Process & Decision Documentation

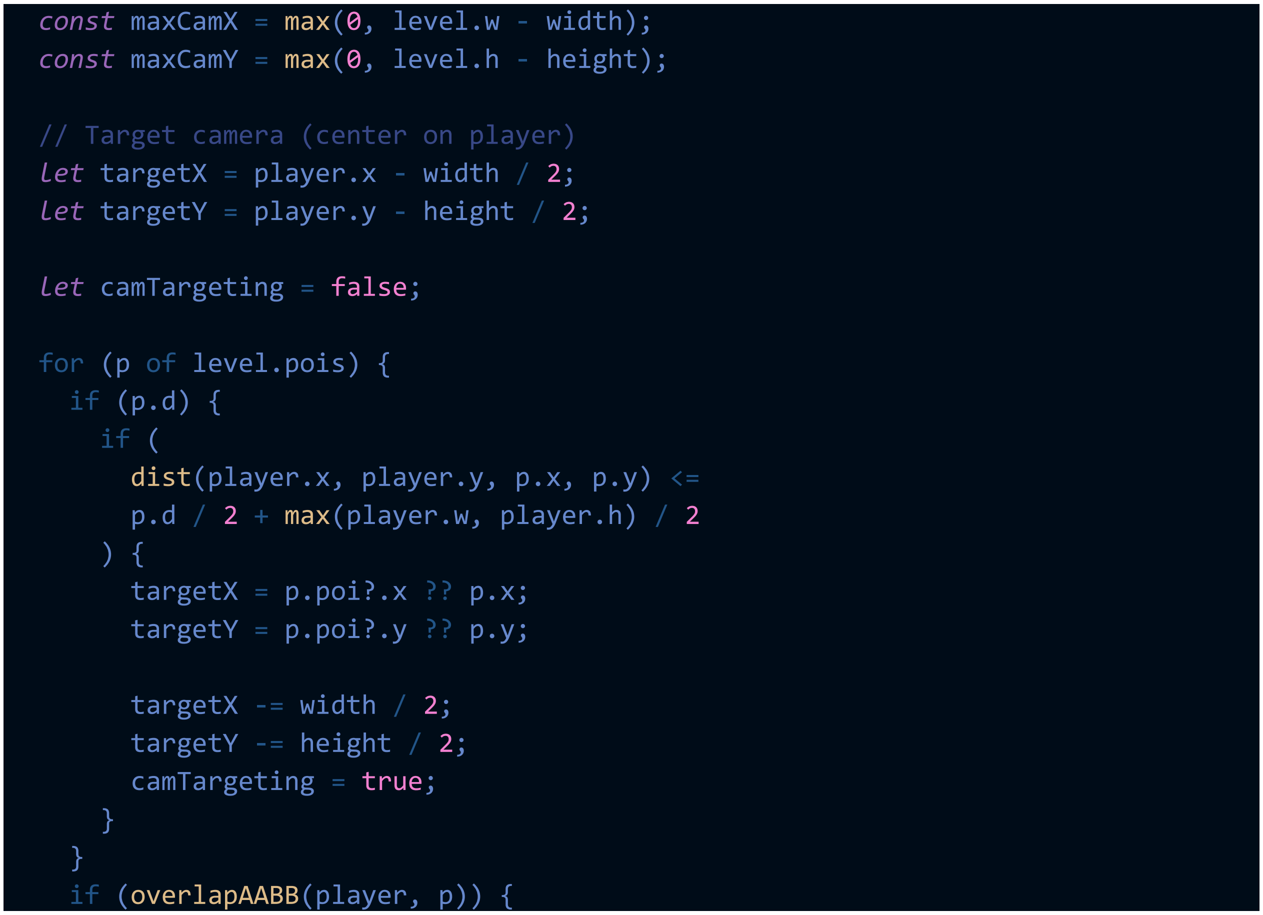
Project/Assignment Decisions

To evoke an emotion (excitement/wonder) from the player's movement and camera, I'm going to make it so that the camera lags farther behind the player's movement and returns slowly so it feels like the player cube is dragging you around the game world, excited for exploration but moving too fast to keep up with.

