SURVIVAL MAZE GAME

DUȚU ALIN CĂLIN

CONTENTS

1 INTRODUCTION

Welcome to the documentation of the Survival Maze Game project, a puzzle action game developed using OpenGL. This document is a comprehensive guide for understanding the architecture, design principles, and technical details behind the game's development.

1.1 About the game

The Survival Maze Game is an immersive experience set in a meticulously crafted world, leveraging the power of OpenGL to deliver interesting and smooth gameplay. Whether you're exploring intricate puzzles, or engaging in combat, this game is designed to discover the multitude of capabilities OpenGL has to offer.

- Genre: First-Person Shooter, Puzzle
- Platform: Windows
- Engine: Custom-built using OpenGL
- Target Audience: Casual gamers, Puzzle enthusiasts

1.2 Objectives

This documentation is intended for developers, designers, and technical enthusiasts interested in understanding the inner workings of the Survival Maze Game. It covers everything from the game's architecture, shader programs, and rendering techniques to input handling, physics integration, and optimization strategies.

In addition, the project aims to provide an overview of developing a game in OpenGL at an intermediate level. The following objectives are taken into consideration:

- Create a protagonist that can shoot projectiles
- Create an enemy model that can deal damage and get damaged by the player
- Set multiple maze environments with different configurations
- Implement collisions between entities

2 OVERVIEW

The Survival Maze is set in one of the many short, but challenging mazes. The story follows the narrative of a young explorer who is in search of a priceless treasury, the Glitched Rabbit.

Figure 1: The Glitched Rabbit

The player is going to meet with the guardians of the treasure. The guardians are blocking the path to the treasure by spinning in their set position. To neutralize the guardians, the explorer has one ace up his sleeve.

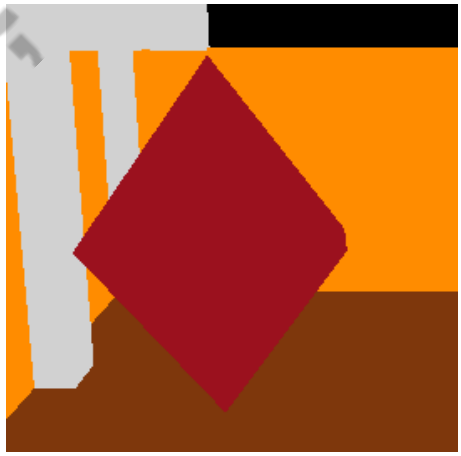

Figure 2: Guardian

In the game, the player must explore the mazes, defeat enemies, and find the treasure until time runs out. The game offers a combination of real-time action, strategy, and puzzle-solving, making for diverse and engaging gameplay. The player will also have access to a weapon that can be used to deal with enemies throughout the game.

The Survival Maze Game is developed using the OpenGL framework, which provides the foundation for rendering 3D graphics. The game incorporates specific technologies, such as shaders and skeletal animations, and is optimized for both high-end and mid-range hardware.

3 IMPLEMENTATION DETAILS

3.1 Explorer

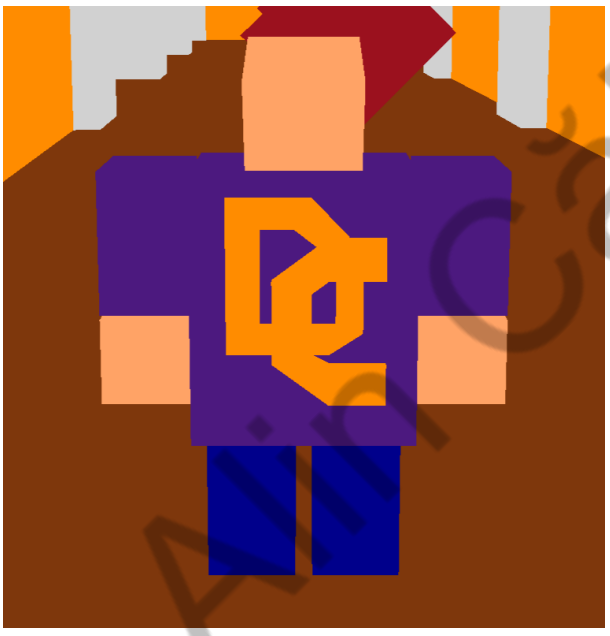

Figure 3: The explorer

The goal of the game is to go through the mazes, fight with the enemies, and find the treasure until the time runs out. When starting the game, the explorer is spawned in one of the many labyrinths and has to find the way to the treasure.

The explorer is the main character of the game. Its objective is to move through the maze and find the treasure as soon as possible. The player is able to move him using the following commands:

- W - move forward
- S - move backward
- A - move to the left
- D - move to the right

When the game starts, the player can be seen in the third person and the camera can be oriented using mouse movement. Horizontally, the player can look at a 360-degree angle,

however, vertically, the camera can be moved at a maximum elevation of 60 degrees and a maximum depression of -60 degrees.

3.1.1 Explorer's design

The Explorer's design has been created using a multitude of modified cubes called boxes that had been translated, rotated to the specific position of the body part, and scaled to a more natural size to make the final design look realistic.

Additionally, 2 custom shapes had been created resembling 3D shapes of C and D which have been added on the back of the Explorer's t-shirt in order to add a plus on the aesthetics and originality part of the character's design.

