

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.

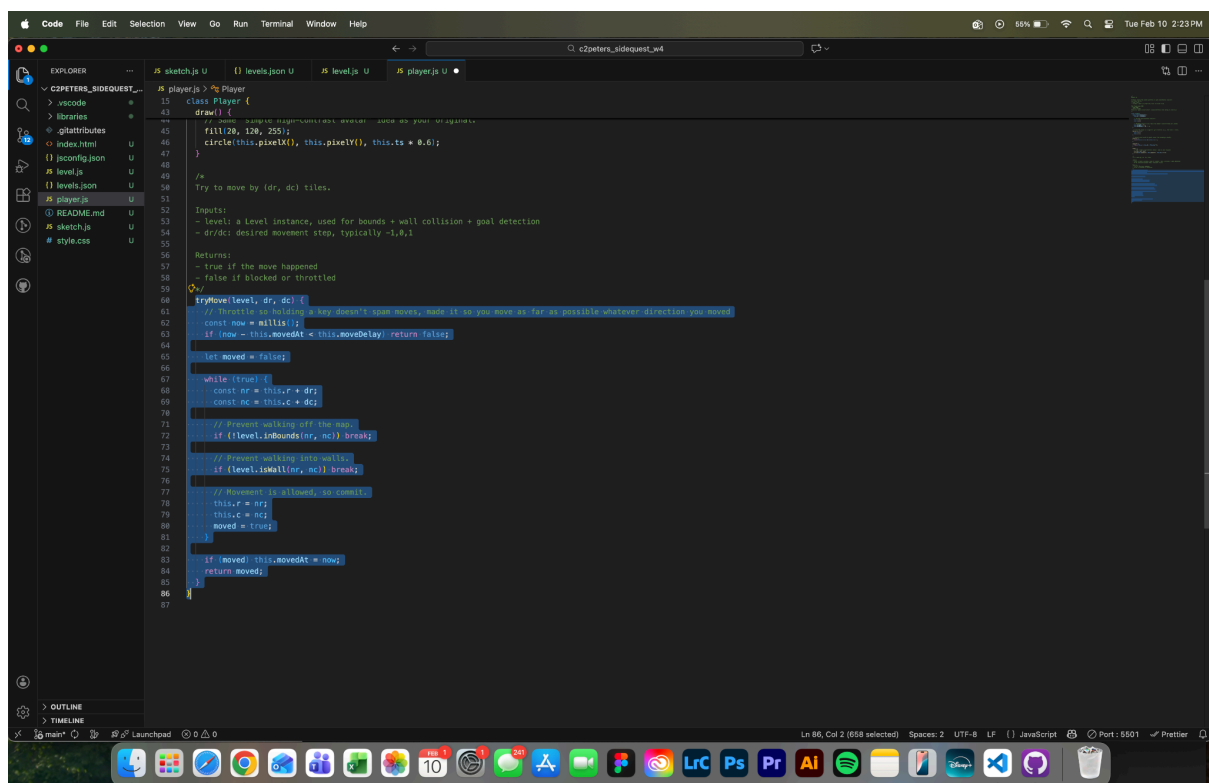
Goal of work session: The goal was to make a level based game using arrays or JSON data.

Tools used: VS code

AI documentation: No GenAI used for this task.