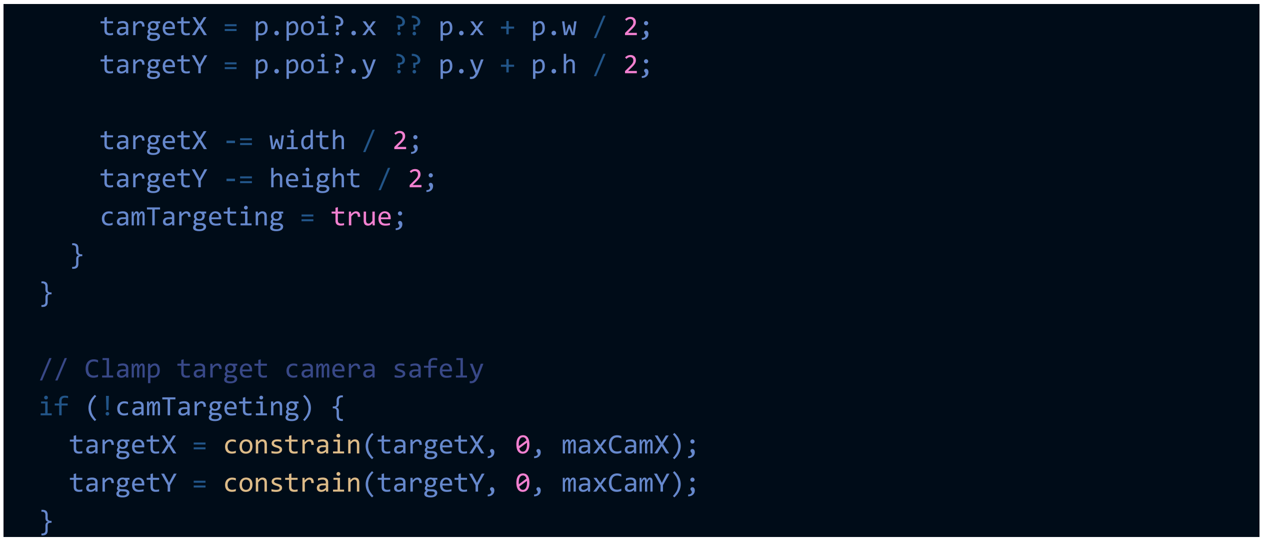
I am deciding to add a POI (point of interest) system to the example's camera. The way it will work is that when the player stands in a specific position in the game world, the camera will unlatch from it's default constraints and will move its target to something that is dictated by the "POI field" the player is within. This will help reinforce that "wonder" emotion by giving the player things to notice and find.

Process Evidence

Final Draft of POI camera logic:



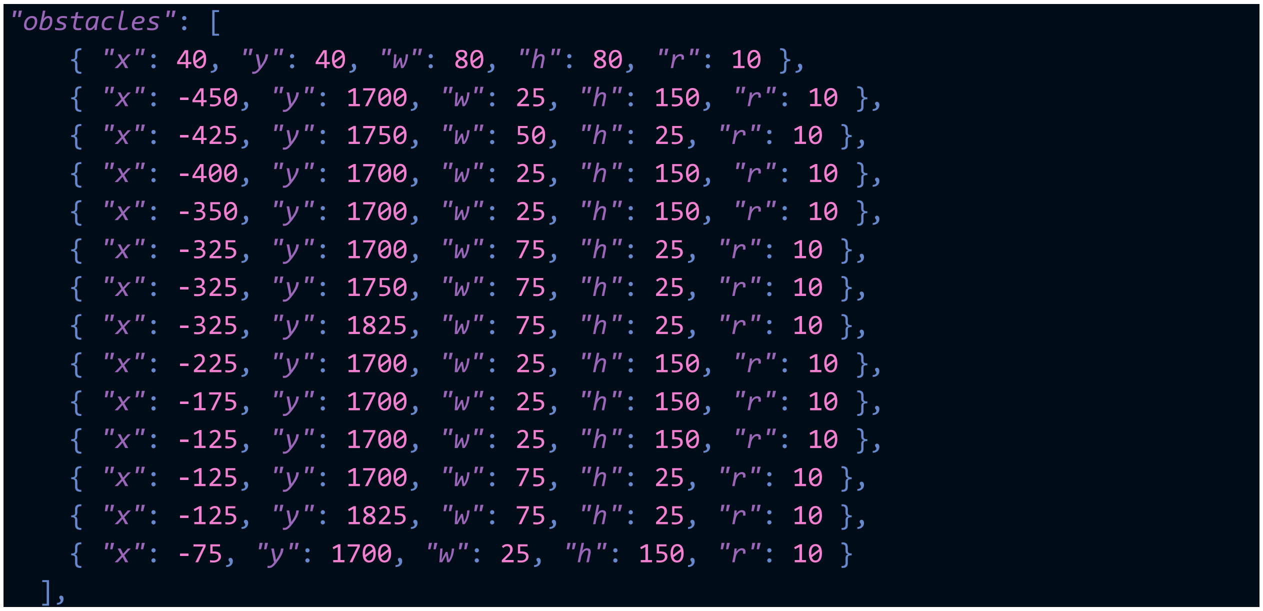
```
  const maxCamX = max(0, level.w - width);
  const maxCamY = max(0, level.h - height);

  // Target camera (center on player)
  let targetX = player.x - width / 2;
  let targetY = player.y - height / 2;

  let camTargeting = false;

  for (p of level.pois) {
    if (p.d) {
      if (
        dist(player.x, player.y, p.x, p.y) <=
        p.d / 2 + max(player.w, player.h) / 2
      ) {
        targetX = p.poi?.x ?? p.x;
        targetY = p.poi?.y ?? p.y;

        targetX -= width / 2;
        targetY -= height / 2;
        camTargeting = true;
      }
    }
    if (overlapAABB(player, p)) {
```