3.1.2 Weapon

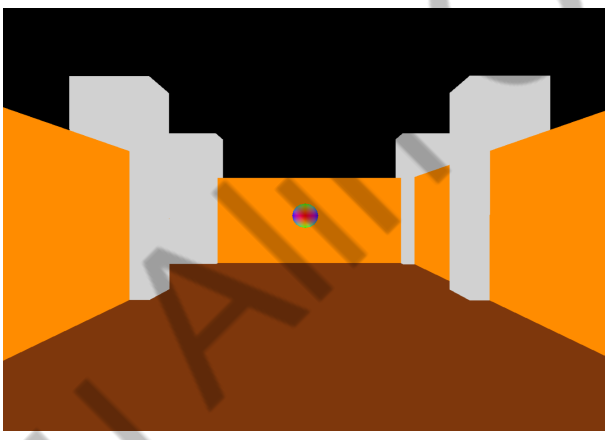

Figure 4: Siege mode

For the corner cases that might appear in the maze, the Explorer is equipped with a weapon that launches projectiles. In order to access it, the player has to use the CTRL command to switch to the Siege mode.

In the Siege mode, the perspective switches to first-person, and the aim of the weapon is shown on the screen as a multicolored sphere. The camera on the first-person view has the same limits as the third-person view.

In addition, when the player clicks, a projectile will be launched from the weapon. The projectile has enough power to eliminate a single enemy with ease, but it will go no further.

3.2 Guardians

The Guardians of the Treasure [Fig.2] are represented as rhomboids and distinguish themselves with their rotating movement on their position. They will deal damage to the Explorer if he collides with one of them, so it won't let the player go further without any loss of HP.

To effectively deal with guards, the Explorer should use his weapon and fire at the guards. When a guard is shot, it will stop spinning, expand to a bigger size, and then implode into oblivion. The death animation is an up-scaling animation implemented in the Vertex Shader.

3.3 HUD and ending scenarios

Figure 5: HUD

The HUD can be seen on the top left of the screen and it contains the HP bar colored with red and the time bar colored with white.

The HP bar represents the Explorer's health and it drops if he collides with the guardians. The Explorer can withstand only 5 collisions with the guardians until he collapses, in which case the game ends in a loss with the "WORST ENDING" scenario.

The time bar represents the time left until the maze closes forever and it retracts for every second it passes. The time set for the Explorer is set to 180 seconds. If the Explorer doesn't manage to find the treasure until the time runs out, the game ends in a loss with the "BAD ENDING" scenario.

However, if the player manages to find the treasure of the Glitched Rabbit on time and with HP left to spare, the game ends in a win with the "GOOD ENDING" scenario.

3.4 Levels

The maze levels are predefined in maze configuration files having a 21x21 square dimension and are chosen randomly at the game's startup. The notations used in the maze configuration files are :

- $+$ - for a pillar
- $-$ - horizontal wall
- $|$ - vertical wall
- E - Enemy position
- F - Treasure position (Finish point position)

Additionally, the file has a parameter, after the maze definition, that represents the rotation of the treasure pillar to make it face the Explorer when he reaches it.

An example of a maze configuration is:

```
+-+-+-+-+-+-+-+-+-+-+-+
|   | |E  | |   | |F|
+ +-+ +-+ + + + + + + +
| | |E  |     |E| |E|
+ + +-+ + + + + + + + +
|  E    | | | |   |E|
+ +-+-+-+-+-+ + + + +
|E|    E  |   | |E|E|
+ +-+-+-+ +-+ +-+ + +
|   |  E  |E| |    E|
+-+ + +-+ + + +-+-+ +
|  E|   | | |  E|  E|
+-+ +-+-+ + + +-+-+ +
|     |E| |   |E  |E|
+ +-+ + + + +-+-+ + +
|  E|    E      | |E|
+ + + + +-+ +-+-+ + +
|E| | | |E  |E| |E|E|
+-+-+ +-+-+-+ + + + +
|E     E          E|
+-+-+-+-+-+-+-+-+-+-+
0
```

When the game starts, the program randomly chooses a number from 0 to 10 representing the name of the chosen configuration file. The game parses the configuration file and stores the data necessary for drawing the maze and then the drawing process starts.

The drawing process takes the parsed data and first draws the floor, then the pillars, the walls, and the treasure are drawn in the environment. Next, the enemies are drawn to their position and animated, followed by the Explorer himself.

3.5 Collisions

Multiple collisions are happening between entities and between entities and objects that are detailed in the following table:

Collision	**Effect**
Explorer - Wall	Explorer is blocked as he cannot go through walls.
Explorer - Guardian	The Explorer loses health. If he collides 5 times, the game ends with a loss.
Explorer - Treasure	The game ends with a win.
Projectile - Wall	The wall is stronger, so the projectile doesn't pass through.
Projectile - Guardian	The Guardian is eliminated.

Table 1: Collision table

4 CONCLUSIONS

The Survival Maze Game has been developed as a practical demonstration of using OpenGL for real-time 3D game development. This project highlights the fundamental aspects of graphical programming, including rendering, shading, input handling, and physics, all while maintaining a focus on efficient performance and smooth user interaction.

While there is always room for further optimization and feature development, The Survival Maze Game demonstrates a solid foundation in OpenGL-based game programming. It has been a valuable learning experience and a step towards more advanced game development projects in the future.

Thank you for taking the time to explore this documentation.

Bibliography

[1] EGC team @ Polytechnic University of Bucharest. Tema 2 - Survival Maze. https://ocw.cs.pub.ro/courses/egc/teme/2021/02. Latest access: 08.09.2024.

[2] Mozilla Foundation. 3D collision detection. https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_collision_detection. Latest access: 08.09.2024.

[3] EGC team @ Polytechnic University of Bucharest. Camera implementation. https://ocw.cs.pub.ro/courses/egc/laboratoare/05. Latest access: 08.09.2024.

[4] EGC team @ Polytechnic University of Bucharest. Shader models. https://ocw.cs.pub.ro/courses/egc/laboratoare/06. Latest access: 08.09.2024.