

# Process & Decision Documentation

## Project/Assignment Decisions

### Side Quests and A4 (Individual Work)

I made the decision to adjust the lerp value to increase the smoothness and slowness of the camera movement and effectively evoke the feeling of relaxation and meditation. To do this, I started off by increasing the lerp value in the JSON file from 0.12 to 0.5 to see what effect it would have on the camera movement. I opened the sketch in my browser and noticed that it had done the opposite of what I intended; the camera was moving quicker and more sharply. To fix this, I went back to my code and adjusted the lerp value to 0.03 instead, which produced the delayed effect I wanted.

#### *GenAI Documentation*

**Date Used:** Thursday, February 19, 2026 and Friday, February 20, 2026

**Tool Disclosure:** ChatGPT 5.2

**Purpose of Use:** I used GenAI to generate the image of clouds for the background world and to generate the image of the dove for the player. I also used it to help me write code for replacing the existing background and player while ensuring that the program could still function properly.

**Summary of Interaction:** The tool was able to produce an image of clouds. Then it helped me understand what parts of the code needed to be changed so that the background could effectively be replaced with the new image. Afterwards, the GenAI produced an image of a dove to replace the current player, and it helped me change lines of code in the player constructor and in the sketch so that it would work properly.

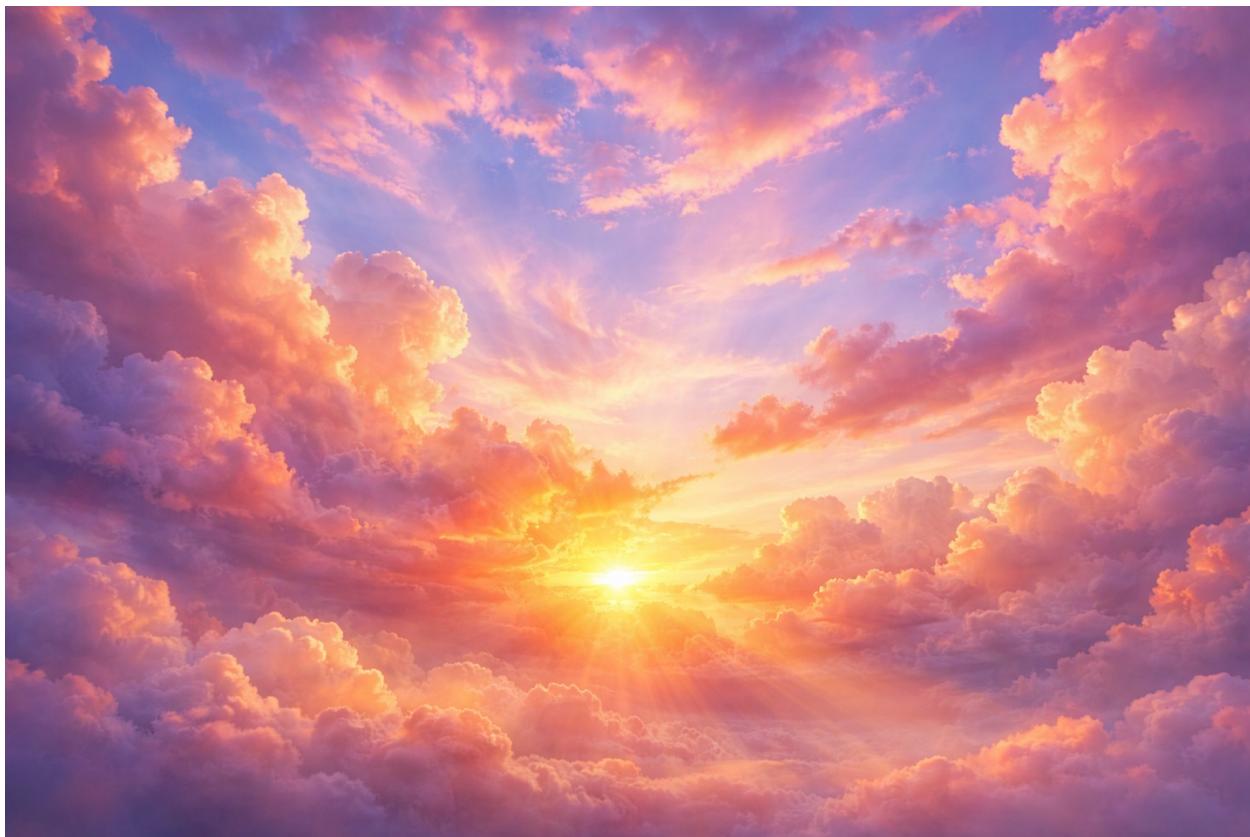
**Human Decision Point(s):** When changing the background to the new image, I used the tips that the GenAI provided, but it still wasn't working; the game was functioning properly, but it now looked the same as it did in the beginning. I looked at the code myself and noticed that I had to remove the "this.bg" line from the world level constructor and the "drawBackground" function entirely, since a background was no longer required with the image being loaded overtop. After I did this, the project displayed the image as the background with all the original functions intact.

