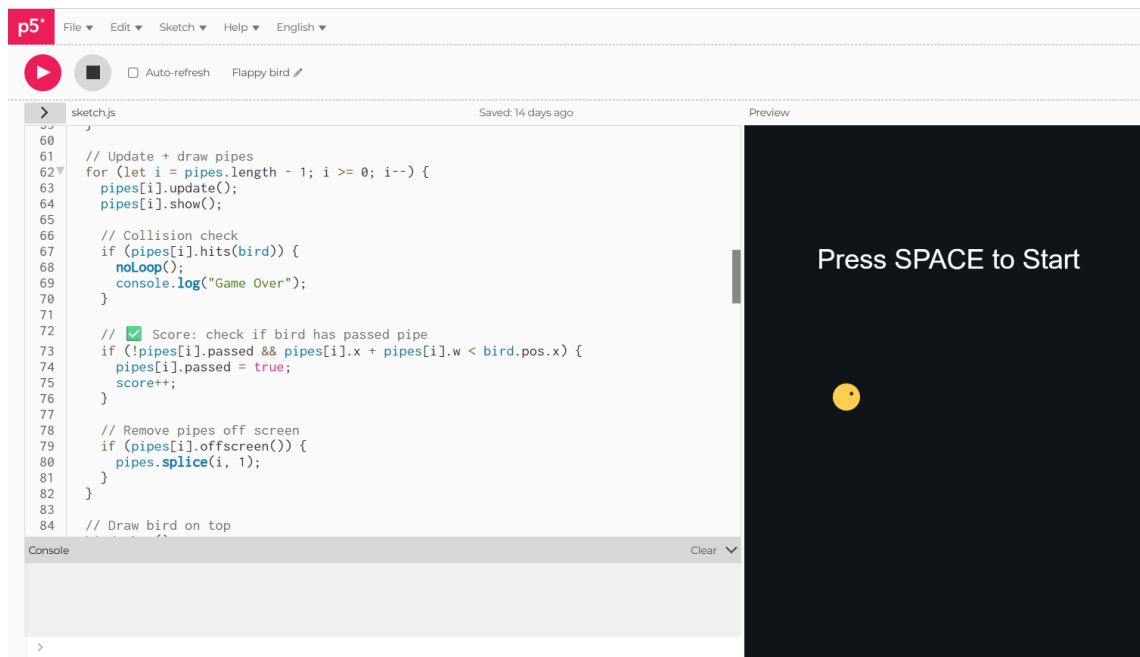


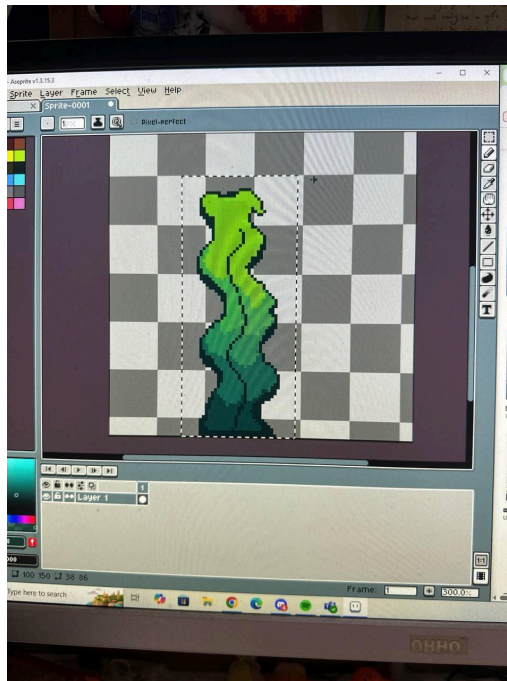
FlappyJelly Process Journal + Reflection

In this project, I used Javascript and Aseprite to recreate Flappybird. In addition... I did use the assistance of AI in some sections which are commented on in my code. I'll be sharing my progress and my thoughts on the activity towards the end. My partner and I thought of redesigning it into an underwater theme with underwater elements like seaweed as pipes and a jellyfish as the player. We used the guided template to design the game mechanics so it was quite simple to create the physics, collision and score system.

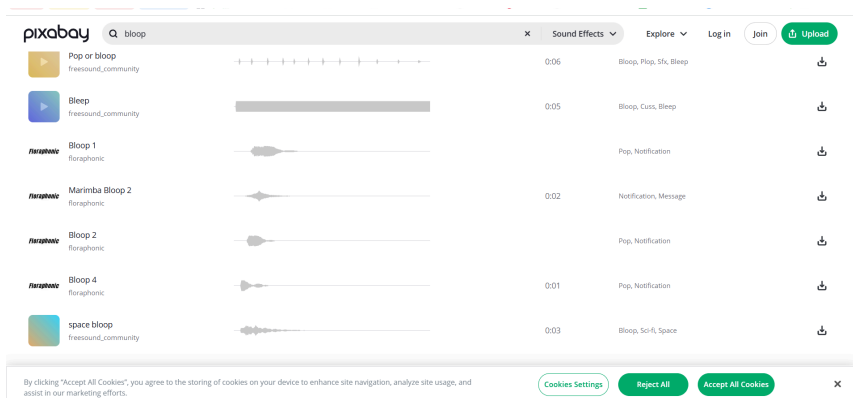


In the initial stage while playtesting the game, I felt like it was too hard to press play and to rush to the livescreen to catch the bird as it fell. I did ask ChatGPT for help and it told me the solution was to use a boolean to stop at the front page unless a button is pressed. Besides that, I just adjusted the spawn rate and the pipe gap to make the game more playable.

The fun part was designing the game's UI/UX and elements. We first added the code to create the "gameover", "press to start" and "score number", it took some trial and error but it was relatively straightforward.



I made use of Aseprite to draw the background, seaweed pipe, and the jellyfish. I only drew two frames of jellyfish because I envisioned the jellyfish to change to image02 upon pressing the space button. Sue Tyne helped me code that part which I'm grateful for. The background image needed some tweaking because I forgot to check the size of the game canvas before I started drawing. Afterwards, I had inputted the image of the seaweed. The tricky part was resizing it accurately and flipping it without accidentally rotating it, it took me a few hours...



After playtesting it a few more times, I really wanted to make the game more interactive. So I added sound effects to the bouncing and Gameover scene. I got the sounds from Mixkit and Pixabay and used a variety of "bloop" sounds. I was also worried we had to

manually add images for every score number but luckily I found out that it was possible to just upload a font, the format was similar to adding sounds so it wasn't too difficult compared to programming the game mechanics which we had assistance in from the base code.



The final part was making the UI text consistent, I chose a font similar to the gameover image that Sue found, it was also quite straightforward to change the colours and make a shadow which we learnt in earlier lessons. I wanted to go one step ahead and try generating bubbles from scratch, it was a bit tricky and I had to consult ChatGPT only to realise I couldn't use spawnrate again, I had to create a new bubblerate variable.

All in all, this project was really fun and doable thanks to a lot of assistance in the code. I still find Js a lot more straightforward and beginner friendly compared to C# so this project made me really happy to work on. If given a choice between pair work and individual work, I feel like I would like to try doing a project myself in the next round. While doing a project in pairs allowed me to push myself towards a higher standard, I felt like the project could have been done faster if I did it myself. When it comes to developing a game, since programming is not a very collaborative work, I think it would be difficult to work in pairs when there is a different workstyle within the group