se23ucse139 - P Sai Manohar

Se23ucse185 – V Roop Shekar

Se23ucse229 – K Sai

Se23ucse179 – Harsha vardhan

Se23uari143 – Varun Goud

FOUNDATIONS OF IMMERSIVE TECHNOLOGY

3-D Ball Game

PROJECT REPORT

This project will be targeting the design and implementation of a simple multi-level 3D game using Unity game engine. The idea is to show basic concepts of immersive technology such as 3D interaction, physics-based translation, collision detection, and handling of scene and interface. The game will have a ball controlled by keyboard keys which will have to move through some ramps and platforms to reach a finish mark. The game will have different levels, a respawn mechanism, timer, and a win screen.

1. Introduction

Immersive technology can be described as creating an environment where interaction with digital inputs is accomplished in real time. Platforms such as Unity engines can be used to create such environments utilizing 3D graphics, physics, and interface systems.

Its objective will be to create a simple but working 3D game in Unity which can showcase immersive interaction with immediate feedback.

2. Object

The aims of this project are:

- The project aims to create a 3D Interactive Game using Unity
- For physics-based player movement
- To design different game levels
- To implement collision detection and triggers
- Add elements of UI such as Timer and Win Screen
- To grasp scene management and game flow

3. Tools & Technologies Used

- Game Engine: Unity 2022 LTS
- UI System: Unity Canvas & TextMeshPro
- Physics Engine: Unity Physics (Rigidbody & Colliders)

4. System Design

4.1 Player Controls

A 3D ball game object symbolizes the player. The Unity game engine’s Rigidbody component is used in the game for simulation of physics in ball movement. The keyboard is used to push the ball.

4.2 Camera System

A third-person camera will follow the player through gameplay with fluid motion, giving a good view of the game environment.

4.3 Level Design

Each level is composed of platforms, ramps, and obstacles made using basic 3D shapes. The level difficulty is raised in the succeeding level with alterations in the structure.

5. Game Features

5.1 Respawn

In case the player loses balance and steps out of the platform, the ball will reappear at a predefined point called a spawn point.

5.2 Finish Trigger

A trigger collider is added at the end of each level. As soon as the player enters the end of a level:

•If it is not the last level, the next scene loads.

•If it is the final level, the win screen will appear.

5.3 Multi-Level

Every level is a separate Unity scene. SceneManager is used to load the subsequent level in accordance with the build index order.

5.4 Timer System

Timer implementation is achieved through Time.deltaTime and rendered via TextMeshPro UI. Timer persists in levels via DontDestroyOnLoad.

5.5 Win Screen and Restart

A win screen appears when all levels have been achieved. A restart button is available to give a player another chance to begin playing from level one.

6. Implementation Details

- Scene Management: SceneManager
- Game Pause : This is achieved by using Time
- Persistent Objects, DontDestroyOnLoad
- User Interface Interaction: Button OnClick event

7. Challenges Faced

- MPI control of canvas and TextMesh pro

- The problem of debugging trigger detection
- Handling Time.timeScale during a switch of scenes
- Continuation of the timer in multiple levels

Such issues were fixed using debugging, optimization, and setting up a scene.

8. RESULTS

•Smooth player movement

• Correct level transitions

•Serial Respawn System

• Timer display accuracy

• Correct win screen activation

The game is working well and achieving all project requirements successfully.

9. Conclusion

In this project, practical implementation of concepts of immersive technology using Unity is gained. While working on a multilevel 3D game, important concepts such as physics simulation, handling scenes, and handling UI have been learned and implemented successfully. Further implementation of this project can be achieved by incorporating sound effects, complex obstacles, scoring, and virtual reality capabilities.

10. Future Enhancements

•Add sound effects and background music

•Introduce moving and rotating obstacles

• Add score-based evaluation

•Endorse Unity XR Toolkit for VR implementation