BENHA UNIVERSITY

System Analysis Project
Second Year
2023-2024

جامعة بنها
كلية حاسبات و ذكاء اصطناعي

CG Project

- ***3D Maze Game***

- ***Dr. R. Orban***
- ***Eng.Doha Abdel Waha***

1) ***محمد محمود السيد إبراهيم***
2) ***محمد علاء احمد فوزي***
3) ***محمد السيد حسين إبراهيم***
4) ***سندس خالد حسين سيد***
5) ***اية خالد محجوب***
6) ***امل سعيد عبده***

🎮 3D Maze Game – Collect All Coins to Win:

(🧠 Project Overview)

This project is a simple 3D maze game developed using C++ and the OpenGL graphics library. The main objective of the game is for the player to navigate through a 3D maze, collect all the gold coins scattered throughout, and reach the exit point only after collecting every coin.purpose and then leave the queue thus the counter down count down one and if new individual come to queue counter up count up one.

The game includes essential elements of computer graphics such as:

3D world rendering
Camera control
Mouse and keyboard input handling
Lighting and textures
Collision detection
HUD (Heads-Up Display) for score and messages

It serves as a great beginner-level project to understand how interactive 3D environments are built using OpenGL and GLUT.

- 🧱 **Key Features of the Game**

🔷 **Maze Layout**

- The maze is defined as a 10x10 grid .
- Each cell can either be:
  - **1** – a wall (impassable)
  - **0** – an empty path (walkable)
- Walls are rendered using textured cubes.

🪙 **Gold Coins System**

- Coins are represented as rotating golden spheres with torus rings around them.

- When the player gets close enough, the coin is automatically collected.
- A score increases with each coin collected.
- The player must collect all coins before reaching the exit .

🚪 **Exit Point**

- The exit is shown as a yellow sphere located at the center of the maze.
- If the player tries to reach it without collecting all coins, a warning message appears.
- Once all coins are collected, the player can win by reaching the exit.

🖱️ **Camera and Mouse Control**

- The player navigates the maze from a first-person perspective .
- Pressing the spacebar toggles mouse capture mode:
  - In this mode, moving the mouse changes the camera’s direction.
  - The mouse is automatically re-centered to allow continuous rotation.
- Pressing V switches between first-person view and a top-down view of the maze.

💡 **Lighting and Visuals**

- Basic lighting system with one directional light source.
- Ambient light ensures visibility even in shadowed areas.
- Walls have a texture applied to them.

- If the texture file (**wall.bmp**) is not found, a fallback checkerboard pattern is used.

📊 Heads-Up Display (HUD)

- Displays real-time information:
  - Current score
  - Number of coins collected vs total
  - Victory message when the player wins
  - Warning message if the player reaches the exit too early

⌨️ How to Play

Keyboard Controls:

KEY	ACTION
↑ ↓ ← →	Move forward, backward, left, right
Space	Toggle mouse look mode (capture/uncapture)
V	Switch between first-person and top-down view
ESC	Exit the game

Gameplay Instructions:

1. Use the arrow keys to move through the maze.
2. Look around by capturing the mouse using Space or clicking.
3. Collect all coins that are visible in the maze.

4. You will not be able to enter the final door until all coins are collected.
5. Reach the yellow sphere (exit) to win the game.

🏆 **Winning the Game:**

- Once all coins are collected, go to the yellow sphere.

✅ Summary

This 3D Maze Game is a complete educational project that introduces key concepts in computer graphics and interactive programming:

- Drawing 3D objects and applying textures
- Implementing camera and lighting systems
- Handling user input (keyboard and mouse)
- Managing collisions and game logic
- Displaying real-time feedback via a simple HUD

It provides a solid foundation for beginners to explore more advanced topics like:

- Adding sound effects
- Creating multiple levels
- Using shaders with modern OpenGL
- Improving UI design

📁 Notes for Running the Game

- Make sure the file **"wall.bmp"** exists in the same folder as the executable to load the wall texture.

- If the texture file is missing, the game uses a default black-and-white checkerboard pattern.
- The game runs using the GLUT library and works on any C++ development environment that supports OpenGL.

Let me know if you'd like me to generate this as a downloadable Word (.docx) file or add additional sections like:

- Screenshots of gameplay
- Code flow diagram
- Suggestions for future improvements