**Scope of GenAI Use:** GenAI contributed to image generation and writing code for replacing the existing background and player. GenAI did not contribute to the concept/idea for the project or for adjusting text size, colour, and font.

**Limitations or Misfires:** When I asked the engine for help with replacing the background, it didn't account for the fact that the drawBackground function was preventing the new background image from displaying; I had to figure out and fix this mistake manually.

## Appendix

generate a 2400w by 1600h image of a dreamy, meditative, cloudy sky with sunset colours



WorldLevel.js

JavaScript

sketch.js

JavaScript

why is the background image not loading? the viewport shows a blank screen

Your viewport is blank because your **background image is never actually loading into the sketch that's running**.

Here's the issue 

---



## Problem #1 (Most Likely): You Have TWO `preload()` Functions

You have:

**In** `WorldLevel.js`

```
let backgroundImg;

function preload() {
  backgroundImg = loadImage("Sunset Background.png");
}
```

**In** `sketch.js`

```
function preload() {
  worldData = loadJSON("world.json");
}
```

 In p5.js, **only one `preload()` function can exist globally**.

The second one overrides the first one.

So your image never loads → `backgroundImg` is `undefined` → nothing draws → blank screen.

---



## Fix It Properly

Move the image loading into `sketch.js` and remove the `preload()` from `WorldLevel.js`.

---

### Step 1 — Remove this from `WorldLevel.js`

Delete:

```
let backgroundImg;

function preload() {
```

```
backgroundImg = loadImage("Sunset Background.png");  
}
```

---

## Step 2 — Add image loading to `sketch.js`

At the top of `sketch.js`:

```
let backgroundImg;
```

Inside `preload()`:

```
function preload() {  
    worldData = loadJSON("world.json");  
    backgroundImg = loadImage("Sunset Background.png");  
}
```

---

## Step 3 — Pass the image into the level

Update your `setup()`:

```
level = new WorldLevel(worldData, backgroundImg);
```

---

## Step 4 — Modify `WorldLevel` constructor

In `WorldLevel.js`:

```
class WorldLevel {  
    constructor(json, bgImg) {  
        this.backgroundImg = bgImg;
```

---

## Step 5 — Update `drawBackground()`

```
drawBackground() {  
    image(this.backgroundImg, 0, 0, this.w, this.h);  
}
```

---



## Problem #2 (Also Important)

You're drawing this after:

```
level.drawBackground();
```

```
push();  
translate(-camX, -camY);  
level.drawWorld();
```

But your background isn't inside the camera translation.

So it might not behave how you expect.

If you want the background to move with the world:

Move it inside the push/translate block:

```
push();  
translate(-camX, -camY);  
level.drawBackground();  
level.drawWorld();  
player.draw();  
pop();
```

---



## Problem #3 (File Name)

Make sure:

- The file is exactly named:
  - Sunset Background.png
  - It's in the **same folder as your sketch**
  - No extra spaces
  - Case sensitive (important!)
- 



## Why the Viewport Is Blank

When `backgroundImg` is undefined:

```
image(backgroundImg, 0, 0);
```

p5 tries to draw something that doesn't exist.

