 Project work

Documentation For the Assignment

Submitted by – James Limbu (23085133)

Submitted to - Aashish Acharya

Contents

Introduction & Features --------------------------------------------------------------------- 1

Code Structures & Saving and Loading ------------------------------------------------- 2

Detailed Component Description ---------------------------------------------------------- 3

Game Controls --------------------------------------------------------------------------------- 4

Flow Chart -------------------------------------------------------------------------------------- 6

Conclusion --------------------------------------------------------------------------------------- 7

**Introduction**

* Maze of terror is user-friendly based game which is developed in C++ where user can go through randomly generated maze, collect coins, avoid enemies, and aims to reach the exit. The games become harder according to level the user reaches. If player reaches level 2 then there will be a greater number of enemies and the more level increase the more game adds enemy.

**Features**

* Random maze generates when player reaches new level
* Enemy AI with chasing and finding way out of the maze
* Coin collection and score tracking system
* Save and Load function available
* Level progression according to the player
* Threaded enemy movement for dynamic gameplay

**CODE STRUCTURE**

The code is organized in following ways:

**Entity Class**

* A base class representing any game object with coordinates (x,y) and a direction

Player Class

* Derived from Entity, it handles movement, score, and level progression

Enemy Class

* Derived from Entity, uses basic AI to follow or randomly move around the maze.

Exit Class

* Derived from entity, represents the goal and calculates initial direction based on path finding,

Maze Generation

* Creates a randomized maze with walls, coins, and a reachable exit.

Game Loop

* Main control loop that handles drawing, user input, and checks for game events.

Multithreading

* Enemies move independently in a separate thread.

Saving and Loading

* The game state (player, maze, enemies, and exit) is serialized and stored in file.

**Detailed Components Descriptions**

**Maze Generation**

* The maze is generated with outer walls and randomly placed internal walls. Coins are placed in empty locations. A BFS algorithm ensures the player can reach the exit before finalizing the maze layout.

**Player Mechanics**

* The player can move in four directions using WASD keys. On collecting coins, the score increases. Reaching the exit progresses to the next level, which generates a new maze and resets position**.**

**Enemy AI**

* Enemies move every 500 milliseconds. They have a 60% chance to follow the player using directional logic and a 40% chance to move randomly, making the game unpredictable.

**Game Saving and Loading**

* The current state of the game including the player, enemies, maze, and exit is saved in a binary file. The user can resume the game later by loading the saved state.

**Multithreading**

* Enemy movement is handled in a separate thread, allowing smooth and real-time updates even while the player is idle. Mutex locks ensure safe concurrent access to shared game data.

**Game Controls**

**W – Move Up**

**A – Move Left**

**S – Move Down**

**D – Move Right**

**P – Save Game**

**Q – Quit Game**

**Flow Chart Describing Game Structure:**

**Initialize Game**

Main Menu

Spawn Players & Enemies

Generate Maze

Load Saved Data

Game Loop

Draw Maze & HUD

Get Input

Move Player

Save Game

Check Collision

Game over

Add Score

Next Level

**Conclusion:**

Maze of Terror offers a fun, challenging gameplay experience with gradually increasing difficulty and dynamic enemy AI. The game utilizes various C++ features including OOP, file I/O, multithreading, and STL containers to deliver a robust console game.

Here is short clip describing the game for further information:

