

If GenAI was used (keep each response as brief as possible):

Date Used: February 9, 2026

Tool Disclosure: Claude Haiku 4.5

Purpose of Use: I used Ai to help me create my game with different levels as it is a better coder than me.

Summary of Interaction:

- It made the levels
- it made the end screen
- it made the different effects in the levels
- It made the levels harder

Human Decision Point(s): I had to tell AI to change some of the levels because the blob would fall off the platforms and not respawn and I would get stuck on the level with no blob. Also, I made sure to change the colours off the levels to know there was a switch when you progressed. One change I made too was making the levels harder because they were all the same so now there more challenging the farther you go.

Integrity & Verification Note: I made sure the comments it made were true by checking in game and verifying.

Scope of GenAI Use: In the code I changed colours and changed some names. I changed some of the gravities to make the levels harder and also changed platform colors.

Limitations or Misfires: Sometimes when I asks the AI to make me levels they don't work or for example the portal to move onto the next level wouldn't take me to the next level.

Appendix

Please include a full transcript of your conversation with the GenAI. GenAI transcripts are used for transparency and verification only and are not assessed for writing quality or completeness.

Note: If GenAI is used in another language, you must include both the original language and a translated transcript in the appendix

Prompt:

Can you give me idea on a type of game to make.

Answer:

Here are some engaging game ideas you could build with your current p5.js + JSON/class skills:

Easy → Medium

1. **Puzzle Platformer** - Extend your current platformer with:
 - Moving platforms (use JSON to define paths/speeds)
 - Collectible items (coins, keys) that unlock level sections
 - Environmental hazards (spikes, lava) to avoid
2. **Breakout/Brick Breaker** - Classic but satisfying:
 - Ball physics bouncing off paddle/bricks
 - Power-ups (multi-ball, sticky paddle, slow-mo)
 - JSON for level layouts (brick patterns)
3. **Tower Defense** - Defend from waves:
 - Place towers on a grid
 - Enemies follow paths (defined in JSON)
 - Upgrade systems and resource management
4. **Maze/Dungeon Crawler** - Top-down adventure:
 - Generate mazes procedurally (like your level generation!)
 - Collect items, defeat enemies
 - Multiple rooms/levels with different themes

Medium → Hard

5. **Roguelike** - Permadeath with progression:
 - Procedurally generated rooms
 - Combat/dodging mechanics
 - Meta-progression (unlock abilities between runs)
6. **Match-3 Puzzle** (Candy Crush style):
 - Grid-based swapping
 - Cascade effects and combos
 - Progressive difficulty with special blocks
7. **Flappy Bird Clone** (deceptively fun):
 - Simple but challenging mechanics
 - Procedurally generated obstacles (like your platforms!)
 - Leaderboard system
8. **Resource Management Simulation**:
 - Farm/city builder with JSON-based grids
 - Build structures, manage resources
 - Time-based mechanics (day/night, seasons)

My Recommendations:

- **Easiest win:** Extend your platformer with collectibles + hazards
- **Most fun:** Roguelike or Maze Crawler (builds on procedural generation skills)
- **Best portfolio piece:** Tower Defense (shows UI, game loops, data structures)





What genre appeals to you most? I can help you prototype whichever sounds exciting!

Prompt:

Awesome can you help me build a game that involves jumping onto platforms and needing to get to a certain spot in order to move on to the next level where it is a new environment

Answer:

The game we just built has:

-  **Platform jumping** (your blob player)
-  **Goal spots** to reach (orange pulsing circles)
-  **Auto-progression** to next level when you reach the goal
-  **New environments** (Level 1 has light gray theme, Level 2 has dark/neon theme)

What we can enhance:

Would you like me to add any of these to make it more interesting?

