Perfect — this is going to be fun and organized 🔧🗺️ Let's begin **Step 1: Build a Static Gameboard Layout** using only **HTML + CSS**, and keep it flexible so you can loop through project posts later.

✅STEP 1 GOAL

Create:

- A grid layout for your game board tiles
- A fixed player piece starting at tile 1
- A sidebar placeholder for project info

🔼FILE STRUCTURE FOR THIS STEP

Inside your theme (likely `/funbrain`), add or edit:

```
📄 board.php          ← new template file (or modify index.php)
📁 /css/style.css      ← add board styles
📁 /assets/player.png  ← player icon with propeller hat
```

We'll use `board.php` so we can keep the homepage clean.

HTML: Static Board Layout (Progressively Built)

1. Outer Wrapper (`div.gameboard-wrapper`)

This container holds both the board and the sidebar, allowing us to lay them out side-by-side.

```
<div class="gameboard-wrapper">
```

- Purpose: Enables flexible layout with `display: flex`
- Will hold both the sidebar and the game board

---

2. Sidebar (`aside.sidebar`)

This is where we'll dynamically show info about each tile as you hover.

```
    <aside class="sidebar">
      <h2>💼 Project Info</h2>
      <p>Hover a tile to learn more!</p>
    </aside>
```

   • Purpose: Holds dynamic text (updated later via JS)
   • Styled to match a game card or info box

---

3. Gameboard Grid (`section.board`)

This is the actual grid of tiles.

```
    <section class="board">
      <div class="tile" data-tile="1">1</div>
      <div class="tile" data-tile="2">2</div>
      <div class="tile" data-tile="3">3</div>
      <div class="tile" data-tile="4">4</div>
      <div class="tile" data-tile="5">5</div>
      <div class="tile" data-tile="6">6</div>
```

   • Purpose: Visually represents each project space on the board
   • `data-tile` attributes will be useful for JS animations

---

4. Player Piece Image (`img.player-piece`)

This represents the user's avatar, which will move across the board.

```
    <img src="<?php echo get_template_directory_uri(); ?>/assets/player.png"
        alt="Player Piece"
        class="player-piece"
        id="player">
  </section>
</div>
```

   • Purpose: Visually indicates current position; moves with JS
   • Positioned absolutely over the board

---

5. Full Template (`board.php`)

Wrap it all together inside a WordPress template:
```

```php
<?php
/* Template Name: Gameboard */
get_header();
?>

<div class="gameboard-wrapper">
  <aside class="sidebar">
    <h2>💼 Project Info</h2>
    <p>Hover a tile to learn more!</p>
  </aside>

  <section class="board">
    <div class="tile" data-tile="1">1</div>
    <div class="tile" data-tile="2">2</div>
    <div class="tile" data-tile="3">3</div>
    <div class="tile" data-tile="4">4</div>
    <div class="tile" data-tile="5">5</div>
    <div class="tile" data-tile="6">6</div>

    <img src="<?php echo get_template_directory_uri(); ?>/assets/player.png"
         alt="Player Piece"
         class="player-piece"
         id="player">
  </section>
</div>

<?php get_footer(); ?>
```

🔙CSS: Grid Layout + Player Piece (Progressively Explained)

1. Wrapper Layout

```css
.gameboard-wrapper {
  display: flex;              /* Places sidebar and board side-by-side */
  flex-wrap: wrap;            /* Allows wrapping on smaller screens */
  gap: 2rem;                  /* Space between sidebar and board */
  justify-content: center;    /* Center contents horizontally */
  align-items: flex-start;    /* Align top edges */
  padding: 2rem;              /* Space around the whole layout */
}
```

2. Sidebar Styling

```css
.sidebar {
  flex: 1 1 250px;          /* Flexible column base 250px */
  max-width: 300px;         /* Limit to prevent overgrowth */
  background: #fff;         /* White background */
  border-radius: 16px;      /* Rounded corners */
  padding: 1.5rem;          /* Internal spacing */
  box-shadow: 4px 4px 0 #000;/* Comic-style drop shadow */
  font-family: "Comic Sans MS", cursive;  /* Arcade-style font */
}
```

3. Grid of Tiles

```css
.board {
  position: relative;       /* Needed for absolutely positioning player */
  display: grid;            /* Use CSS grid for layout */
  grid-template-columns: repeat(3, 100px); /* 3 columns */
  gap: 1rem;                /* Space between tiles */
  background-color: #222;   /* Dark board background */
  padding: 1rem;            /* Padding inside board area */
  border-radius: 12px;      /* Rounded board edges */
}
```

4. Individual Tile Styling

```css
.tile {
  width: 100px;
  height: 100px;
  background: #ffd700;        /* Arcade yellow */
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 1.4rem;
  font-weight: bold;
  border-radius: 16px;
  box-shadow: 2px 2px 0 #444; /* Depth effect */
  position: relative;
  cursor: pointer;            /* Clickable look */
}
```

5. Player Piece

```css
.player-piece {
  width: 60px;                        /* Avatar size */
  position: absolute;             /* Positioned over tiles */
  top: 0;
  left: 0;
  transform: translate(0, 0);      /* Will move later with JS */
  transition: transform 0.4s ease;   /* Smooth movement */
  z-index: 10;                      /* On top of everything */
}
```

✅Checkpoint

You should now see:

- A grid of yellow tiles (labeled 1–6)
- A propeller hat image sitting at tile 1 (upper-left)
- A sidebar ready for info display

🈚Optional: You can drop a background image or rounded edges on tiles to mimic the Funbrain "path" shape later.

🚫Next Step?

We can now:

- Start GSAP animation on hover
- Make the `.player-piece` move to tile positions

Let me know when you're ready to jump into **Step 2: GSAP Movement + Sidebar Change**.