3D Coin Collector / Treasure Hunt Game

1. Game Title & Team Members

Game Title

3D Coin Collector / Treasure Hunt Game

Course Information

- **Course:** CSC-317 – Computer Graphics
- **Engine:** Unity Game Engine
- **Programming Language:** C#

Team Members

- **Student 1:** Khaled Hassan Ezz
- **Student 2:** Ranin Mohamed
- **Student 3:** Roqia Shehab

---

2. Tools & Frameworks Used

The development of this project relied on a combination of modern game development tools and frameworks to achieve both functional gameplay and visually appealing graphics:

- **Unity Game Engine:** Used as the main development environment for building, rendering, and testing the game.
- **C# Programming Language:** Used for implementing gameplay logic, player control, collision handling, and game state management.
- **Unity Standard Assets:** Utilized for the third-person character controller and animation setup.
- **NavMesh & NavMeshAgent:** Used to implement enemy artificial intelligence and patrol behavior.
- **Unity UI System:** Used to display score information and win messages on the screen.
- **Visual Studio:** Used as the code editor for writing and debugging C# scripts.

---

3. Game Overview & Objective

Game Overview

The *3D Coin Collector / Treasure Hunt Game* is a third-person 3D game developed using Unity. The player navigates a graveyard-style environment, collects coins scattered across the level, avoids enemies, and aims to reach the final checkpoint to win the game.

Game Objective

The main objectives of the game are:

- Collect all coins available in the level.
- Unlock a hidden key after collecting all coins.
- Use the key to open the main gate.
- Avoid enemy characters patrolling the area.
- Reach the checkpoint to complete the game successfully.

---

4. Graphics Techniques Used

4.1 Lighting

Unity's built-in lighting system is used to illuminate the environment. A **Directional Light** simulates natural sunlight, providing realistic shadows and depth perception. Lighting enhances the visibility of objects such as coins, enemies, and environmental assets.

4.2 Camera Transformations

The game uses a **third-person camera** that follows the player character. The camera movement is dynamically adjusted based on the player's direction and position, allowing smooth rotation and better spatial awareness. Camera transformations enhance immersion and gameplay control.

4.3 Texture Mapping

Texture mapping is applied to all 3D models, including:

- Ground surface
- Walls and environment structures
- Coins and interactive objects
- Enemy and player characters

Textures improve realism by adding surface detail and visual variety to the scene.

4.4 Shaders

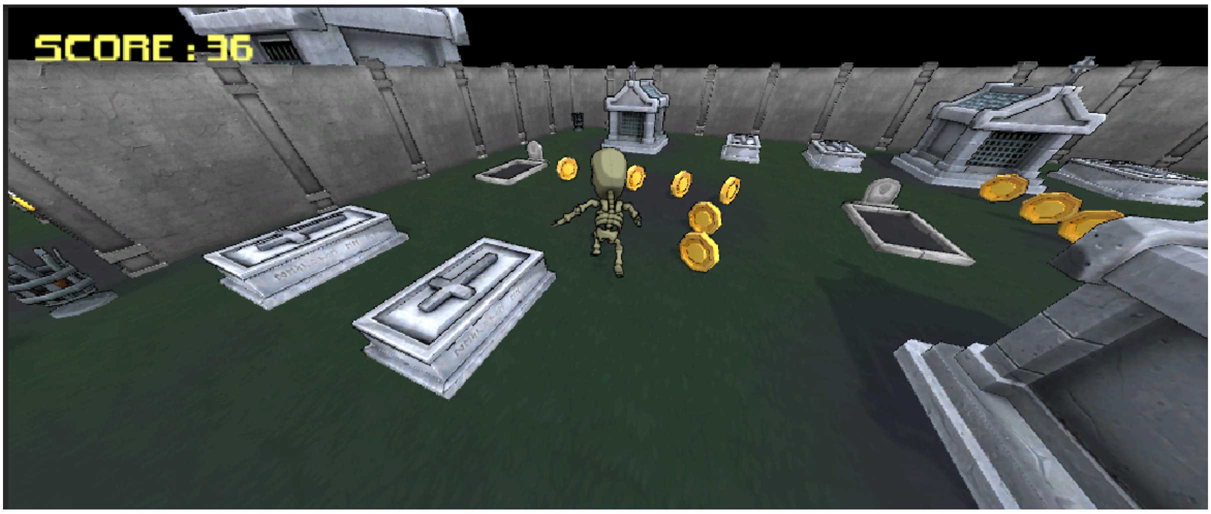
The game uses Unity's **default standard shaders**. No custom shaders were implemented. These shaders handle material properties such as color, light reflection, and shading, which are sufficient for this project's scope.