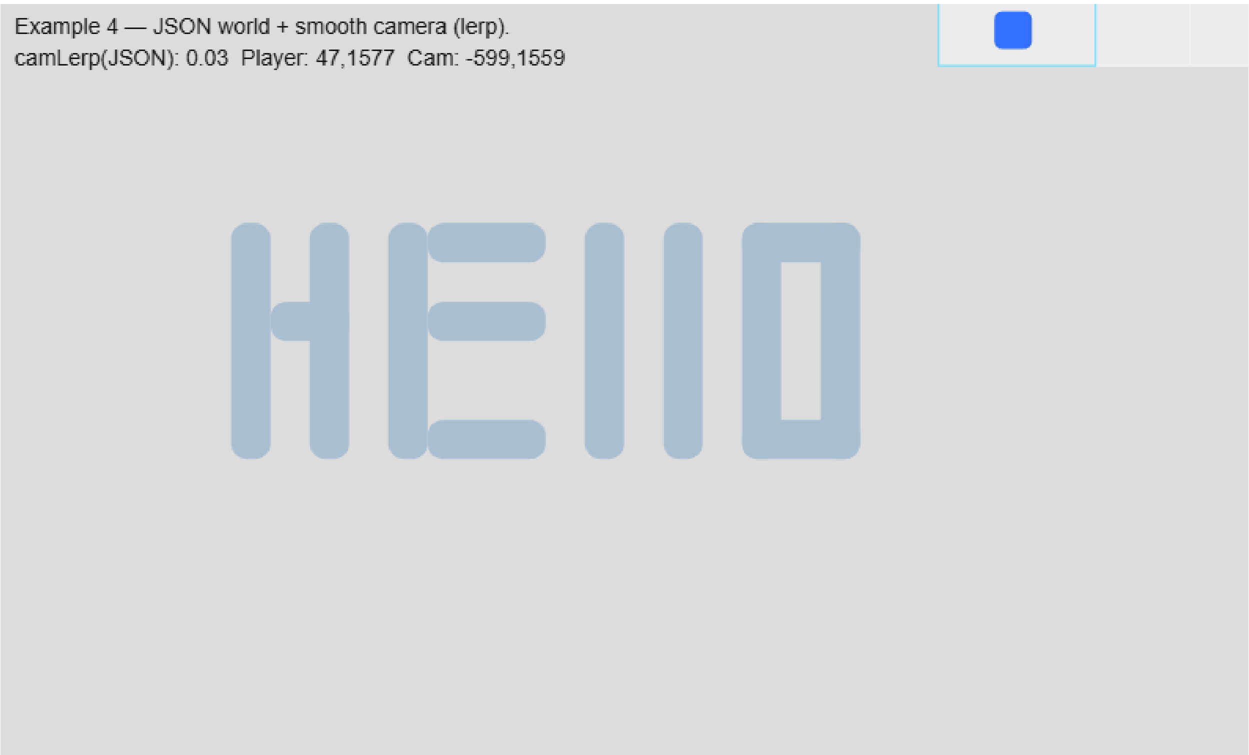
```
      targetX = p.poi?.x ?? p.x + p.w / 2;
      targetY = p.poi?.y ?? p.y + p.h / 2;

      targetX -= width / 2;
      targetY -= height / 2;
      camTargeting = true;
    }
  }

  // Clamp target camera safely
  if (!camTargeting) {
    targetX = constrain(targetX, 0, maxCamX);
    targetY = constrain(targetY, 0, maxCamY);
  }
```

