Process & Decision Documentation

Project/Assignment Decisions

I updated Example 4 with a new level, and 2 new collectibles: a Pink PowerUp that makes the player glow pink, and purple coins that give players 100 points.

GenAI Documentation

GenAI was used in this assignment.

Date Used: February 9th, 2026

Tool Disclosure: Claude Sonnet 4.5

Purpose of Use: I used it to explain how the code works and what exactly the side quest assignment was asking of me, and how I could possibly code my ideas (in a way that makes sense for someone who has not used Javascript in a long time) (e.g., brainstorming, debugging, summarization, wording support).

Summary of Interaction: The tool first briefly explained the components of the example code and how it worked together. Then it created a comprehensive guide on how I could implement my ideas for the assignment into the example code, and updated sketch.js, player.js, level.js, and levels.json.

Human Decision Point(s): It provided a very strong updated code, so I mainly adjusted the display text for clarity, and adjusted the new features to my liking with small tweaks (adjusting timing, colours, and fixing certain features). I also added a new 3rd level, adjusting the new level 2 the AI gave me but configured in an altered way.

Integrity & Verification Note: I ran the new GenAI code into my repository to check to see if the code worked, regardless of user error (such as going the wrong way within the level). This allowed me to see that the text was not well placed for example.

Scope of GenAI Use: The GenAI did most of the work except the HTML and Javascript text, the level 3 design, and resetting the pink glow effect after every level.

Limitations or Misfires: It struggled with the visual design elements that included colour or text placement. It did not look very appealing initially, as the purple coins had this bright gold in the middle that did not mesh well, and the text was overlapping with the level design.

Summary of Process (Human + Tool)

- Once I got the new code, I wanted a way for important information to still be displayed, but in a way where the text wasn't too small or too cramped together. After trying to make the text smaller in the Javascript, I moved the title and instructions to the HTML, keeping anything updated by the Javascript inside those spheres so there would be minimal conflict. I also spread out the Javascript text, placing the Glowing message to the right instead of its original left on top of the level (which also made it unclear). I made all the text a little bit smaller, and increased the contrast between the level and text colour so anyone could read it)
- I adjusted the colour of the purple coin to have a light purple center
- I wanted the condition of the player to be reset every level, normally the pink glow would last across levels but I added " `player.isGlowing = false;` " to the LoadGame function so it would reset each time. This made the feature something to keep chasing after every time
- However, it took a long time to realize how to change this feature. I kept adding "this.isGlowing" in files that were not player.js and I kept putting it within every if statement until I checked how the AI was connecting variables set with classes (since I have not used them in a long time and did not know how classes were best used)

Decision Points & Trade-offs

Describe one or two key decisions you made:

