

Process & Decision Documentation

Entry Header

Name: Nat Nguyen

Primary responsibility for this work: adding a second level using JSON level data.

Goal of Work Session

For this Side Quest, I wanted to create another level with tiles/platforms using arrays.

Tools, Resources, or Inputs Used

- GenAI tools (if used)
- Example Code from Week 4 Lecture

GenAI Documentation

Date Used: February 10, 2026

Tool Disclosure: ChatGPT 5.2

Purpose of Use: Debugging + guidance on how to structure a second JSON level.

Summary of Interaction: ChatGPT showed me how to add another level object inside the "levels" array in levels.json

Human Decision Point(s): I changed the AI-generated level layout by manually adjusting platform positions (x/y values) to improve jump spacing and make the level more playable. I also changed the background colour to a lighter shade of blue so the black text would be readable. I also experimented with the blob movement → adjusting gravity to change the pacing of movement and jumping.

Integrity & Verification Note: I verified the output by testing the level in the browser, ensuring the jumps, platforms, and controls worked correctly. I also made sure I was able to jump between platforms and reach the top.

Scope of GenAI Use: GenAI did not write the full game code or all the personal, colorful design choices. The existing BlobPlayer physics, platform collision system, and level loading functions were already developed from the example code. GenAI was only used to help guide level structure and example platform placements.

Limitations or Misfires: Some of the suggested platform coordinates caused platforms appearing off-screen. The suggested background colour was also too dark at first, making the HUD text hard to read. I had to revise these values manually.

Appendix

Me:

How can I Generate a level using arrays or JSON data. I want to create a second layer with “Wonderland” vibes using colours. Here is my levels.json code:

```
{ "schemaVersion": 1, "levels": [ { "name": "Intro Steps", "gravity": 0.65, "jumpV": -11.0,
  "theme": { "bg": "#F0F0F0", "platform": "#C8C8C8", "blob": "#1478FF" }, "start": { "x": 80, "y":
  220, "r": 26 }, "platforms": [ { "x": 0, "y": 324, "w": 640, "h": 36 }, { "x": 120, "y": 254, "w": 120,
  "h": 12 }, { "x": 300, "y": 204, "w": 90, "h": 12 }, { "x": 440, "y": 144, "w": 130, "h": 12 }, { "x":
  520, "y": 254, "w": 90, "h": 12 } ] ] }
```

ChatGPT:

Sure — easiest way is to add a second level object in your levels.json file. Your sketch.js already supports switching levels when you press N because it uses data.levels.length. To make Level 2 different, you can change the theme colours and platform layout. For example, you can add more platforms in a zigzag shape or change the y-values so the level feels more vertical. You can update the "theme" colours and rewrite the "platforms" array with different x, y, w, h values.

Here's an example of what your Wonderland code could look like:

```
{
  "schemaVersion": 1,
  "levels": [
    {
      "name": "Intro Steps",
      "gravity": 0.65,
      "jumpV": -11.0,
      "theme": { "bg": "#F0F0F0", "platform": "#C8C8C8", "blob": "#1478FF" },
      "start": { "x": 80, "y": 220, "r": 26 },
```

```
"platforms": [  
  { "x": 0, "y": 324, "w": 640, "h": 36 },  
  { "x": 120, "y": 254, "w": 120, "h": 12 },  
  { "x": 300, "y": 204, "w": 90, "h": 12 },  
  { "x": 440, "y": 144, "w": 130, "h": 12 },  
  { "x": 520, "y": 254, "w": 90, "h": 12 },  
  { "x": 700, "y": 100, "w": 50, "h": 12 }  
]  
,  
  
{  
  "name": "Wonderland Steps",  
  "gravity": 0.65,  
  "jumpV": -11.0,  
  "theme": { "bg": "#0E1020", "platform": "#A7F3D0", "blob": "#F472B6" },  
  "start": { "x": 70, "y": 260, "r": 26 },  
  
  "platforms": [  
    { "x": 0, "y": 330, "w": 820, "h": 30 },  
  
    { "x": 60, "y": 280, "w": 100, "h": 12 },  
    { "x": 190, "y": 240, "w": 80, "h": 12 },  
    { "x": 320, "y": 210, "w": 110, "h": 12 },  
  
    { "x": 480, "y": 250, "w": 90, "h": 12 },  
    { "x": 620, "y": 220, "w": 100, "h": 12 },
```

```
{ "x": 740, "y": 180, "w": 70, "h": 12 },  
{ "x": 650, "y": 150, "w": 80, "h": 12 },  
  
{ "x": 520, "y": 120, "w": 80, "h": 12 },  
{ "x": 380, "y": 140, "w": 90, "h": 12 },  
  
{ "x": 250, "y": 110, "w": 70, "h": 12 },  
{ "x": 120, "y": 150, "w": 80, "h": 12 }  
]  
}  
]  
}
```

If you want, I can also make a third level using a grid array and loops. Do you want me to do that?