In the world data each “POI” is treated as a circle or box depending on the given arguments. It also has an option “poi” property. When the player collides with the POI field (circle or box collider), the camera’s targetX and targetY become disconnected from the player and are now tied to either: the POI field’s centre (x, y OR x – w /2, y – h/2) or the optional “poi” property’s point (x, y). As well, under normal conditions the camera is unable to go beyond the world’s boundaries so to retain that function I added the camTargeting variable to easily keep track of when the player is inside a POI field and to unrestraint the camera’s movement accordingly.

First addition to the interest points in the game world (POI: -600, 1600):

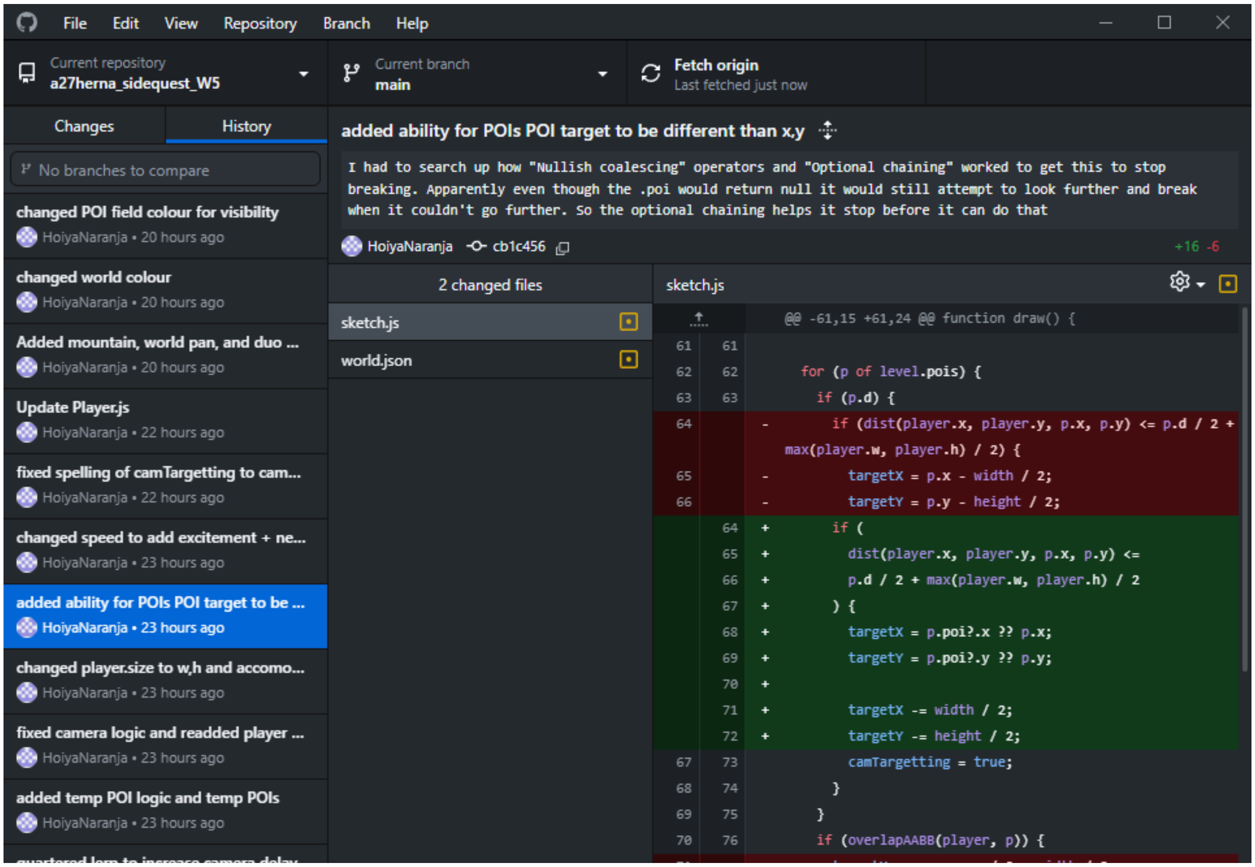


```
"obstacles": [
    { "x": 40, "y": 40, "w": 80, "h": 80, "r": 10 },
    { "x": -450, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -425, "y": 1750, "w": 50, "h": 25, "r": 10 },
    { "x": -400, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -350, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -325, "y": 1700, "w": 75, "h": 25, "r": 10 },
    { "x": -325, "y": 1750, "w": 75, "h": 25, "r": 10 },
    { "x": -325, "y": 1825, "w": 75, "h": 25, "r": 10 },
    { "x": -225, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -175, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -125, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -125, "y": 1700, "w": 75, "h": 25, "r": 10 },
    { "x": -125, "y": 1825, "w": 75, "h": 25, "r": 10 },
    { "x": -75, "y": 1700, "w": 25, "h": 150, "r": 10 }
  ],
```



