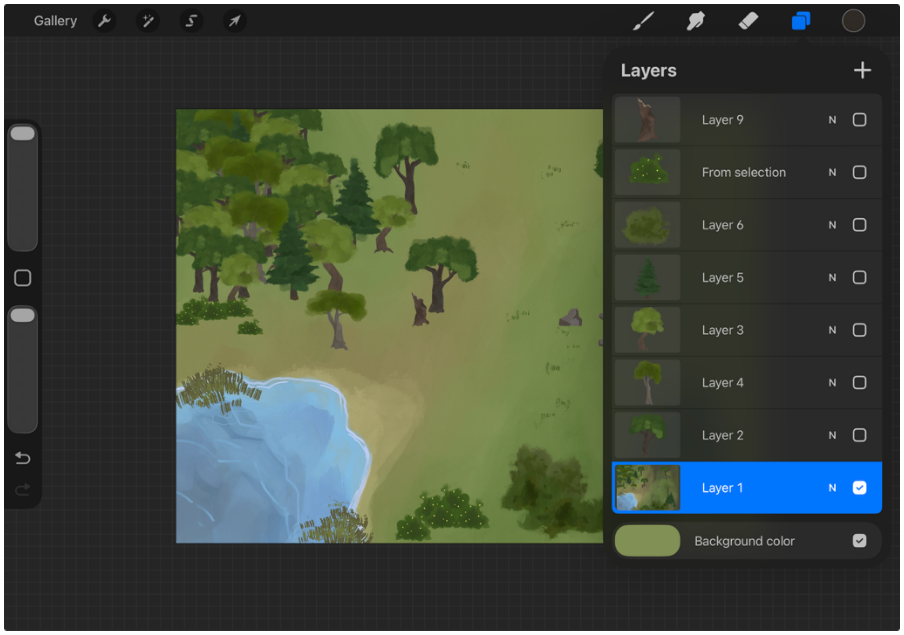
Process & Decision Documentation

Project/Assignment Decisions

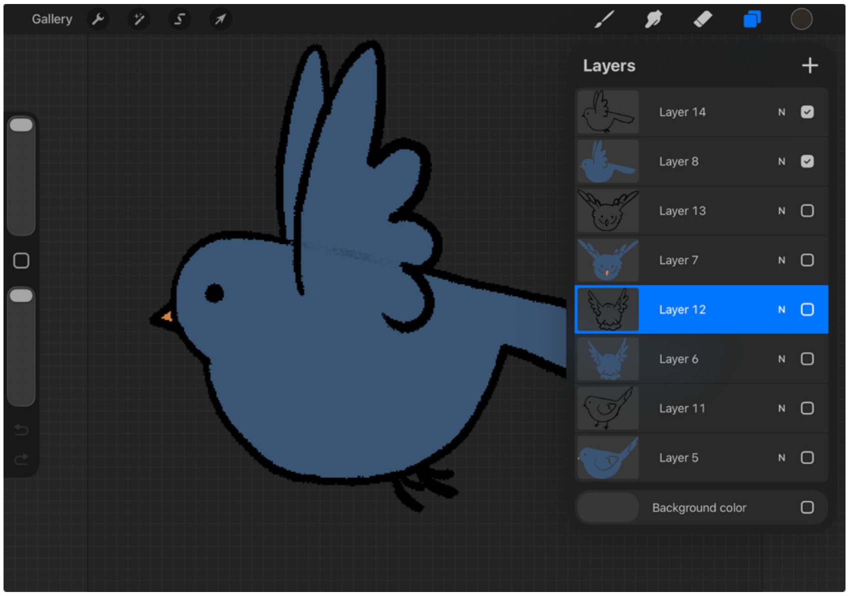
Side Quests and A4 (Individual Work)

One significant change I made was changing the canvas size to be bigger (it was originally a 200x200 square) so that the player doesn't have to squint to look at it. I also added the ability to speed up using shift because in case the player wants to fly faster.

