately closing on loss and replace it with a pause and display the text, on which the user can exit the window. I proceeded to add a game over text to the snake game as well. Since there were definite end states, I didn't add any score metric which would have involved similar logic.

This assignment was significantly easier than the previous ones, probably in part due to the experience we have gained through the course and as a result of the continuous emphasis on building an engine which was capable of handling more games and not only being tightly coupled to the 2D platformer. I believe the decisions I took made my life much easier and this was definitely a great experience learning SFML, different game engine modules and learning to build them from scratch.