



TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
A Project Report
On
2.5D game using Unity

Submitted By:

Nishan Shrestha (HCE081BEI027)

Submitted To:

Department of Electronics and Computer Engineering

Himalaya College of Engineering

Chyasal, Lalitpur

Feb-22, 2026

ACKNOWLEDGMENT

We express our sincere gratitude to all those who have supported and guided us throughout the process of conducting this report on 2.5D pong game using unity. This endeavor would not have been possible without their valuable contributions and assistance.

The authors extend their heartfelt thanks to our supervisor, Sushant Pandey, whose expertise and guidance have been instrumental in shaping the direction of this research. Their insightful feedback and continued encouragement have been invaluable in refining the scope and focus of this study.

We also would like to thank our friends and colleagues for their meaningful discussions, brainstorming sessions, and moral support. Your input has enriched the quality of this report and helped us navigate challenging concepts.

This work would not have been possible without the collective efforts of these individuals and organizations. Although any shortcomings in this report are solely our responsibility, their contributions have significantly enriched its content.

Thank you.

Nishan Shrestha (HCE081BEI027)

ABSTRACT

This project focuses on developing a 2.5D game using Unity. The game features a player-controlled paddle, ball mechanics, scoring, etc. The primary goal of the project is to create a fun, engaging, and interactive game. The game includes object collision, rigid body, different scripts for different objects and UI.

Table of Contents

ACKNOWLEDGMENT	i
ABSTRACT	ii
List of Figures	iv
List of Tables	v
List of Abbreviations	vi
1. Introduction	1
1.1 Background Introduction	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Scope	1
2. Literature Review	3
2.1 Game Development with Unity Engine	3
2.2 Understanding 2.5D Game Design	3
2.3 Polygon Filling	3
2.4 Physics and Collision Detection in Unity	3
2.5 Artificial Intelligence (AI) in Games:	3
3. Methodology	4
3.1 Development Environment and Tools	4
3.2 Game Design and Mechanics	4
3.3 Implementation Details	4
3.4 Testing Strategy	5
3.5 Screenshots	5
4. Result and Analysis	7
4.1 Implemented Features	7
4.2 Performance Analysis	7
4.3 Challenges and Solutions	7
5. Conclusion and Future Enhancement	8
5.1 Conclusion	8
5.2 Future Enhancements	8
A. Appendices	9
A.1 Key Controls (Example)	9
Bibliography	10

List of Figures

Figure 3.1	Game Environment after asset creation	5
Figure 3.2	Collider, bouncing material and rigidbody used in ball object . . .	6

List of Tables

Table 3.1	Core Script Functionality Table for 2.5D Pong Game	5
-----------	--	---

1. INTRODUCTION

Developing a 2.5D Pong game using Unity involves combining 2D gameplay mechanics with 3D visual presentation, a common approach in modern game development to create immersive experiences while retaining simplified gameplay [1]. Unity, as a versatile game engine, supports both 2D and 3D game creation, making it suitable for such projects. The core principle of a 2.5D game often involves rendering 2D gameplay elements within a 3D environment, leveraging perspective cameras and 3D models to add depth and visual appeal.

1.1 Background Introduction

For a 2.5D Pong game, the paddles and ball, while appearing as 3D objects, would typically be constrained to move within a 2D plane. This can be achieved by setting their Z-axis position to a constant value, effectively limiting their movement to the X-Y plane. Implementing the physics for such a game often involves utilizing Unity's 2D physics system, including Rigidbody2D and Collider2D components, to ensure accurate collision detection and realistic movement behavior. This approach is crucial for maintaining deterministic physics, which is essential for games like Pong, where precise interactions are key to gameplay.

1.2 Motivation

Making a 3D game for a beginner is very hard. So, I made a game that looks 3D in visual but works in 2D mechanics. It is a good start for making games and learning to use unity.

1.3 Objectives

The main objectives of the project are listed below:

- To learn the basic use of Unity 2D and 3D platform.
- To create a playable pong game.
- To learn basics of C.
- To learn positioning, scaling, linking, etc.