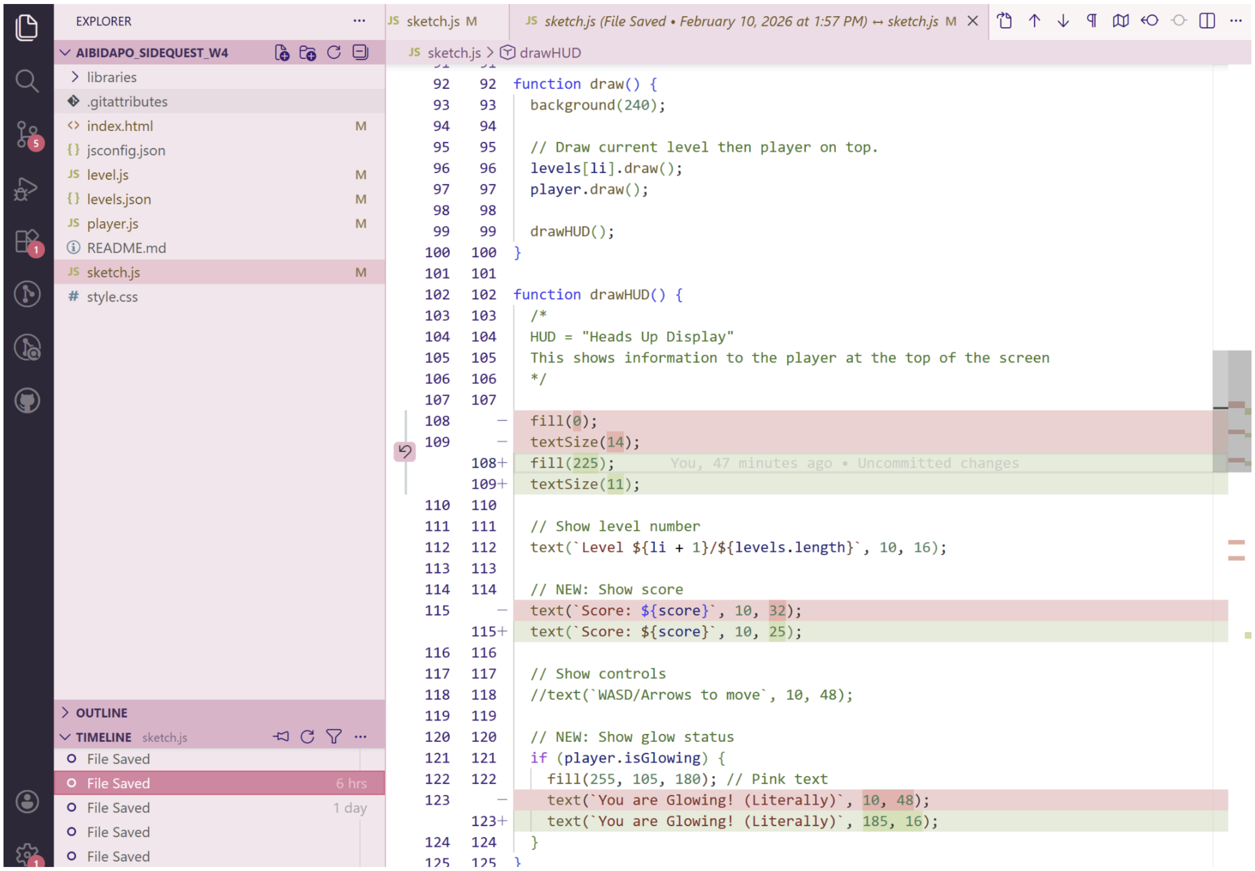
- I initially was trying to copy and paste the changes done by the AI into the original example code, however it resulted in too many errors that I did not have the time to debug
- I decided to make the glow last shorter and reset after every level. I made this choice since it made it seem like each level was its own stage, not something completely continuous like the glow lasting did

