

10 Static Web Game Specifications

This document outlines ten game concepts designed for high engagement and simple implementation using static web technologies (HTML5 Canvas, CSS3, and Vanilla JavaScript).

1. The Markov Mind Reader (AI Prediction)

Description: A psychological "Rock-Paper-Scissors" clone where the computer uses a Markov Chain to predict the user's next move based on their historical patterns. **Playing Flow:**

1. User clicks one of three buttons: Rock, Paper, or Scissors.
2. The computer reveals its choice (calculated *before* the user's reveal).
3. The game displays the win/loss streak.
4. An "Insight" panel shows the computer's confidence percentage for its prediction.

2. Gradient Descent: The Hill Climber

Description: A physics-based puzzle where the user must guide a ball to the global minimum of a complex, randomly generated mathematical function (curve). **Playing Flow:**

1. A 2D wave-like terrain is rendered on a Canvas.
2. The user clicks to place a "ball."
3. The user adjusts "Momentum" and "Learning Rate" sliders.
4. The ball rolls based on the slope; the goal is to reach the lowest point within 10 seconds.

3. Semantic Distance (Word Association)

Description: Inspired by *Contexto*, users guess a secret word, and the game provides a "similarity score" based on a local JSON distance matrix. **Playing Flow:**

1. User types a word into a text box.
2. The game calculates the "distance" to the secret word.
3. A progress bar fills up as the user gets "warmer" (semantically closer).
4. The user wins when they hit the exact word.

4. The Entanglement Puzzle

Description: A logic game based on Quantum Entanglement. Changing the state of one "particle" (node) affects its entangled pair elsewhere on a grid. **Playing Flow:**

1. A grid of 12 nodes appears in random states (Spin Up/Down).

2. Clicking Node A flips Node A *and* its hidden entangled partner, Node B.
3. The user must figure out the pairs by observation.
4. The goal is to set all nodes to "Spin Up" simultaneously.

5. CSS 3D Maze Runner

Description: A first-person perspective maze using CSS `perspective` and `transform-3d`. No WebGL required—just DOM manipulation. **Playing Flow:**

1. User sees a "3D" hallway rendered via CSS.
2. They use Arrow keys or WASD to move.
3. Collision detection checks against a 2D array representing the map.
4. The objective is to find the "Exit" door before the timer hits zero.

6. ASCII Rogue-lite

Description: A dungeon crawler where the entire world is rendered in text (ASCII). **Playing Flow:**

1. The screen is a `<pre>` tag. `@` is the player, `#` are walls, `E` are enemies.
2. Player moves turn-by-turn using the keyboard.
3. Each room is procedurally generated using a simple BSP (Binary Space Partitioning) algorithm.
4. Collecting `$` increases score; reaching the `>` staircase goes to the next floor.

7. The Kernel Painter (CNN Simulation)

Description: A creative puzzle where users must "filter" a blurry image into a sharp one by manually setting values in a 3×3 convolution matrix. **Playing Flow:**

1. Two canvases: "Target Image" (Sharp) and "Your Image" (Blurry/Original).
2. User enters numbers into a 3×3 grid (The Kernel).
3. The "Apply" button runs the convolution over the original image.
4. A "Similarity %" shows how close the filtered result is to the target.

8. Genetic "Critter" Evolution

Description: A sandbox where simple shapes (critters) with randomized joint angles try to "walk" across the screen. **Playing Flow:**

1. 10 critters are spawned at the start line.
2. They move using randomized sine-wave cycles for their limbs.

3. After 15 seconds, the one that traveled furthest is "cloned" with slight mutations.
4. The user watches the population evolve to become better walkers over several generations.

9. Hexadecimal Color Hunter

Description: A fast-paced arcade game where the user is given a Hex code (e.g., #FF5733) and must pick the correct color from a sea of floating bubbles. **Playing Flow:**

1. A Hex code appears at the top.
2. Dozens of colored circles bounce around a Canvas.
3. User must click the correct one before it disappears.
4. As levels increase, the colors become more similar (e.g., #FF5733 vs #FF5834).

10. Bit-Shift Defuser

Description: A cyberpunk-themed math game where you "defuse a bomb" by performing binary operations. **Playing Flow:**

1. A target decimal number is shown (e.g., 42).
2. The user starts with a 8-bit byte 00000000 .
3. Buttons for << 1 (Left Shift), >> 1 (Right Shift), OR 1 , and NOT are available.
4. The user must reach the target decimal using the fewest binary operations possible.