Sometimes it fails silently.

Sometimes it breaks rendering entirely.

---

## 🎯 Final Clean Version (Safe Structure)

Your sketch should have:

```
sketch.js  
WorldLevel.js  
Player.js  
world.json  
Sunset Background.png  
index.html
```

And **only one preload function — in sketch.js.**

---

If you'd like, paste your folder structure or console error message and I'll pinpoint it exactly in 10 seconds.

background image is in a file called assets

Perfect — that explains it 👍

Right now you're loading it like this:

```
backgroundImg = loadImage("Sunset Background.png");
```

But since it's inside an **assets folder**, p5 can't find it.

---

## ✅ Correct Path

Change it to:

```
backgroundImg = loadImage("assets/Sunset Background.png");
```

Path format is:

```
"folderName/fileName"
```

---



## Your Project Should Look Like This

```
project-folder/
├── index.html
├── sketch.js
├── WorldLevel.js
└── world.json
    └── assets/
        └── Sunset Background.png
```

---



## If It's STILL Blank

Open your browser console (Right click → Inspect → Console).

If you see something like:

```
Failed to load resource: 404
```

That means:

- The file name is slightly different
- There's a capitalization mismatch
- There's an extra space

For example:

✗ "Sunset background.png"  
✗ "sunset background.png"  
✗ "SunsetBackground.png"

Must match **exactly**.

---



## One More Important Thing

If you're running this by double-clicking `index.html` (`file://`), images sometimes fail to load.

You should run it using:

- VS Code Live Server

- Or a local server

If you're in GBDA302, you're probably using Live Server already.

---

If it's still blank after updating the path, tell me:

- Are you getting a console error?
- Is the canvas showing but just white?
- Or is nothing rendering at all?

We'll fix it fast.

generate a cartoon image of a dove with a transparent background(png)



make it so the dove is flying with its wings spread out



Player.js

JavaScript

I want to replace the player, which is currently a small blue square, into an image of the same size

You're actually **very close already** ☺

Your `draw()` function is trying to draw an image:

```
image(this.dove, this.x - 12, this.y - 12, 24, 24, 5);
```

But the problem is:

- 👉 `this.dove` is never defined
- 👉 So nothing is being drawn

---

✓ Here's Exactly How To Replace the Blue Square With an Image

We'll do this cleanly and properly.

---

## STEP 1 — Load the Player Image

In `sketch.js`, at the top:

```
let playerImg;
```

Inside your `preload()`:

```
function preload() {
  worldData = loadJSON("world.json");
  backgroundImg = loadImage("assets/Sunset Background.png");
  playerImg = loadImage("assets/dove.png"); // ← your dove file
}
```

Make sure:

`assets/dove.png`

exists and matches exactly.

---

## STEP 2 — Pass Image Into Player

In `setup()` change this:

```
player = new Player(start.x, start.y, start.speed);
```

To:

```
player = new Player(start.x, start.y, start.speed, playerImg);
```

---

## STEP 3 — Update Player Constructor

In `Player.js`, modify constructor:

```
class Player {
  constructor(x, y, speed, img) {
    this.x = x;
    this.y = y;
    this.s = speed ?? 3;
    this.img = img; // store image
  }
}
```

---

## STEP 4 — Fix draw()

Replace your current draw function with:

```
draw() {
    image(this.img, this.x - 12, this.y - 12, 24, 24);
}
```

(Remove that extra 5 at the end — that parameter doesn't belong there.)

---

### 🎯 Why `this.x - 12`?

Your player is centered on `(x, y)`.

Since the image is 24x24:

$$24 / 2 = 12$$

So subtracting 12 centers it.

---

### 📦 Final Player.js Should Look Like This

```
class Player {
  constructor(x, y, speed, img) {
    this.x = x;
    this.y = y;
    this.s = speed ?? 3;
    this.img = img;
  }

  updateInput() {
    const dx =
      (keyIsDown(RIGHT_ARROW) || keyIsDown(68)) -
      (keyIsDown(LEFT_ARROW) || keyIsDown(65));

    const dy =
      (keyIsDown(DOWN_ARROW) || keyIsDown(83)) -
      (keyIsDown(UP_ARROW) || keyIsDown(87));

    const len = max(1, abs(dx) + abs(dy));
    this.x += (dx / len) * this.s;
    this.y += (dy / len) * this.s;
  }
}
```

```
    draw() {
        image(this.img, this.x - 12, this.y - 12, 24, 24);
    }
}
```

---

## If It Shows Nothing

Open console.