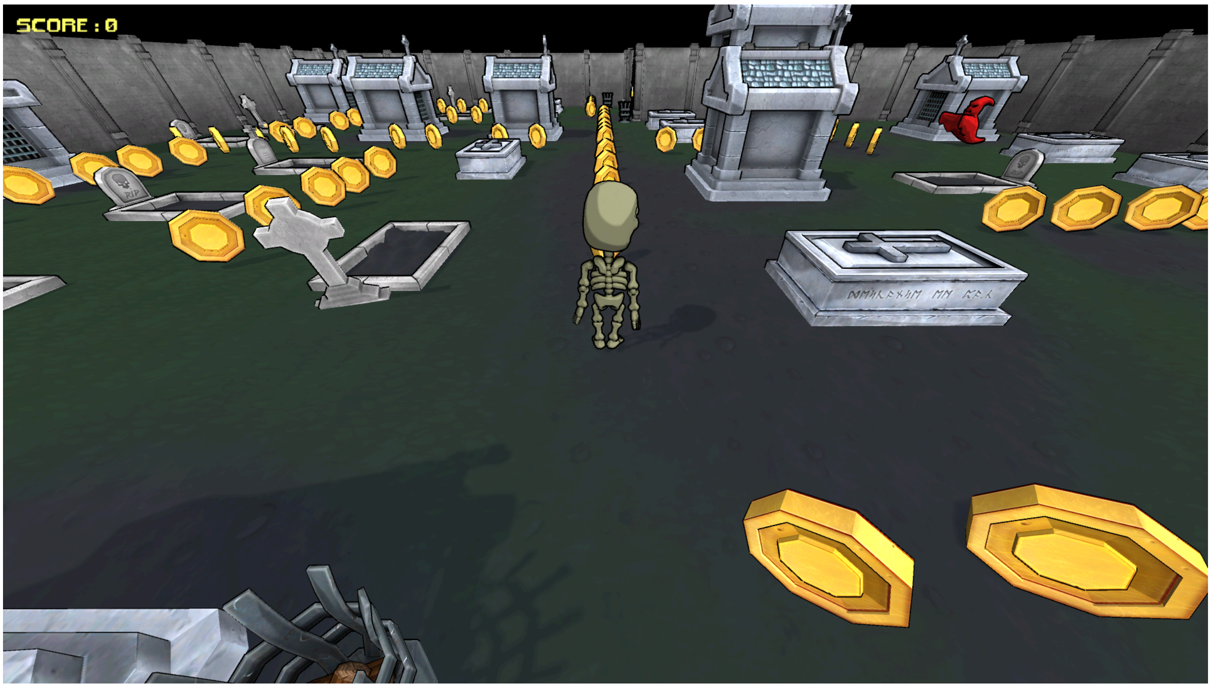
4.5 Animations

Character animations include idle, walking, running, and jumping actions. Animations are controlled using Unity’s **Animator Controller**, which ensures smooth transitions between different movement states. Root motion is applied for realistic character movement.

5. Screenshots of the Game

The following screenshots demonstrate key aspects of the game:





6. How to Run the Game

System Requirements

- Windows PC
- Unity Hub installed
- Compatible Unity version

Steps to Run

1. Open **Unity Hub**.
2. Select the appropriate Unity version.
3. Open the project folder.
4. Load the main game scene.
5. Click the **Play** button in the Unity Editor.

Game Controls

- **W / A / S / D:** Move the player
- **Space:** Jump
- **C:** Crouch
- **Shift:** Walk slowly

---

7. Division of Work Among Team Members

Khaled Hassan Ezz

- Player movement and control system
- Coin collection and scoring system
- Enemy patrol logic
- Game state management (Win / Game Over)
- Integration of UI elements
- Testing and debugging

Ranin Mohamed

- Level design and environment setup
- Placement of coins and obstacles
- Lighting configuration
- Asset organization

Roqia Shehab

- Character and environment assets integration
- Animation setup and testing
- Audio effects integration
- Gameplay testing and refinement

---

8. Conclusion

This project successfully demonstrates fundamental concepts of computer graphics and game development using Unity and C#. It integrates player control, basic AI behavior, collision detection, lighting, textures, and animations into a complete playable game. The project meets the learning objectives of the CSC-317 Computer Graphics course and provides a strong foundation for future enhancements such as additional levels, advanced AI behavior, and improved visual effects.

---

9. Future Improvements

Possible future enhancements include:

- Adding multiple levels
- Implementing a health system
- Introducing advanced enemy AI behaviors
- Enhancing graphics using custom shaders
- Adding a pause menu and settings screen