**Small Development Blog**

**VIDEO PRESENTATION LINK (UNLISTED VIDEO)**

[**https://youtu.be/1I19ZqO9Iws**](https://youtu.be/1I19ZqO9Iws)

[13/01/21]

First entry log; because of my poor luck and time management I’ve decided to stick with a simple theme of the game Mario. It will allow me more time to continue with the game functionalities and keeping it simple would allow me more room for mistakes.

I’ve began following a tutorial about tile based level maps which I believe would be a good foundation. I’ve also researched more into JavaScript language and its syntax.

[15/01/21]

Following from the tutorial, I had issues implementing similar features which I had to find workaround for. Finally, I have a display using HTML5 that draws my tile based level from array of characters. It only loads the level and ground elements and player, which are just scaled boxes by 20x20 with a different colour using the style sheet.css.

Following this tutorial further, it shows the use of a canvas that will allow drawing of sprites instead of just blocks of colour. This is my next step before I start construction on the game functionalities

[17/01/21]

I’ve finished converting to a canvas display, which now allows me to draw any image on my actors. I’ve also managed to create simple movement and collision detection of the player to test out the viewport centring on player with this canvas.js that now is working.

I’ve started sourcing and editing most of the graphics and sounds I needed for the game before I continue any further with the game.

[19/01/21]

I’ve gathered all the assets I need to develop the game further, I’ve loaded each asset as its own variable for future use when new actors are added in. The player can now jump, collide with walls and boundaries. The sprite now flips horizontally based on which direction its velocity.x is moving. The game has a fully functional state system where depending on state status of the game, it will change respectively; “playing” = the player is playing the game “lost”= the game restarts, or if “won” it will start the next level and go back to the status “playing”.

Subsequently, I will focus on adding more actors very similar on how the player class is created and added to the actor list.

[21/01/21]

Past few days I’ve been struggling to add a specific actor that needs to be in the static collision map and the actor list and also they both need to act as the same object. I’ve managed to work around it, which is very sloppy, but it was draining my time. Since I’ve found a solution for my problem I’ve continued and added all the actors I needed into the game with no functionalities yet. They only are loaded and drawn. Yet I did manage to create the coin actor that now interacts by overlap with the player, which disappears when player overlaps the coin and score is added.

[24/01/21]

Once again I’ve encountered an obstacle which stops me from spawning actors on the fly, I’ve attempted to work around the current system yet the timing for state updates doesn’t allow me to add a new actor to the list because it seems to be drawn a state before which is why it doesn’t show on the canvas. I’ve decided to not spawn actors on the fly so, some of the features such as the ? block in Mario won’t be implemented fully.

Since I have not done much throughout those days, I plan to continue adding all actor functionalities before polishing the game and creating static levels.

[28/01/21]

All actors now have their functionalities working, which includes a collision response and simple update function. I’ve managed to add these actors into the game; player, coin, powerup, ? block, 3 enemies, water, wall, finish flag. They all seem to work and load fine without any bugs.

I’ve also added 5 levels which a simple duplicated templates for later level design. The first level will be a main screen, next 2,3,4 will be 3 game levels that will be playable and the last will be the end screen when all 3 game levels are complete. I also added temporary high scores / scores for each level displayed on the canvas as UI using fronts. The last level will contain all high scores and a re-playable feature.

The game was lacking a background since it was only filled with colour. So, I implemented an infinite scrolling parallax background based on the viewport, it includes 4 layers of images all moving at different x speeds.

Next of the list is polishing the game and creating levels.

[30/01/21]

Within two days I’ve polished all the assets and functionalities and the states, so the game can be re-playable and loops all through 5 levels. I’ve added simple fonts to display, tiles, scores, tips. I’ve also created the 3 levels of the game by trail and error. The game does seem playable on a browser but when wrapping the game in android studio, it seems to run slower and slower and the fill-parent or match-parent functions in webview did not work with my project. Probably something to do with my canvas. Sadly I don’t have enough time to implement mobile input, so the game only uses wasd/r inputs.

Other than that, the game runs fine and is playable and reaches the specifications. I do need to comment all the code as I only did it briefly when worked on it.