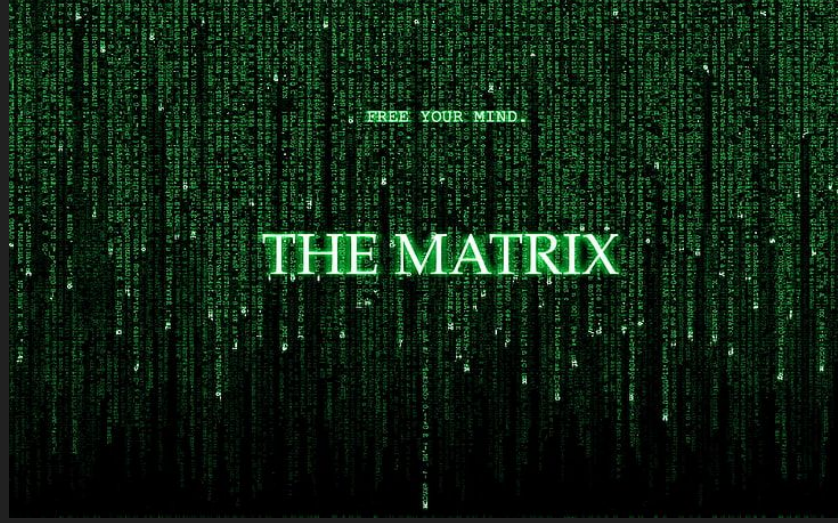
DM-GY 6063 2024F B
Mid-Term Project

Priyadarshan Sabarikannan
Net id:ps4907
2024/11/04

THE MATRIX



Brief Plot Summary:

In "The Matrix," the story is set in a future where humans unknowingly live in a simulated reality created by intelligent machines. The main character, Neo, is a hacker who feels that something is fundamentally wrong with the world around him. His life changes when he is contacted by a group of rebels led by Morpheus, who reveal that the reality he knows is an illusion. To discover the truth, Neo takes a red pill that wakes him up to the real world, where he learns about the ongoing war between humans and machines. As he trains with Morpheus and the rebels, Neo discovers that he is believed to be "The One," a figure destined to free humanity from the clutches of the machines. The film explores themes of reality, choice, and the struggle for freedom.

THE MATRIX

Aspect of Film Explored:

The project cover/poster focuses on the theme of reality versus illusion, a central concept in "The Matrix." It visually contrasts the digital green code of the Matrix with the desaturated tones of the real world, highlighting the conflict between simulated environments and harsh realities.

Description of Logic Implemented:

Falling Characters: The setup() function creates 100 characters that fall from the top of the canvas, influenced by gravity, which varies based on mouse position via the mouseMoved() function.

Color Palette: Characters are assigned colors from a predefined palette of green shades, ensuring a cohesive visual style.

Mouse Interaction: The mousePressed() function toggles characters between free-falling and a circular pattern. If a pattern exists, characters return to falling behavior with new colors.

Circular Pattern Formation: The formCirclePattern() function selects 20 random characters, calculates their positions based on angles to form a circle, and changes their color to white while in the pattern.

Character Updates: The updateFallingChar(char) function updates character positions based on whether they are falling or in a pattern. If a character falls off-screen, it reappears at a random position at the top.

CHALLENGE

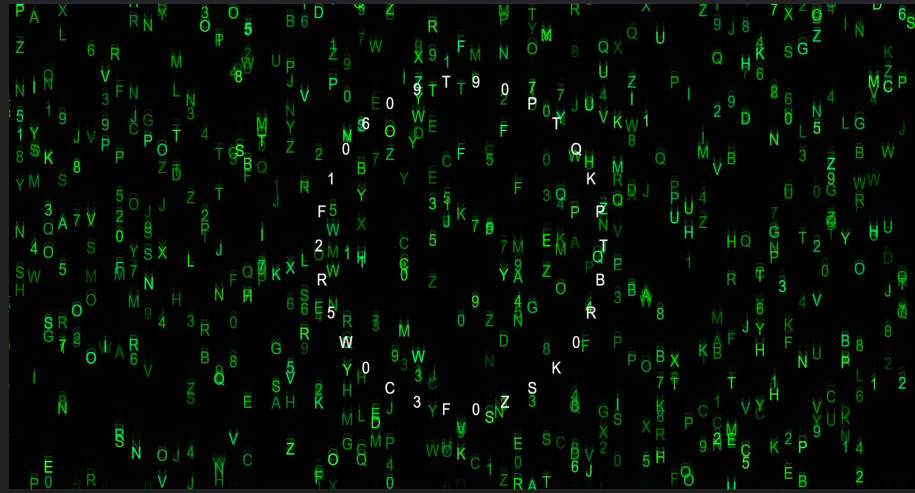
What I Thought Would Be Challenging:

I initially thought the most challenging aspect would be replicating the visual aesthetic of "The Matrix," particularly the iconic falling green code. I anticipated difficulties in achieving a smooth animation and creating a seamless integration of interactive elements that would reflect the film's themes.

What Was Actually Challenging:

In reality, the most challenging part was managing user interactions effectively. Ensuring that the falling characters not only reacted to mouse movements but also formed distinct patterns without overlapping or disrupting the flow of the animation required careful coding and debugging. Balancing the aesthetics with interactive functionality proved to be more complex than expected, particularly in maintaining a user experience while adhering to the thematic elements of the film.

HIGHLIGHT



```
// Function to form a circular pattern
function formCirclePattern() {
  let centerX = width / 2;
  let centerY = height / 2;

  currentPatternChars = randomSubset(fallingChars, 30); // Reduced number for performance

  for (let i = 0; i < currentPatternChars.length; i++) {
    let char = currentPatternChars[i];
    let angle = map(i, 0, currentPatternChars.length, 0, TWO_PI);
    char.targetX = centerX + cos(angle) * radius;
    char.targetY = centerY + sin(angle) * radius;
    char.inPattern = true;
    char.color = color(255); // Set color to white for pattern
  }
}
```

DEMO

LINK: https://priyadarshan-dm-gy-6063b.github.io/Midterm/

THANK YOU