Design Document (H.W 1)

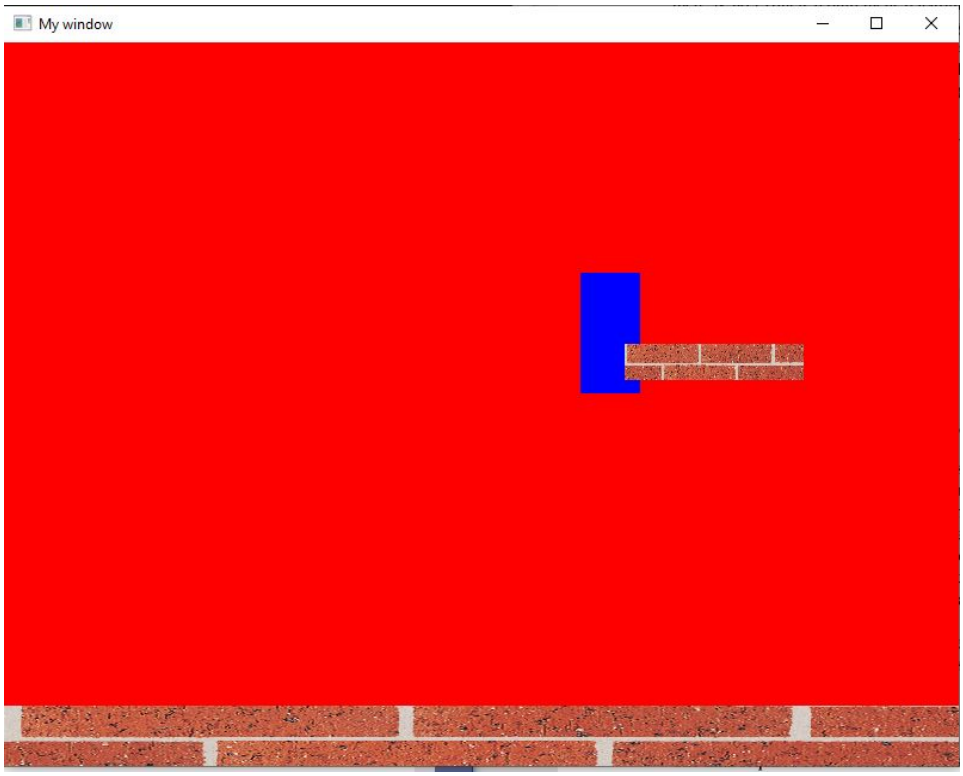
By Karan Rakesh

Section 0-1 : The installation of VS 2019 and SFML software went by without much issues. But the linking of the files as mentioned in the documentation provided proved tricky. I had initially tried including everything as a static dependency but there seemed to be an error, and to move on ahead I decided to switch to dynamic linking. Unfortunately, even on further reading mentioned in the documentation, I was unable to understand how to get static linking to work. Though I have had experience coding in C++, I am not very familiar with VS, so it took some trial and error to figure out where to place the dlls to run the files. The window resizing part was pretty straightforward, though when you go into fullscreen mode, I was unable to figure out how to access the navbar, and had to Alt-Tab out to force close it.

Section 2: Design -wise I decided to inherit all my shapes from RectShape since I expected that at some point down the line, the independent intersect() functions might come into use if there are other obstacles which have to collide with these three. I also made all these classes completely independent since the specification and my intuitive understanding of a character, platform and moving platform didn't have much in common to link them through deeper inheritance (moving platform being a child class of platform). I faced a major roadblock at understanding how to write the constructor to successfully create a RectShape after inheriting from it and finally got clarity about it from one of my peers. I also decided to put the texture on the platform and moving platform since they can both use a similar texture, and loading multiple textures is an expensive operation as per the documentation.

Section 3: Input handling was fairly straightforward. I was able to implement left, right, up and down. But then to add the element of gravity I intentionally added a down motion whenever up button wasn't being pressed. I also decided to omit the use of chrono.h to implement a timed jump and instead implemented moving upwards with the up button and falling on releasing it. I also manually implemented motion of the moving platform by using a ticker that counts up number of renders before reversing the direction of the moving platform. I had posted on the forum and gotten clarity regarding this part. I also tested using mouse movements, but decided against using it in the final implementation.