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

Working screenshots:



```
15 class Player {
16   draw() {
17     fill(20, 120, 255);
18     circle(this.px, this.py, this.r * 0.6);
19   }
20   /*
21    * Try to move by (dr, dc) tiles.
22    *
23    * Inputs:
24    * - level: a Level instance, used for bounds + wall collision + goal detection
25    * - dr/dc: desired movement step, typically -1,0,1
26    *
27    * Returns:
28    * - true if the move happened
29    * - false if blocked or throttled
30    */
31   tryMove(level, dr, dc) {
32     // throttle: blocking a key doesn't "lose" moves. Make it so you move as far as possible whatever direction you move
33     const now = millis();
34     if (now - this.movedAt < this.moveDelay) return false;
35
36     let moved = false;
37
38     while (true) {
39       const nr = this.r + dr;
40       const nc = this.c + dc;
41
42       // Prevent walking off the map
43       if (!level.inBounds(nr, nc)) break;
44
45       // Prevent walking into walls
46       if (level.isWall(nr, nc)) break;
47
48       // Movement is allowed, so commit
49       this.r = nr;
50       this.c = nc;
51       moved = true;
52     }
53
54     if (moved) this.movedAt = now;
55     return moved;
56   }
57 }
```

```
1 {
2   "levels": [
3     [
4       [1, 1, 1, 1, 1, 1, 1, 1, 1],
5       [1, 1, 0, 0, 0, 0, 0, 0, 1],
6       [1, 2, 1, 1, 0, 1, 1, 3, 0, 1],
7       [1, 0, 1, 0, 0, 1, 0, 1, 0, 1],
8       [1, 0, 1, 0, 1, 1, 0, 1, 0, 1],
9       [1, 0, 0, 0, 0, 0, 0, 0, 1],
10      [1, 1, 1, 1, 1, 1, 1, 1, 1],
11    ],
12  ],
13  [
14    [1, 1, 1, 1, 1, 1, 1, 1, 1],
15    [1, 0, 0, 0, 1, 0, 0, 1, 1],
16    [1, 0, 1, 0, 1, 0, 1, 0, 1],
17    [1, 0, 0, 2, 1, 0, 0, 0, 1],
18    [1, 0, 0, 1, 1, 0, 1, 0, 1],
19    [1, 1, 0, 0, 0, 1, 1, 3, 0, 1],
20    [1, 1, 1, 1, 1, 1, 1, 1, 1],
21  ],
22  [
23    [1, 1, 1, 1, 1, 1, 1, 1, 1],
24    [1, 2, 0, 1, 0, 0, 0, 1, 0, 1],
25    [1, 0, 1, 0, 0, 1, 0, 1, 0, 1],
26    [1, 0, 1, 1, 0, 1, 1, 1, 0, 1],
27    [1, 0, 0, 1, 0, 0, 0, 1, 0, 1],
28    [1, 1, 0, 1, 1, 1, 0, 1, 0, 1],
29    [1, 0, 0, 0, 1, 0, 0, 0, 1],
30    [1, 0, 1, 1, 0, 1, 1, 1, 0, 1],
31    [1, 0, 1, 1, 0, 0, 0, 1, 0, 1],
32    [1, 1, 0, 1, 1, 1, 0, 1, 0, 1],
33    [1, 0, 0, 0, 1, 0, 0, 0, 1],
34    [1, 0, 1, 1, 0, 1, 0, 1, 3, 1],
35    [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
36  ],
37  ]
38 }
39
```

```
39 function setup() {
40   //
41   // Convert raw JSON grids into Level objects.
42   // levelsData.levels is an array of 2D arrays.
43   //
44   levels = levelsData.levels.map(grid => new Level(copyGrid(grid), TS));
45
46   // Create a player.
47   player = new Player(TS);
48
49   // Load the first level (sets player start + canvas size).
50   loadLevel(0);
51
52   noStroke();
53   textFont("sans-serif");
54   textSize(14);
55 }
56
57 function draw() {
58   background(240);
59
60   // Draw current level then player on top.
61   levels[0].draw();
62   player.draw();
63
64   drawHUD();
65 }
66
67 function drawHUD() {
68   // HUD matches your original idea: show level count and controls.
69   // change the color of the font to white so it is easier to read
70   fill(255);
71   text('Level ' + (l + 1) / S / levels.length + ' - WASD/Arrows to move', 10, 10);
72 }
73
74 function keyPressed() {
75   //
76   // Convert key presses into a movement direction. (WASD + arrows)
77   //
78   let dr = 0;
79   let dc = 0;
80
81   if (keyCode === LEFT_ARROW || key === "a" || key === "A") dc = -1;
82   else if (keyCode === RIGHT_ARROW || key === "d" || key === "D") dc = 1;
83   else if (keyCode === UP_ARROW || key === "w" || key === "W") dr = -1;
84   else if (keyCode === DOWN_ARROW || key === "s" || key === "S") dr = 1;
85   else return; // not a movement key
86
87   // Try to move. If blocked, nothing happens.
88   const moved = player.tryMove(levels[l], dr, dc);
89
90   // If the player moved onto a goal tile, advance levels.
91   if (moved && levels[l].isGoal(player.r, player.c)) {
92     nextLevel();
93   }
94 }
95
```