Verification & Judgement

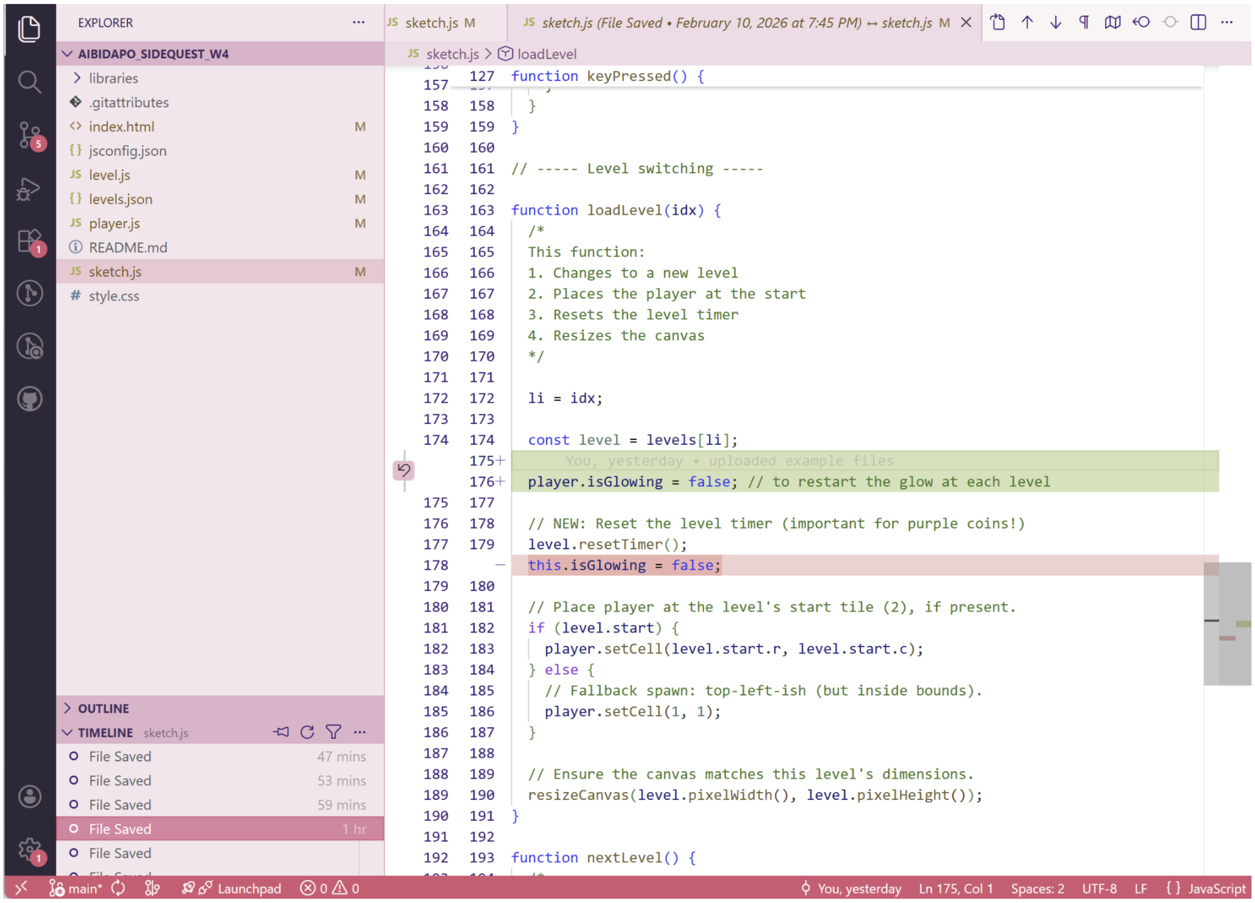
Explain how you evaluated whether your decision or change was appropriate:

- Kept playing it on my own and It felt a lot smoother of a user experience

Evidence of Human Editing:



```
function draw() {
  background(240);

  // Draw current level then player on top.
  levels[li].draw();
  player.draw();

  drawHUD();
}

function drawHUD() {
  /*
  HUD = "Heads Up Display"
  This shows information to the player at the top of the screen
  */

− fill(0);
− textSize(14);
+ fill(225);          You, 47 minutes ago • Uncommitted changes
+ textSize(11);

  // Show level number
  text(`Level ${li + 1}/${levels.length}`, 10, 16);

  // NEW: Show score
− text(`Score: ${score}`, 10, 32);
+ text(`Score: ${score}`, 10, 25);

  // Show controls
  //text(`WASD/Arrows to move`, 10, 48);

  // NEW: Show glow status
  if (player.isGlowing) {
    fill(255, 105, 180); // Pink text
−   text(`You are Glowing! (Literally)`, 10, 48);
+   text(`You are Glowing! (Literally)`, 185, 16);
  }
}
```



```
function keyPressed() {
  }
}

// ----- Level switching -----

function loadLevel(idx) {
  /*
  This function:
  1. Changes to a new level
  2. Places the player at the start
  3. Resets the level timer
  4. Resizes the canvas
  */

  li = idx;

  const level = levels[li];
      You, yesterday • uploaded example files
  player.isGlowing = false; // to restart the glow at each level

  // NEW: Reset the level timer (important for purple coins!)
  level.resetTimer();
  this.isGlowing = false;

  // Place player at the level's start tile (2), if present.
  if (level.start) {
    player.setCell(level.start.r, level.start.c);
  } else {
    // Fallback spawn: top-left-ish (but inside bounds).
    player.setCell(1, 1);
  }

  // Ensure the canvas matches this level's dimensions.
  resizeCanvas(level.pixelWidth(), level.pixelHeight());
}

function nextLevel() {
```