Section 4: My decision to use RectShape for the character made this part very straightforward. I only had to implement the intersect() and make sure the bounding boxes didn't touch. Since I was building a basic implementation, I didn't consider corner cases like the character hitting the moving platform midway through up/down motion. Instead, I just dropped the speed to zero so the character will hover at the same spot till the platform bounding box and the character bounding box stop intersecting. I have included a figure to explain this.



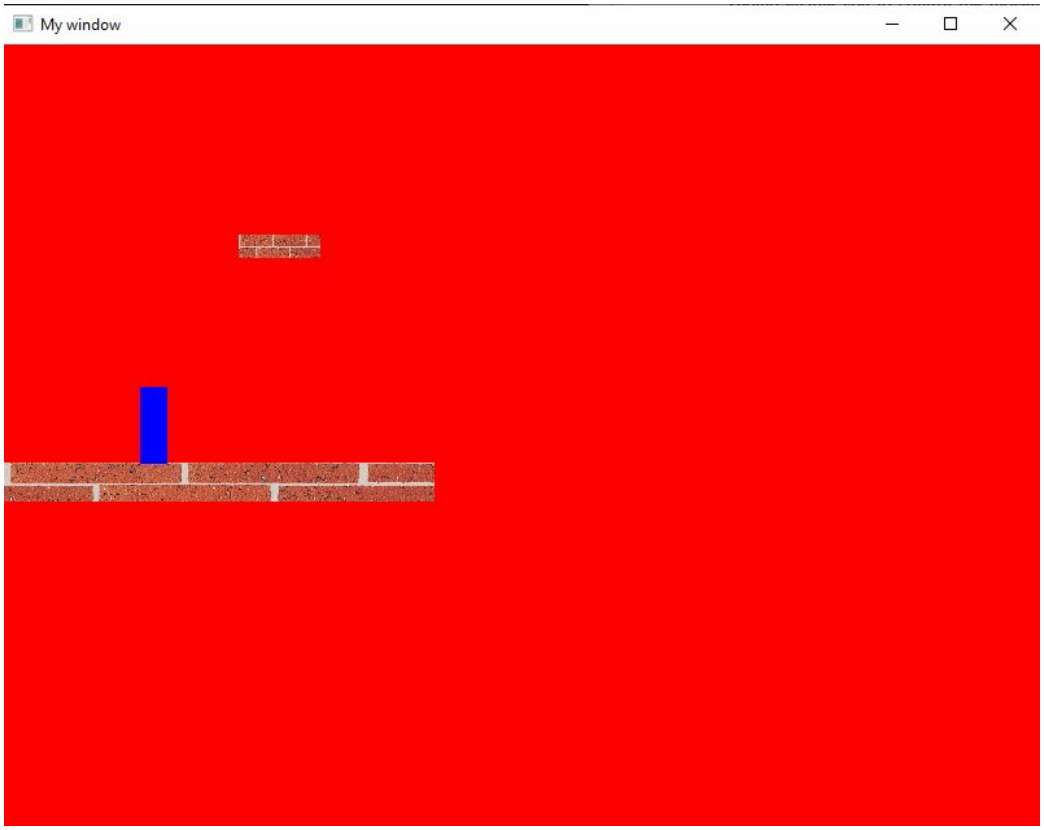
Section 5 : The window in my installation seemed to exhibit proportional mode by default and so initially I looked up tutorials regarding the same and was first led to a tutorial in SFML v2.2 but found similar updated documentation in v2.5. So I implemented constant size by using the setView() function. I bound the switching to key ‘R’ and to ‘T’. I made to implementations to perform the same function due to the ambiguous nature of the requirement. I used ‘R’ to leverage the event system to handle the toggling between modes and used real-time input with ‘T’ to implement the same functionality. The event system implementation works better than the real-time input since it handles the case of multiple keypresses registering for a supposedly single keypress. This method of resizing works fine in most cases but fails for the corner case where you :

- Turn on constant size
- Resize to a bigger size
- Turn on proportional size
- Reduce size back to original size

Reverting to the original view requires the following steps :

- Turn on proportional size
- Resize to prev bigger size
- Turn on constant mode
- Reduce size back to original size.

I have attached an image of this corner case below.



.

All my decisions were aimed at getting a basic implementation up and running since I spent a significant amount of time trying to figure out how to get the RectShape to render properly on my window itself, so I had to plan the remainder accordingly. I will probably make some changes as we move to further assignments and I get more comfortable with SFML.