Example 4 — JSON world + smooth camera (lerp).
camLerp(JSON): 0.03 Player: 47,1577 Cam: -599,1559

Screenshot of commit changes + description for POI field targeting commit:



changed POI field colour for visibility
HoiyaNaranja • 20 hours ago

changed world colour
HoiyaNaranja • 20 hours ago

Added mountain, world pan, and duo ...
HoiyaNaranja • 20 hours ago

Update Player.js
HoiyaNaranja • 22 hours ago

fixed spelling of camTargetting to cam...
HoiyaNaranja • 22 hours ago

changed speed to add excitement + ne...
HoiyaNaranja • 23 hours ago

added ability for POIs POI target to be ...
HoiyaNaranja • 23 hours ago

changed player.size to w,h and accomo...
HoiyaNaranja • 23 hours ago

fixed camera logic and readded player ...
HoiyaNaranja • 23 hours ago

added temp POI logic and temp POIs
HoiyaNaranja • 23 hours ago

added ability for POIs POI target to be different than x,y

```
I had to search up how "Nullish coalescing" operators and "Optional chaining" worked to get this to stop
breaking. Apparently even though the .poi would return null it would still attempt to look further and break
when it couldn't go further. So the optional chaining helps it stop before it can do that
```

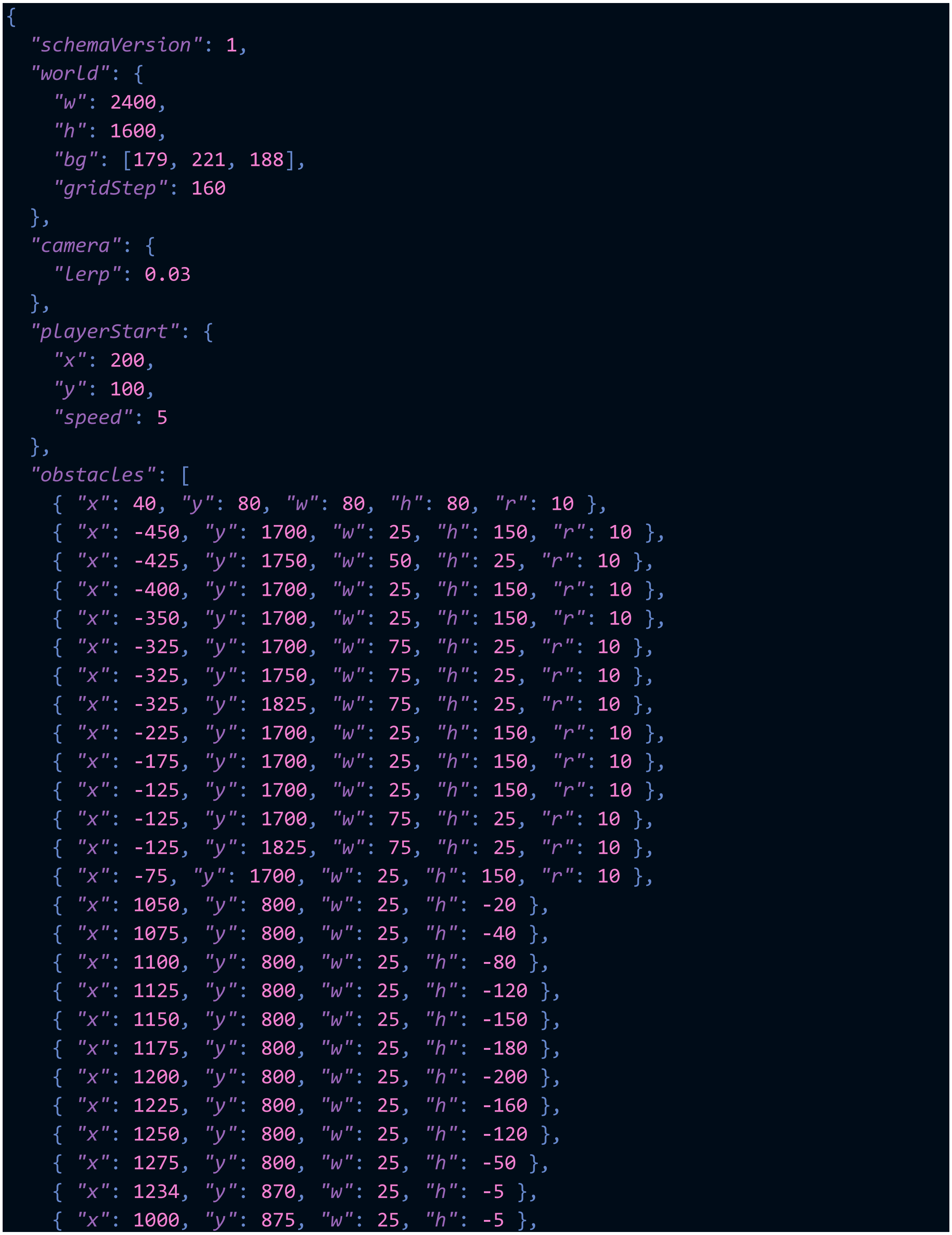
HoiyaNaranja cb1c456 +16 -6

2 changed files
sketch.js
world.json

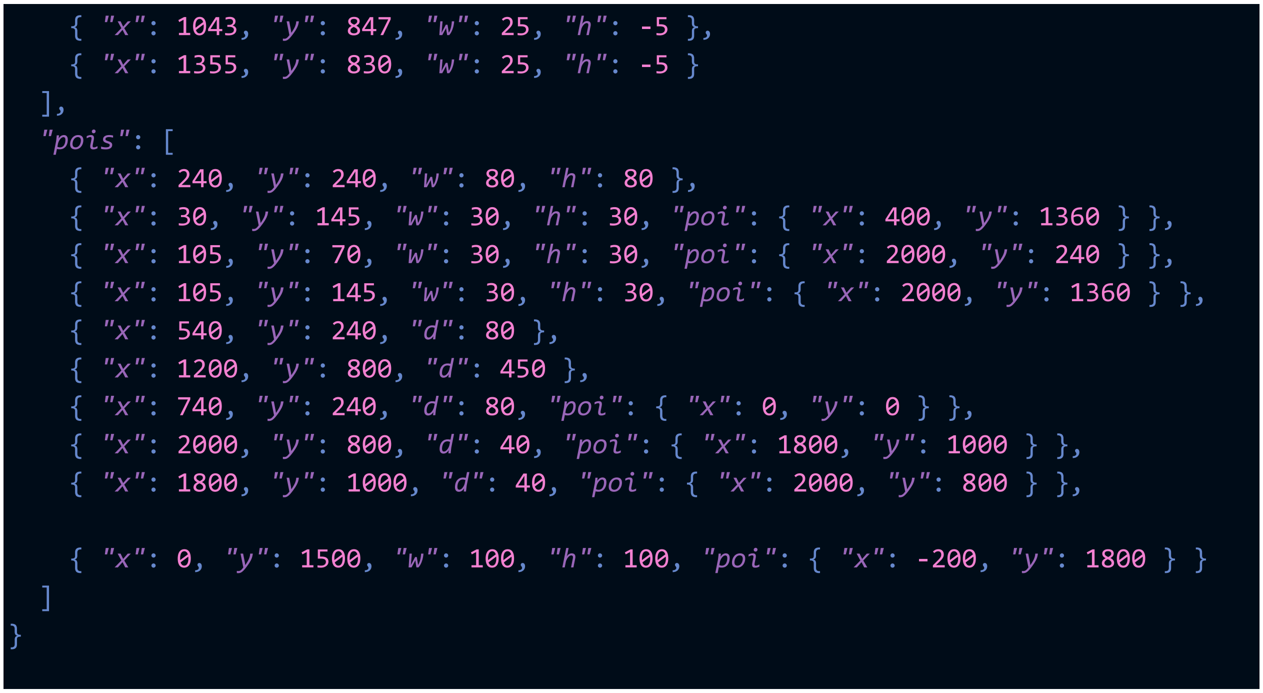
sketch.js

```
@@ -61,15 +61,24 @@ function draw() {

      for (p of level.pois) {
        if (p.d) {
-         if (dist(player.x, player.y, p.x, p.y) <= p.d / 2 +
max(player.w, player.h) / 2) {
-           targetX = p.x - width / 2;
-           targetY = p.y - height / 2;
+         if (
+           dist(player.x, player.y, p.x, p.y) <=
+           p.d / 2 + max(player.w, player.h) / 2
+         ) {
+           targetX = p.poi?.x ?? p.x;
+           targetY = p.poi?.y ?? p.y;
+
+           targetX -= width / 2;
+           targetY -= height / 2;
            camTargetting = true;
          }
        }
        if (overlapAABB(player, p)) {
```