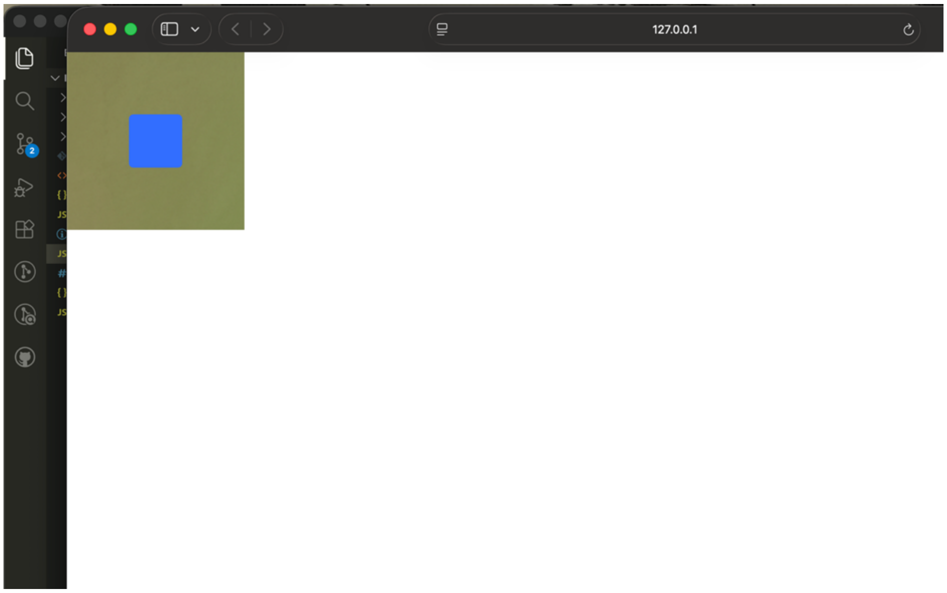
Role-Based Process Evidence



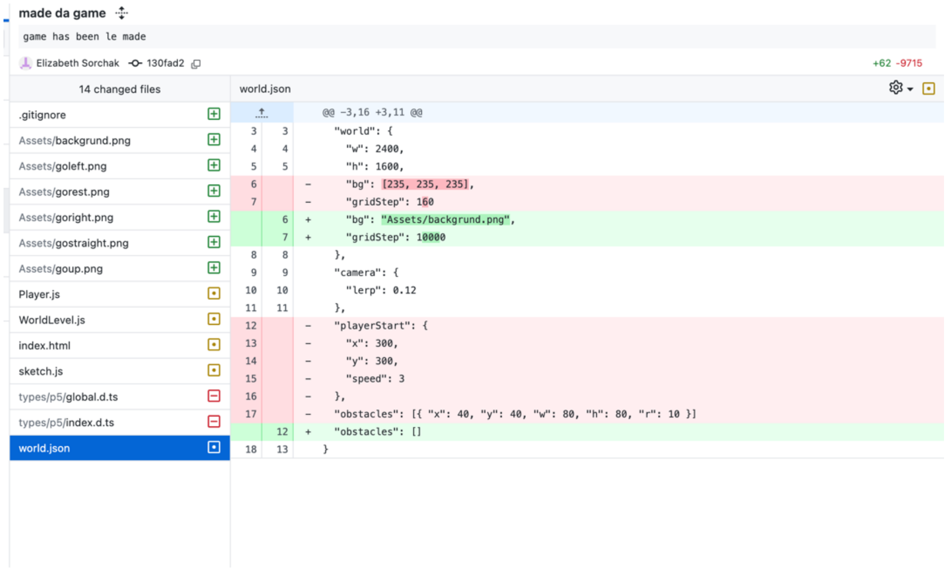
Screenshot of background in my drawing app. Layers with assets I drew are visible.



Screenshot of bird sprite in my drawing app, with all the different states visible in the layers.



Early version of the game when I had finally got the background to work.



made da game

```
game has been le made
```

Elizabeth Sorchak 130fad2

+62 -9715

14 changed files

- .gitignore
- Assets/backgrund.png
- Assets/goleft.png
- Assets/gorest.png
- Assets/goright.png
- Assets/gostraight.png
- Assets/goup.png
- Player.js
- WorldLevel.js
- index.html
- sketch.js
- types/p5/global.d.ts
- types/p5/index.d.ts
- world.json

world.json

```diff
@@ -3,16 +3,11 @@
    "world": {
      "w": 2400,
      "h": 1600,
-     "bg": [235, 235, 235],
-     "gridStep": 160
+     "bg": "Assets/backgrund.png",
+     "gridStep": 10000
    },
    "camera": {
      "lerp": 0.12
    },
-   "playerStart": {
-     "x": 300,
-     "y": 300,
-     "speed": 3
-   },
-   "obstacles": [{ "x": 40, "y": 40, "w": 80, "h": 80, "r": 10 }]
+   "obstacles": []
  }
```

Commit in GitHub with code differences.

Entry Header

Name: Liza Storchak

Role(s): SideQuest 5 creator

Primary responsibility for this work: creating SideQuest 5

Goal of Work Session

My goal was to create a peaceful explorer game showing player view vs world view schema mechanics.

