How to Play the Game: For the best experience, I recommend playing this on a computer rather than a phone or tablet. On a computer, use the arrow keys to move left and right, while on a phone or tablet, simply tap in the direction you want to go. The goal is to keep jumping upwards by landing on different types of platforms while avoiding obstacles. There are green stone platforms, which are permanent and safe to land on, while pink platforms give you a huge jump boost when landed on. Be careful with orange platforms, as they disappear after just one step, making timing crucial. The biggest challenge is avoiding the red sliding blocks, which act as obstacles—touch one, and it's game over.

View my Code: https://github.com/Patel112506/DoodleClimber

Context: I decided to recreate Doodle Jump as a way to challenge myself and level up my coding skills. While the game itself is pretty simple, building it from scratch turned out to be way harder than I expected. But throughout the process, I sharpened my JavaScript skills and got much more comfortable using TypeScript.

The Coding Languages I Used: I leaned heavily on TypeScript (**TypeScript is basically an extension or advanced level of Javascript**) for the game logic—setting up how things like power-ups, obstacles, and other game functions work, while keeping everything organized. For the User Interface (like start button, game over screen, etc.) I used React and JSX a little bit. And to handle all the visuals and give it that slick, colorful, and polished look, I used HTML5 Canvas and CSS.

Challenges I Faced:

- **Game Dynamics:** Getting the jump to feel just right, especially with the collision detection for red obstacles, was way trickier than I imagined.

- **Canvas & Background color Management:** Keeping everything running smoothly without stutters and lag was time consuming.

- **TypeScript Complexity:** Wrangling types, interfaces, and game state without totally breaking things or lagging was super frustrating at times.

- **Input Handling:** Balancing keyboard and touch controls to make sure they worked seamlessly took a lot of trial and error.

What I Learned: This project definitely wasn’t all sunshine and rainbows. It was a lot tougher than I thought it’d be. Whenever I hit a roadblock, I’d reach out to my computer programming professor at the University of Cincinnati or ask for advice on Reddit threads from MIT and Georgia Tech students. But in the end, this whole project was a blast! It wasn’t just about building a game—it really helped me refine my coding skills and taught me a lot along the way.