Final “world.json”:



```json
{
  "schemaVersion": 1,
  "world": {
    "w": 2400,
    "h": 1600,
    "bg": [179, 221, 188],
    "gridStep": 160
  },
  "camera": {
    "lerp": 0.03
  },
  "playerStart": {
    "x": 200,
    "y": 100,
    "speed": 5
  },
  "obstacles": [
    { "x": 40, "y": 80, "w": 80, "h": 80, "r": 10 },
    { "x": -450, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -425, "y": 1750, "w": 50, "h": 25, "r": 10 },
    { "x": -400, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -350, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -325, "y": 1700, "w": 75, "h": 25, "r": 10 },
    { "x": -325, "y": 1750, "w": 75, "h": 25, "r": 10 },
    { "x": -325, "y": 1825, "w": 75, "h": 25, "r": 10 },
    { "x": -225, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -175, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -125, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": -125, "y": 1700, "w": 75, "h": 25, "r": 10 },
    { "x": -125, "y": 1825, "w": 75, "h": 25, "r": 10 },
    { "x": -75, "y": 1700, "w": 25, "h": 150, "r": 10 },
    { "x": 1050, "y": 800, "w": 25, "h": -20 },
    { "x": 1075, "y": 800, "w": 25, "h": -40 },
    { "x": 1100, "y": 800, "w": 25, "h": -80 },
    { "x": 1125, "y": 800, "w": 25, "h": -120 },
    { "x": 1150, "y": 800, "w": 25, "h": -150 },
    { "x": 1175, "y": 800, "w": 25, "h": -180 },
    { "x": 1200, "y": 800, "w": 25, "h": -200 },
    { "x": 1225, "y": 800, "w": 25, "h": -160 },
    { "x": 1250, "y": 800, "w": 25, "h": -120 },
    { "x": 1275, "y": 800, "w": 25, "h": -50 },
    { "x": 1234, "y": 870, "w": 25, "h": -5 },
    { "x": 1000, "y": 875, "w": 25, "h": -5 },
```



```
    { "x": 1043, "y": 847, "w": 25, "h": -5 },
    { "x": 1355, "y": 830, "w": 25, "h": -5 }
  ],
  "pois": [
    { "x": 240, "y": 240, "w": 80, "h": 80 },
    { "x": 30, "y": 145, "w": 30, "h": 30, "poi": { "x": 400, "y": 1360 } },
    { "x": 105, "y": 70, "w": 30, "h": 30, "poi": { "x": 2000, "y": 240 } },
    { "x": 105, "y": 145, "w": 30, "h": 30, "poi": { "x": 2000, "y": 1360 } },
    { "x": 540, "y": 240, "d": 80 },
    { "x": 1200, "y": 800, "d": 450 },
    { "x": 740, "y": 240, "d": 80, "poi": { "x": 0, "y": 0 } },
    { "x": 2000, "y": 800, "d": 40, "poi": { "x": 1800, "y": 1000 } },
    { "x": 1800, "y": 1000, "d": 40, "poi": { "x": 2000, "y": 800 } },

    { "x": 0, "y": 1500, "w": 100, "h": 100, "poi": { "x": -200, "y": 1800 } }
  ]
}
```

I added more POI fields to help show what it is capable of. Two fields bounce between each other, and three are placed on the original “obstacle” to act as a controller for the map to see where things are before fully exploring.

GenAI Documentation

No GenAI used for this task.