Name :Mfanafuthi

Surname :Pomba

Student no:222358416

Report: Pac-Man WebGL Game

Introduction:

The Pac-Man WebGL game is a simple browser-based game implemented using WebGL (Web Graphics Library) for rendering graphics. It follows the classic Pac-Man arcade game concept where the player controls Pac-Man to eat dots while avoiding ghosts. This report provides an overview of the game structure, its components, functionality, and potential improvements.

Game Structure:

1. HTML Structures

- The game interface is defined using HTML5 markup.

- It consists of a canvas element where WebGL rendering occurs.

- Minimal styling is applied to the canvas and body to ensure proper display.

2. Shader Programs:

- The game utilizes two shader programs: vertex shader (vsSource) and fragment shader (fsSource).

- These shaders handle the transformation and rendering of graphical elements on the canvas.

3. JavaScript Implementation:

- The core game logic is implemented using JavaScript.

- WebGL context is initialized, shaders are compiled, and programs are created.

- Pac-Man, Ghosts, and Dots are represented as circles, each with specific attributes.

- Game state is updated continuously based on user input and predefined behavior.

- Rendering is performed within the game loop, updating the canvas with each frame.

Components:

1. Pac-Man:

- Controlled by the player using arrow keys.

- Moves around the canvas, eating dots and avoiding ghosts.

- Has a mouth animation that opens and closes periodically.

2. Ghosts:

- Move autonomously, chasing Pac-Man.

- Collision with Pac-Man results in game over.

3. Dots:

- Scattered across the canvas.

- Eaten by Pac-Man upon collision, reducing the total count.

- Game ends when all dots are eaten.

Functionality:

1. Movement:

- Pac-Man moves in the direction indicated by arrow keys.

- Ghosts chase Pac-Man using simple AI based on distance calculation.

2. Collision Detection:

- Collision detection ensures interaction between Pac-Man, Ghosts, and Dots.

- Game over occurs when Pac-Man collides with a ghost.

- Dots are removed from the canvas upon collision with Pac-Man.

3. Game Loop:

- The game loop continuously updates and renders the game state.

- It ensures smooth animation and real-time interaction.

Potential Improvements:

1. Graphics Enhancement:

- Implement more visually appealing graphics for Pac-Man, Ghosts, and Dots.

- Add animations and effects to enhance the gaming experience.

2. Level Design:

- Introduce multiple levels with increasing difficulty.

- Design unique mazes to navigate through.

3. Audio Integration:

- Include sound effects for actions such as eating dots, collision, and game over.

4. Scoring System:

- Implement a scoring system to track and display the player's progress.

5. Optimization:

- Optimize code for better performance, especially for handling a larger number of game elements.

Conclusion:

The Pac-Man WebGL game provides a fun and interactive experience for players, reminiscent of the classic arcade game. Its simple yet engaging gameplay mechanics make it suitable for casual gaming sessions. With further enhancements and refinements, it has the potential to become even more captivating for players of all ages.