Design

When starting out to design my game I had already used assets for my game proposal, so I decided to keep using them. They were cute 8x8 tiles from Itch.io, a website used by many indie game developers to show off & share their games and assets. These were free assets from Pico8, a free virtual machine & game engine I had played games on before.

I knew I wanted to use the provided tilesets, images made up of many small images known as “tiles” which are very often used to build entire games’ levels since they make it both more efficient when designing and easier to manage in the long run. I wanted to use Tiled, a free map editor designed to use tilesets in any way you could possibly want, since you can simply “draw” your map out as opposed to individually placing every single tile.

I started out designing a testing map since I would need to make sure I knew how it all worked before I went about building levels, or bits of levels, as I had originally planned. This was relatively simple, import tilesets, embed them to the map file, create the tile layers I’d be using, and get placing tiles. I then exported this as a .JSON file which is the format that Phaser supports tilemaps in and made sure every file I used was in the assets folder.

Then got to importing the tilemaps, the sprites, background images, tilesets, etc.

Once this was all done, I used our previous Software Project labs to figure out the syntax on how to get the basics made; the controllable player drawn on the canvas, the enemies, and collision detection. I then referenced the Phaser documentation to figure out how to use my tilemap and add collision to the solid platforms layer of my map. I had also placed items around the tilemap on a different layer, and a background. I assumed if I assigned a property to the individual tiles in the tilesets I was using in Tiled, I would be able to easily access them in Phaser and work them into the game that way. But after many an hour going through the documentation and solutions online, I gave up. I quickly enough figured out how to make an enemy, how to make it follow me once I got close, how to have it kill me when it’s bounding box came into contact with the character, etc. But using the properties I had set in Tiled proved to be a lot more difficult than I had originally expected. I decided I would add each item in later as individual sprites, but it was hard to bring myself to do it after spending so long trying to figure out how to use the tilemap items, so I instead focused on a more worrying problem..

After playing around in my testing map with the enemy for a while, I noticed something strange. The slime enemies I had added would throw incorrect collision detection events when trying to see if I had landed on a slime’s head or not as I wished to kill them when this happened.

Implementation & Testing

Requirements

Project Management