1. **More Levels** - I can create 3-5 additional levels with totally different themes/difficulty
2. **Collectibles** - Coins/gems to collect before reaching the goal
3. **Hazards** - Spikes, moving obstacles, or falling blocks to avoid
4. **Moving Platforms** - Platforms that move side-to-side or up-down
5. **Power-ups** - Jump boost, slow-motion, shield, etc.
6. **Enemy Patrol** - Simple enemies to avoid/jump over
7. **Level Timer** - Time limits to add challenge
8. **Visual Polish** - Better animations, particles, sound effects

Which sounds fun to you? Pick 2-3 and I'll add them! Or if you want to test the current game first, you can open [index.html](#) in a browser to see it in action.

Prompt:

Can you make more levels and I will adjust the way they look and feel.

Answer:

Perfect! Now let me add a level counter to the HUD so players know their progress:

Edited

sketch.js+3-1

Done! 🎮 I've added **4 new levels** to your game:

New Levels:

3. **Forest Path** 🌲 - Green forest theme

- Medium difficulty with wider platforms
- Two rows of floating platforms

4. **Volcanic Escape** 🌋 - Red/orange volcanic theme

- Slightly heavier gravity (harder jumping)
- Mix of diagonal and floating platforms

5. **Ice Crown** ❄️ - Blue icy theme

- Lower gravity (easier, higher jumps)
- 9 stairs + lots of platforms
- Most platforms for precision jumping

6. **Final Challenge** 💜 - Purple endgame theme

- Hardest difficulty with multiple obstacle types
- 10 stairs + 3 different platform patterns
- True test of skill!

Features:

- Each level has **unique colors & themes**
- **Different gravity/jump values** change the feel
- **Level counter** in HUD shows progress (e.g., "Level 1 / 6")
- Automatically loops back to Level 1 after beating all 6
- Progressively harder with more platforms and larger gaps

Try playing through them! The difficulty ramps up as you progress. 🚀

Prompt:

make it so there is an end to the game if you pass all the levels.

Answer:

Let me check the current state of the file:

Read [sketch.js](#)

Perfect! Now let me add the loadLevel function:

Edited

sketch.js+21-0

Perfect! ✅ Done! I've added an **end game screen** that triggers when you complete all 6 levels.

What happens now:

1. **Beat all levels** → Game shows victory screen with:
 - 🎉 Celebration message
 - Level completion count
 - Instructions to restart
2. **Press R** → Restarts from Level 1
3. **Press N** → Jump to specific levels anytime (even on win screen)

The game no longer loops automatically—it stops at the victory screen for a proper ending! 🏆



Prompt:

make it so the games get harder as each level progresses. make some have less gravity or make the space farther apart so you have to time it correctly or make the blob be able to fall off and have to respawn

Answer:

Now let me add fall detection to the sketch:

Edited

sketch.js+6-2

Edited

sketch.js+11-0

Perfect! ✅ I've made the game progressively harder with these changes:

Progressive Difficulty:

Level 1 (Intro) - Easiest

- Standard gravity (0.65)
- Small gaps (45px)
- 8 stairs

Level 2 (Forest) - Slightly Harder

- Lighter gravity (0.62) - easier to control
- Larger gaps (60px)
- 7 stairs + 9 platforms
- Wide spacing requires precision timing

Level 3 (Volcanic) - Medium Difficulty

- Heavy gravity (0.72) - falls fast
- Weaker jump ($\downarrow 10.2$) - harder to reach platforms
- Huge gaps (70px)
- 8 stairs with challenging diagonal patterns

Level 4 (Ice Crown) - Hard

- Low gravity (0.55) - floaty, unpredictable
- Super strong jump ($\uparrow 12.5$) - overshooting is risky
- Biggest gaps (80px)
- 10 stairs - must time perfectly

Level 5 (Final Challenge) - Hardest

- Combination of moderate gravity (0.68) & jump
- Massive gaps (90px)
- 11 stairs + 3 types of obstacles
- Most platforms to navigate

Respawn System:

- **Fall below the screen** → Instantly respawn at level start
- No level reset, just respawn position
- Lets players retry without losing progress

Try it now! Each level has distinct physics, gaps, and platform spacing! 🎮