**5.3** Project Log

Friday 22nd March: Started development of game, managed to get one of the planets in a circular orbit about a point by using the math module of cos and sin functions.

Tuesday 26th March: Created the Planet class and moved this code into it, also created a bunch of planet objects that pass in their own size and distance, radius etc, but at this point they are all moving at the same speed because I didn’t realise that the theta variable was shared between all of the planets. Also added grey circles showing the planets orbit.

Wednesday 27th March: Fixed the planets orbiting at the same speed by creating a self.id variable which is unique to each object, and then using a list to store all of these theta and angle values to calculate orbits. Also, I managed to get a smaller object to orbit around one planet at a time, which I can change by changing one variable in the code.

Friday 29th March: Turned the ship into its own class because there are certain things I need from it. I also made it move between planets automatically, at this point just by a timer and it draws a line between the planets to make a cool effect of kind of a ‘hyperjump’. I started work on the animation that I had planned for the ship, where it would get smaller as it starts to jump, but I can’t get the code right on that, so I’ll have to add it to next week’s goals.

Monday 1st April: Finished the ship animation by creating a variable ‘size’ and testing if it was above a certain point, and if it was it would divide it by 1.25 which gave the ship a cool effect of jumping between the planets. I also started to look into how to change the code over to the arcade window class for user input.

Wednesday 3rd April: Moved everything over to the window class after a lot of struggle. It was really difficult to do this because of the uniqueness of this to arcade, and because there is very little documentation about arcade online, but now it’s ready for player input and stuff. I also managed to get the WASD keys to change how the ship jumps.

Thursday 4th April: Watched a cool YouTube tutorial on how to make 2d sprites in the form of planets in blender. I’ve used blender a lot, so it would be much easier if I could do this instead of having to individually draw the planets. I created an animated sprite for Terra, Morp, and Cronus. Once I got into the rhythm, I could make them pretty fast, only taking about 10 minutes each.

Friday 5th April: Made all of the rest of the sprite, or at least the templates, using a mix of pixilart.com, for the 2d sprites, and pixlr.com/editor/, for GUI elements. Not really much to document here, it just took a long time. Taking a bit of a break over the weekend

8-10th April: All of what I did for these days were just working on the same element, so I just mixed it into one point here. I made some base code that I could use to copy and paste over to the planets and just got it all done. Originally, I had a problem where the sprites where getting drawn over each other and making a mess, but that was fixed by deleting the contents of the Sprite List before adding anything else. I also had some problems with stuff getting drawn on top of other stuff, but I could just fix that by changing the order in the code. Because this didn’t take as long as expected, I changed the project schedule for the rest of the project to 2 days prior than what I wanted

Thursday 11th April: Created system that allowed a new sprite to be drawn if landing on planet by pressing space and stopping everything else being drawn to prevent lag. This went without any problems.

Friday 12th April: I played around with 2d based arrays in python today and found that they aren’t as easy or useful as they are in other languages, so I had to switch it up and use a list for the grid system instead.

15- 17 April: Finished tile-based framework for planetary grid and implemented ways of receiving the data in the list and drawing a sprite at the correct location using this.

Friday 19th April: Created user-based input via mouse input clicking and got buttons to work in the form of the buttons to the right side of the screen where you can click between them and move between the planet landed state. Also started on getting a visual feedback for what tile is clicked using rounding and stuff.

Monday 22nd April: Made buttons and dropdowns for when clicked on a planet and made them change a value in the list grid

Tuesday 23rd April: Started on the code to draw the sprites for the different values in the grid list.

Wednesday 24th April: I’ve created the resource variables and I’m starting to figure out exactly how I’m going to implement them because the way that they work is different to what I have in mind, so I’ll need to rethink this entire system.

Thursday 26th April: Resource system completely implemented, so now I need to actually change these values as a result of what is built and also make it so things can only be built if you have enough resources

29th April – 2nd May: This took a lot longer than expected, mainly because there are a lot of things to worry about when making buildings as everything takes multiple resources so every time you need to build it needs to check if you have enough resources and then deduct these. Because of this I needed to extend the project schedule to 2 days ahead of what I wanted, which is what it originally was. The reason for this taking so long was because of the complex logic for each building and inability to copy and paste code.

Friday 3rd May: Displaying text on the left side of screen for global resources + and – and net. I intended to get more done today but I made a mistake when I first started doing this and meant I had to basically redo the code, but now that I have it, it should be fairly easy to implement to the other text UI elements.

Monday 6th May: All of the text for the different resources are now being displayed in the correct position, so I can now start to test the program and fine tune some of the resources and stuff.

Tuesday 7th May: Win and loss condition added so that if the net population is above 75 you win and below 75 you lose. I also needed to make a word document as a tutorial as the few people I’ve given my game to have all said it’s a bit confusing at first.

Wednesday 8th May: I ran the program through the testing table and fixed anything that needed fixing (see testing table). I finished the game 2 days before the planned finish date, so I’m pretty happy about that and will just play and fine tune small aspects of the game until the due date.