Tools, Resources, or Inputs Used: I used the built in GenAI coder in Visual Studio Code to code (Claude Haiku 4.5 I believe)

GenAI Documentation

If GenAI was used (keep each response as brief as possible):

Date Used: February 13 2026

Tool Disclosure: Claude Haiku 4.5

Purpose of Use: Coding and debugging

Summary of Interaction: I used the GenAI to code the game and make changes in files (ie changing background and sizes), and also to debug.

Human Decision Point(s): I went through several iterations of the game at the beginning as I was trying to figure out how to replace the background. I also made the concept of the game and made all the assets.

Integrity & Verification Note: I ran the game after every iteration to make sure it worked.

Scope of GenAI Use: GenAI did not contribute to the concept or ideas of the game, nor the creation of the assets.

Limitations or Misfires: GenAI struggled to figure out what I was asking it to do in the beginning and failed to correctly change the files to reflect what I wanted until I went in myself, saw what needed to be done, and provided more specific directions.

Summary of Process (Human + Tool)

I used GenAI to change the background and get the functions of the game working. It took a few tries, especially since I could not get the background to change initially, and when it finally did load in It was compressed into the 200x200 space. Once I had the background down it was smooth sailing and I did not encounter further issues.

Decision Points & Trade-offs

A key decision I made was making the background size bigger. Originally it was going to be 200x200, but after some playtesting I realized that it was too small and it was giving me a headache. I made it 300x300, 400x400, 500x500, then settled on 500x450 because it looked the best. I also included the option to make the bird faster last minute. I wasn't going to because it's supposed to be a calm game, but I was getting impatient moving around the map and added that in.

Verification & Judgement

I play tested after each iteration to catch errors and make sure everything was running smoothly.

Limitations, Dead Ends, or Open Questions

The AI struggled to get the sketch to run the background image. I had to iterate multiple times and get it to debug itself multiple times, during which it kept telling me nothing was wrong. I figured out myself what had to be done and directed it from there.

Appendix

Me: using the sketch as a bse, change the game so that the background is backgrund.png. keep the same context of the original game, which is that its using camx and camy and concepts are screen view vs world view- this means that when the game starts the player starts in the middle of the map and can only see a small, 200 by 200 square of it. Make it so that the player can press the arrow keys to move around on the map- for now, keep the square from the original game to do, but resize it if needed to fit in the screen view

I'll help you update the game to use the background image and add camera movement with arrow keys. Let me start by examining the current sketch to understand its structure.

Now I'll update the sketch to use the background image, set the camera view to 200x200, and adjust the player's starting position to be in the middle of the map.

Done! I've updated your game with the following changes:

1. **Set view to 200×200 pixels** - Changed VIEW_W and VIEW_H to 200, which creates the small viewport showing only a portion of the world
2. **Added background image** - Added preload of backgrund.png and display it in the draw loop
3. **Player starts in center of map** - The player now starts at the middle of the 2400×1600 world (level.w / 2, level.h / 2) instead of at a fixed position

4. **Camera follows player** - The existing camera system with cam lerp smoothly follows the player as they move

Arrow keys already work to move the player (the Player.js class already has arrow key input). The camera will stay centered on the player, showing only a 200×200 window of the world at a time.

The reset button (press **R**) will also return the player to the center of the map.

debug. when i run the game on live server its stil the original one.

The file is updated correctly. This is a **browser cache issue**. Your browser is serving the old cached version of the JavaScript files.

Try a **hard refresh**:

- **Windows/Linux**: Ctrl + Shift + R
- **Mac**: Cmd + Shift + R

If that doesn't work, try clearing your browser cache entirely or opening it in an incognito/private window. After the hard refresh, the 200×200 game view with the background image should load.

debug. the original map of the game is still running, instead of the udated version with backgrund.png

The file is definitely updated with the 200×200 view. This is almost certainly a **browser/live server cache issue**. Try these solutions in order:

1. **Close the live server completely** and restart it
2. **Clear your browser cache** entirely (or use incognito/private mode)
3. **Hard refresh multiple times** (Cmd+Shift+R on Mac, then try Cmd+R a few times)
4. **Clear the Live Server cache** - if you have the Live Server extension, check its settings or restart VS Code entirely

If none of that works, try opening your dev tools (F12), go to **Network tab**, and check if sketch.js is being served with the old cached version. If you see that, you can add a cache-breaker to the HTML:

Replace this line in index.html:

- `<script src="sketch.js"></script>`

With this:

- <script src="sketch.js?v=2"></script>

Would you like me to make that change to the HTML file?

i did all and its still not updated. check to see if its an issue wiht the json

Found it! The issue is in world.json. The playerStart is still set to the old fixed coordinates (300, 300). In your sketch code, it uses this line:

Since playerStart exists in the JSON, it uses those hardcoded values instead of calculating the center. I'll remove the playerStart property so the code falls back to centering the player:

Done! The issue was that playerStart in the JSON was hardcoded to (300, 300). I removed it so now your sketch code will use the fallback that centers the player at (level.w / 2, level.h / 2).

