

Name: Cohen Peters

One change I made was adjusting the ball movement so that it slides until it hits a wall with each movement. I did this because it adds a different play style compared to normal maze puzzles.

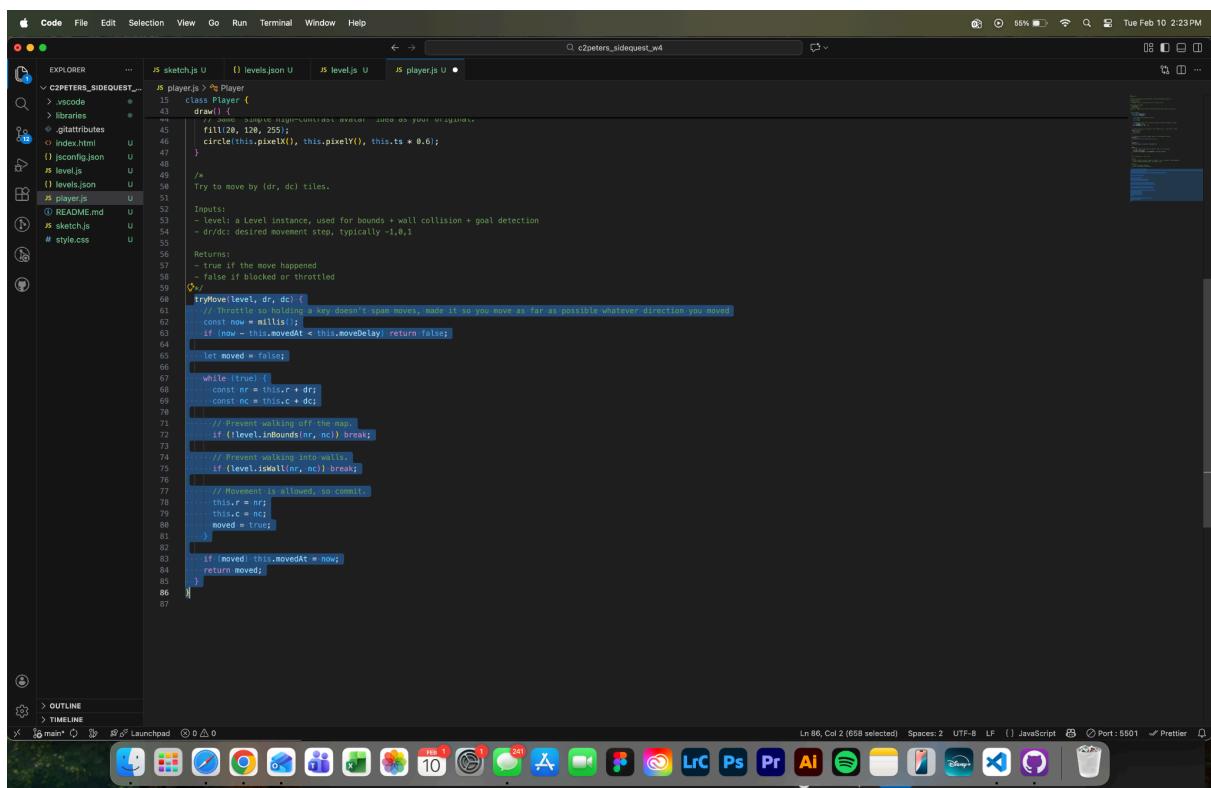
Goal of work session: The goal was to make a level based game using arrays or JSON data.

Tools used: VS code

AI documentation: No GenAI used for this task.

Reference: <https://p5js.org/reference/>

Working screenshots:



The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files in the project folder "C2PETERS_SIDEQUEST": sketch.js, player.js, index.html, jsconfig.json, level.js, levels.json, player.js (another instance), README.md, sketch.js, and style.css.
- Code Editor:** The active file is "player.js". The code defines a "Player" class with a "draw" method and a "tryMove" method. The "tryMove" method takes inputs (dr, dc) and a level object, then performs a series of checks and movements to move the player while avoiding walls and respecting movement limits.
- Status Bar:** Shows the current file is "player.js", line 86, column 2, with 658 selected characters, spaces: 2, and encoding: UTF-8.
- Bottom Taskbar:** Shows various application icons including Finder, Mail, Safari, Calendar, Reminders, Stocks, Wallet, App Store, iTunes Store, iBooks, Photos, System Preferences, Launchpad, and others.

```
Code File Edit Selection View Go Run Terminal Window Help
c2peters_sidequest_w4
EXPLORER JS sketch.js (1) levels.json U JS levels.ts U
VS Code > vscode
> libraries
gitattributes index.html
sconfig.json level.js
level.ts player.js
README.md
sketch.js style.css

function setup() {
    /* Convert raw JSON grids into Level objects.
    LevelsData.levels is an array of 20 arrays.
    */
    levels = LevelsData.levels.map((grid) => new Level(copyGrid(grid), TS));
}

// Create a player.
player = new Player(TS);

// Load the first level (sets player start + canvas size).
loadLevel(0);

noStroke();
textFont("sans-serif");
textSize(14);

}

function draw() {
background(240);

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

function drawHUD() {
// Web matches your original idea: show level count and controls.
// I changed the colour of the font to white so it is easier to read
fill(255);
text(`Level ${li + 1}/ ${levels.length} - WASD/Arrows to move`, 10, 16);
}

}

function keyPressed() {
/*
Convert key presses into a movement direction. (WASD + arrows)
*/
let dr = 0;
let dc = 0;

if (keyCode === LEFT_ARROW || key === "a" || key === "A") dc = -1;
else if (keyCode === RIGHT_ARROW || key === "d" || key === "D") dc = 1;
else if (keyCode === UP_ARROW || key === "w" || key === "W") dr = -1;
else if (keyCode === DOWN_ARROW || key === "s" || key === "S") dr = 1;
else return; // not a movement key

// Try to move, if blocked, nothing happens.
const moved = player.tryMove(levels[li], dr, dc);

// If the player moved onto a goal tile, advance levels.
if (moved && levels[li].isGoal(player.r, player.c)) {
nextLevel();
}
}

Ln 70, Col 13 (80 selected) Spaces: 2 UTF-8 LF ( ) JavaScript ⚡ Port: 8501 ✨ Prettier
OUTLINE
TIMELINE
main.ts Launchpad 0 0 0 0
```