1.4 Scope

This project covers fundamental Unity Game development concepts:

- **Game Engine Proficiency:** A primary scope involves demonstrating proficiency in using the Unity game engine, including its interface, asset management, and scripting environment. This includes understanding how to set up a project, import assets, and configure game objects.
- **C Scripting:** The project will extensively use C for implementing game logic. This includes scripts for paddle movement, ball physics (e.g., collision detection, rebound mechanics, speed adjustments), and scoring systems.
- **2.5D Implementation:** A significant part of the scope is the successful creation of a 2.5D visual style. This involves rendering 2D gameplay elements within a 3D environment, using perspective cameras to simulate depth

2. LITERATURE REVIEW

2.1 Game Development with Unity Engine

Unity stands as a leading game engine widely utilized across various platforms and game genres. [2][3] Its architecture provides an integrated development environment (IDE) that facilitates asset management, scene configuration, and scripting. The C programming language is central to Unity development, enabling the creation of intricate game logic, object behaviors, and interactions.[4]

2.2 Understanding 2.5D Game Design

In game design, the term "2.5D" describes the incorporation of 2D gameplay into a 3D visual environment. By fixing the Y-axis position, this method limits player or object movement to a 2D plane, usually the X-Z axis, while frequently using perspective cameras and 3D models to give the appearance of depth.[1]

2.3 Polygon Filling

Filling polygons is essential for rendering solid shapes. A common technique is the scanline fill algorithm, which computes intersections of polygon edges with horizontal scanlines and fills between pairs of intersections.

2.4 Physics and Collision Detection in Unity

Games like Pong need physics that are both accurate and predictable. Unity's 2D physics system is very helpful for making objects interact in a realistic way. For game objects, this system mostly uses **Rigidbody2D** and **Collider2D** parts. **Rigidbody2D** lets physics forces and collisions affect objects, and **Collider2D** components set the shape for collision detection. Setting these parts up correctly makes sure that the game always has the same results.

2.5 Artificial Intelligence (AI) in Games:

For computer opponents, AI can range from simple reactive behaviors to more complex predictive algorithms, enabling varying difficulty levels and engaging single-player experiences.

3. METHODOLOGY

3.1 Development Environment and Tools

The project will be developed using the Unity Engine (Unity 2022.3.62f3). The primary programming language will be C#. Version control will be managed using Git to facilitate collaborative development and track changes.

3.2 Game Design and Mechanics

The game will feature two paddles, one controlled by the player and the other by an AI. Paddle movement will be restricted to vertical (Y-axis) translation within defined boundaries. The ball will possess physics-based movement, including rebound angles calculated upon collision with paddles and walls. Scoring will occur when the ball passes the opponent's paddle. The 2.5D aesthetic will be achieved through a static perspective camera and 3D models for the paddles, ball, and arena, with their movement constrained to the X-Z plane to maintain 2D gameplay.

3.3 Implementation Details

- **Asset Creation:** 3D models for paddles, the ball, and the game environment (arena, background) will be either custom-created or acquired from Unity Asset Store, ensuring they align with the 2.5D visual style.
- **Scripting:** Core game functionalities will be managed by C# scripts. Key scripts will include `PlayerController` for player input and paddle movement, `BallPhysics` for handling ball movement and collision logic, and `GameManager` for managing game states, scoring, and AI logic. All time-dependent operations, such as movement, will utilize `Time.deltaTime` to ensure frame-rate independence.
- **Physics Implementation:** `Rigidbody2D` components will be attached to the ball and paddles to enable physics interactions. `Collider2D` components (e.g., box colliders for paddles and a circle collider for the ball) will be configured to detect collisions accurately. Physics materials may be used to control friction and bounciness.
- **UI Development:** The user interface will be built using Unity's Canvas system. A scoreboard displaying player and AI scores will be prominently featured. Other UI elements, such as a start screen, pause menu, and game-over screen, will be implemented. The UI will be rendered in World Space for seamless integration with the 3D environment.
- **AI Implementation:** For the AI opponent, a basic reactive algorithm will be

implemented initially. This AI will track the ball's Y-position and attempt to move its paddle to intercept it. Further iterations may explore predictive AI to anticipate ball trajectory.

Script Name	Purpose
PlayerController	Handles player input and paddle movement
BallPhysics	Manages the ball's movement, collision and rebound logic.
GameManager	Controls overall game states, scoring, game setup, and AI.
AIScript	Implements the AI opponent's movement logic.