Hard refresh your browser again and it should work—you should see the 200×200 viewport with the player starting in the middle of the map.

remove the original map of the game, which is world. json and instead have the sketch start up with my map. make sure it doesnt break anything

I need to know what you want your new map to contain. The world.json file currently defines:

World dimensions (w, h) - currently 2400×1600

Background color (bg) - currently light gray

Grid step (gridStep) - spacing for the grid lines

Camera lerp speed - how smooth the camera follows

Obstacles - rectangular objects on the map

What would you like your new map to have? For example:

Different world size?

Different obstacles or obstacle positions?

Different grid spacing?

Something else?

Once you tell me what should be in your map, I'll update world.json to use it instead of the original.

the new map is just the backgrund.png image. im using images for everything in this game.

Done! I've updated world.json to:

Empty the obstacles array (removes the old square)

Set gridStep to 10000 (makes the grid lines invisible so they won't show)

Now your game will show just the backgrund.png image as the map background with the player and camera system working. Everything still works behind the scenes, so you can easily add new game elements when you're ready.

Hard refresh and it should display your background image without the original map elements.

in the json, change "bg": [235, 235, 235], to have the background be th backgrund.png image

Now I need to update WorldLevel.js to handle the image path instead of an RGB array:

Done! I've updated both files:

world.json - Changed "bg" from [235, 235, 235] (RGB color) to "Assets/backgrund.png" (image path)

WorldLevel.js - Updated drawWorld() to check if bg is an RGB array before using it. If it's a string (image path), it skips the colored rectangle drawing

The background image from sketch.js will display, and the obstacles and grid will still render on top. Everything stays compatible!

debug as to why the game runs with the original grey background instead of the asset background

Found the issues:

sketch.js was reverted - It no longer loads or draws the bgImage

playerStart is hardcoded again to (300, 300) instead of centering

level.drawBackground() calls background(220), which is the grey color you're seeing

Let me fix all of these:

Fixed! I've restored and corrected sketch.js:

Added back bgImage variable and preload

Fixed player starting position to center on map (level.w / 2, level.h / 2)

Removed the grey level.drawBackground() call and replaced it with drawing the actual background image

Updated reset button to also center the player

The issue was that sketch.js had been reverted to an older version. Now it will:

Load backgrund.png in preload

Display the image as the viewport background

Start the player in the center of the map

Camera follows the player around

Hard refresh your browser and it should show the asset background image now!

instead of having the background be the whole image, instead use world view vs player view concepts so the background map is enlarged and the player only sees the 200x200 square of it on their screens- this way they can explore it by walking around

Done! I've updated the system to use world view vs player view:

How it works now:

backgrund.png is loaded and drawn at world coordinates (0, 0) to (2400, 1600)

The camera system offsets the view based on player position

The 200×200 viewport only shows a small window of the full background image

As the player moves with arrow keys, the camera follows and reveals different parts of the map

The background image itself is now the explorable world map at full size. When you hard refresh, the player will start in the center of the map and can walk around (using arrow keys) to explore different areas of the background image within the 200×200 window.

replace the blue square with gorest.png. when arrow keys are pressed, this "idle" sprite is replaced with the corresponding sprites: right arrow key: goright.png, left arrow key: goleft.png, up arrow key: goup.png, down arrow key: gostraight.png. resize the sprites so that they are roughly the same size as the blue cube right now. when no arrow keys are being pressed the sprite goes back to gorest.png

Done! I've updated the player system to use directional sprites:

Changes made:

sketch.js - Added sprite loading in preload:

gorest.png - idle/rest state

goright.png - right arrow key

goleft.png - left arrow key

goup.png - up arrow key

gostraight.png - down arrow key

Passes all sprites to the Player class

Player.js - Updated to:

Accept sprites in the constructor

Track current direction based on arrow keys being pressed

Display the appropriate sprite (24×24, same size as the original blue square)

Return to gorest.png when no keys are pressed

When you hard refresh, the blue square will be replaced with the gorest.png sprite. As you press arrow keys, it will swap to the corresponding directional sprites, and return to rest when you release them.

resize the sprite to be 50% bigger

Done! I've increased the sprite size from 24 to 36 pixels (50% bigger). Hard refresh and the player sprite will be noticeably larger.

let the player speed up slightly when they press shift

Done! I've added a speed boost when Shift is pressed. The player will move 1.5x faster (50% speed increase) when holding Shift while moving.

When you hard refresh and play, try holding Shift while pressing arrow keys and you'll notice the player moves noticeably faster.