A black background with blue text

Description automatically generated

# SNAKE GAME USING JAVASCRIPT AND OPENGL(WEBGL).

Name: Siphokazi Veronica

Surname: Mdze

Student Number: 219080682

Module: Computer Graphics 2

Due Date: 15 May 2024

**Introduction**

This report documents the development and implementation of a simple Snake game using WebGL for rendering. The game is designed to provide an interactive and enjoyable experience for players.

**Overview**

The game involves controlling a snake to navigate a grid-based environment, consuming food to grow longer while avoiding collisions with the walls and its own body. The goal is to achieve the highest possible score by eating as much food as possible before encountering a collision.

**Features**

1. **Snake Movement**: Players control the direction of the snake using arrow keys.
2. **Food Consumption**: When the snake's head collides with food, it grows longer, and a new piece of food spawns at a random location.
3. **Collision Detection**: The game detects collisions between the snake and the walls or its own body, ending the game when such collisions occur.
4. **Game Over**: Upon collision, the game ends, displaying a game over message and stopping further gameplay.

**Rules**

1. **Snake Movement**: The snake moves continuously in the direction specified by the player until a new direction is inputted.
2. **Food Consumption**: When the snake's head collides with food, the snake grows longer, and the player's score increases.
3. **Collision Detection**:
   * If the snake's head collides with the walls of the game area, the game ends.
   * If the snake's head collides with its own body, the game ends.
4. **Game Over**: When the game ends, a game over message is displayed, and the player is prompted to restart the game.

**Implementation**

**Setup**

* Canvas and WebGL context are initialized for graphics rendering.
* Game parameters such as block size, grid dimensions, and initial snake position are set up.

**Graphics and Shaders**

* WebGL shaders are used to define how the snake and food are rendered on the canvas.

**Event Handling**

* Keyboard events are listened for to control the snake's movement.

**Movement and Collision Detection**

* Functions handle snake movement, food consumption, and collision detection.

**Rendering**

* The **render** function updates and draws the snake and food on the canvas using WebGL.

**Game Loop**

* A game loop using **requestAnimationFrame** continuously updates and renders the game while it is in progress.

**End Game**

* When the game ends, a game over message is displayed, and the game loop is stopped.

**Conclusion**

In conclusion, the WebGL Snake game offers an engaging and nostalgic gaming experience. With intuitive controls, simple rules, and challenging gameplay, it provides entertainment for players of all ages. Further enhancements and optimizations could be made to improve the game's features and performance.