Table 3.1: Core Script Functionality Table for 2.5D Pong Game

3.4 Testing Strategy

Testing will be conducted iteratively throughout the development process. Unit tests will be performed on individual scripts (e.g., paddle movement, ball physics calculations). Integration tests will verify the interactions between different game components. Playtesting will be essential to assess gameplay balance, responsiveness, and overall user experience. This includes testing collision accuracy, AI behavior, and UI functionality across different simulated conditions.

3.5 Screenshots

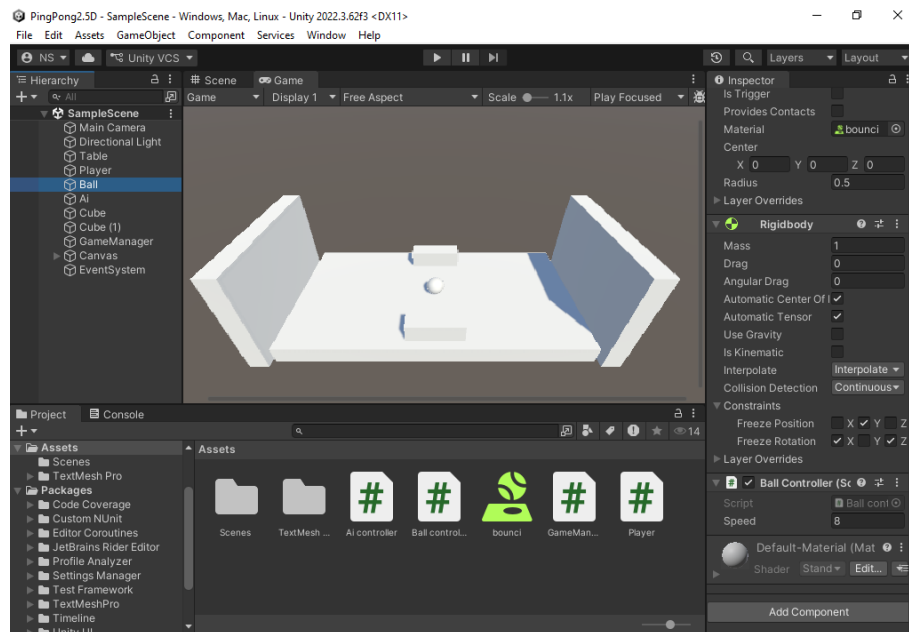


Figure 3.1: Game Environment after asset creation

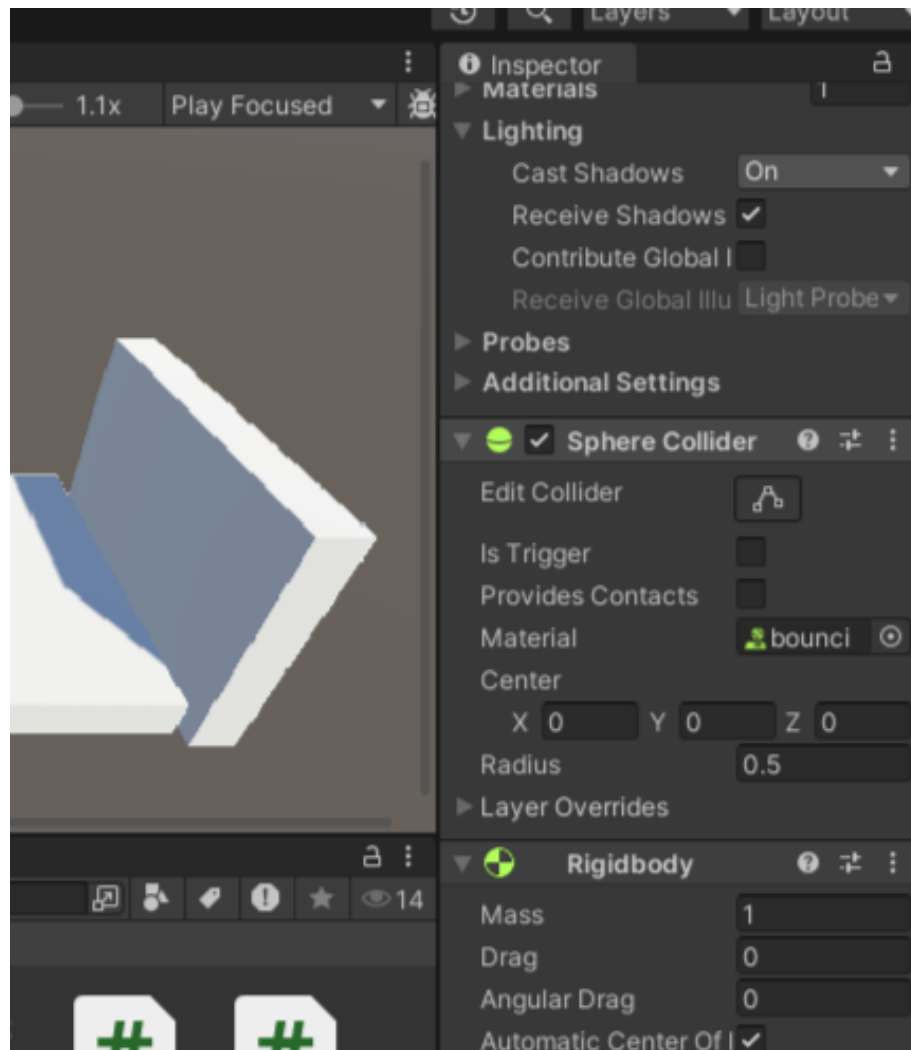


Figure 3.2: Collider, bouncing material and rigidbody used in ball object

4. RESULT AND ANALYSIS

4.1 Implemented Features

The developed 2.5D Pong game will showcase fully functional core mechanics, including responsive paddle controls for the player and a basic AI opponent. The ball will exhibit realistic physics-based movement, including deflections upon collision with paddles and walls. A clear and intuitive UI will display scores and provide essential game feedback. Visuals will demonstrate the intended 2.5D aesthetic, leveraging 3D models and perspective camera views.

4.2 Performance Analysis

Frame rate stability, CPU/GPU usage, and memory footprint will all be used to judge performance. To make sure that the game runs smoothly on all target hardware, optimization methods like efficient asset loading, batching, and the right use of physics calculations will be used. Any performance problems that are noticed and how they were fixed will be written down.

4.3 Challenges and Solutions

During development, challenges may arise in areas such as fine-tuning ball physics (e.g., consistent rebound angles, speed acceleration), balancing AI difficulty, and optimizing rendering for the 2.5D effect. Solutions could involve iterative adjustments to physics parameters, refining AI logic, or implementing specific shaders and post-processing effects within Unity. For instance, ensuring deterministic physics in Pong requires careful management of time steps and collision handling

5. CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

This project successfully outlines the development of a 2.5D Pong game using Unity, demonstrating the integration of classic gameplay with contemporary visual design. The core objectives of implementing functional mechanics, an AI opponent, and an effective UI within a 2.5D environment are addressed through the proposed methodology. The use of Unity's extensive tools and C scripting facilitates a robust and engaging game experience.

5.2 Future Enhancements

- **Advanced AI:** Implementing more sophisticated AI algorithms that utilize machine learning techniques for predictive ball tracking and adaptive difficulty[3]
- **Online Multiplayer:** Integrating networking solutions, such as Photon Unity Network, to enable competitive online multiplayer functionality.[5]
- **Customizable Game Modes:** Introducing options for different game rules, paddle sizes, ball speeds, or environmental obstacles.

A. APPENDICES

A.1 Key Controls (Example)

- **Arrow-left/Arrow-right:** moves paddle left/right

Bibliography

- [1] H. Y. Sutanto, J. Pragantha, and D. A. Haris, "Pembuatan game action-adventure 2.5d "dysphoria: Delirious realm" dengan unity," *Jurnal Ilmu Komputer dan Sistem Informasi*, 2020.
- [2] W. Ke, "Leveraging unity for 2d pixel game development: Techniques and best practices," *ITM Web of Conferences*, 2025.
- [3] J. Li and J. Morris, "Ai-driven simplification of 3d animation: Bridgingthe gap between 2d and 3d with a unity package for predictive pose generation and stream-lined workflows," *Artificial Intelligence and Big Data*, 2024.
- [4] G. Grandy, J. Pragantha, and D. A. Haris, "Pembuatan game 2d rhythm running "catch the beat" pada platform android," *Jurnal Ilmu Komputer dan Sistem Infor-masi*, 2021.
- [5] G. Tiara, H. Husaini, and M. Mahlil, "Synchronous multiplayer shooting in the 2d game "war cops" using unity engine," *Journal of Artificial Intelligence and Soft-ware Engineering (J-AISE)*, 2024.