If you see:

```
Cannot read properties of undefined (reading 'width')
```

It means image didn't load (wrong path).

---

## Optional (Cleaner Way)

If you want size to be adjustable:

Add:

```
this.size = 24;
```

Then:

```
image(this.img, this.x - this.size/2, this.y - this.size/2, this.size,
this.size);
```

---

If you want next level polish, I can show you how to:

- Rotate the dove based on movement direction
- Animate flapping wings
- Add collision hitbox separate from sprite

Just say the word 

it didn't work

Good — that means something specific is failing. Let's debug this properly instead of guessing.

First, I need to narrow it down:

### What exactly is happening?

1. Is the blue square still showing?
2. Is nothing showing at all?
3. Are you getting a red error in the browser console?

Open **Inspect → Console** and tell me what it says.

---

## Most Common Reasons This Fails

### 1. You Forgot to Update Player Constructor Everywhere

If you changed:

```
constructor(x, y, speed, img)
```

But somewhere you're still calling:

```
new Player(start.x, start.y, start.speed);
```

Then `this.img` will be `undefined`.

Double-check in `setup()` that it is EXACTLY:

```
player = new Player(start.x, start.y, start.speed, playerImg);
```

---

### 2. Image Path Is Wrong

If your dove is inside assets, it must be:

```
playerImg = loadImage("assets/dove.png");
```

Things that break it:

- Dove.png vs dove.png
- assets/dove.PNG
- Extra space in filename

File names are case sensitive.

---

### 3. Your Player.js Still Says `this.dove`

Your file earlier had this:

```
image(this.dove, ...)
```

If you didn't replace it with:

```
image(this.img, ...)
```

It will fail.

---

### 4. The Image Is Drawing Offscreen

Try temporarily replacing your draw() with:

```
draw() {
  fill(255,0,0);
  rect(this.x-12, this.y-12, 24, 24);
}
```

If that shows up, your movement is fine.

If that doesn't show, the issue isn't the image — it's something else.

---

### Quick Test (Fastest Way To Know)

Inside draw() of Player, temporarily do:

```
draw() {
  if (!this.img) {
    fill(255,0,0);
    rect(this.x-12, this.y-12, 24, 24);
    return;
  }

  image(this.img, this.x - 12, this.y - 12, 24, 24);
}
```

If you see a red square → image didn't load.

If you see nothing → player isn't drawing at all.

If you see the dove → it works.

---

Reply with:

- What you see
- Any console error message

And I'll pinpoint the exact line causing it.

1536 × 1024 pixels is the image size of my dove, what size should i make height of width is 100 px to maintain aspect ratio

Great — this is just an aspect ratio calculation 🌟

Your image size is:

Width = 1536  
Height = 1024

---

## 🎯 Step 1: Find Aspect Ratio

Divide both by 512:

1536 / 512 = 3  
1024 / 512 = 2

So the aspect ratio is:

3 : 2

---

## 🎯 Step 2: Scale Width to 100px

If width = 100px, then:

height=100×(2/3)height = 100 × (2 / 3)height=100×(2/3) height=66.67pxheight = 66.67pxheight=66.67px

---

## Final Answer

If width = **100px**

Then height should be:

≈ 67px

---

## So in p5:

```
image(this.img, x, y, 100, 67);
```

Or more precise:

```
image(this.img, x, y, 100, 66.67);
```

---

## Pro Tip (Better Way — Automatic)

Instead of hardcoding:

```
let w = 100;
let h = w * (this.img.height / this.img.width);
image(this.img, x, y, w, h);
```

This way it always preserves aspect ratio even if you change the image later.

---

If you want it centered properly for your player, tell me and I'll show you the exact math for that too.