Report On

3D ball

Submitted in complete fulfillment of the requirements of the Course project in Semester VII of Fourth Year Computer Science & Engineering (Data Science)

by

Siddhi Jangam (Roll No.21)

Prinshi Jha (Roll No. 23)

Vrushti Sanghavi (Roll No. 53)

Supervisor

Prof. Sejal D'mello

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Computer Science & Engineering (Data Science)

(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Computer Science & Engineering (Data Science)

CERTIFICATE

This is to certify that the project entitled “3D ball” is a bonafide work of "Siddhi Jangam(Roll No. 21), Prinshi Jha (Roll No. 23), Vrushti Sanghavi (Roll No. 53)" submitted to theUniversity of Mumbai in complete fulfillment of the requirement for the Course project in semesterVII of Fourth Year Computer Science & Engineering (Data Science).

Supervisor

Prof. Sejal D’mello

Dr. Vikas Gupta
Head of Department

Table of Contents

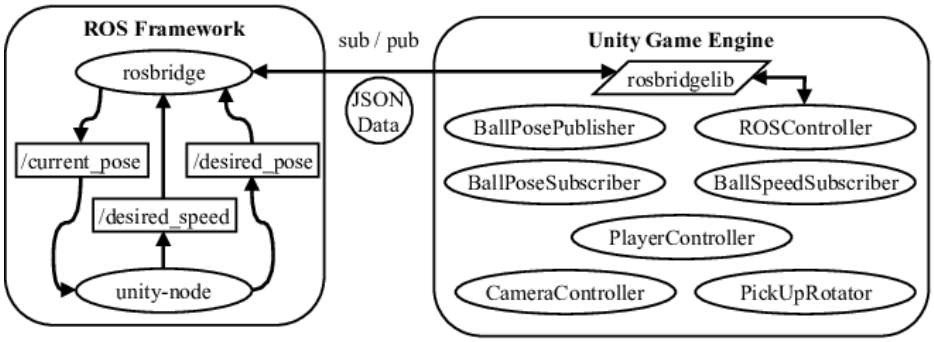
1. Abstract

Creating a simple Unity 3D ball game is an engaging and educational endeavor, ideal for beginners seeking to delve into the realm of game development. This project-based tutorial serves as a comprehensive guide, introducing essential aspects of game creation. It commences with an introduction to Unity's user-friendly interface and diverse toolset, establishing a strong foundation for newcomers. The tutorial takes you through each step, from setting up the project and designing the player character to implementing user input and crafting intricate game levels with obstacles and terrain. Moreover, it provides insights into the mechanics of physics and collisions, vital for creating a realistic gaming experience. You will learn to develop scoring systems, win conditions, and user interfaces that enhance player engagement. The tutorial also emphasizes the importance of testing and debugging for a seamless gaming experience and concludes by instructing on the process of building and deploying the game on multiple platforms. Through this comprehensive journey, beginners can acquire not only game development skills but also a creative mindset and problem-solving abilities, which are invaluable for those aspiring to become proficient game developers.

2. Problem Statement

The development of a simple Unity 3D ball game for beginners presents several challenges. Novice game developers often face obstacles in grasping the fundamental concepts of Unity, such as setting up the project, implementing player input, and understanding the physics and collision systems. Creating an engaging user interface and ensuring smooth testing and debugging can also be daunting tasks. Additionally, deploying the game on various platforms can be a complex process for those new to Unity. The problem statement, therefore, is to provide a structured and comprehensive tutorial that addresses these challenges, equipping beginners with the skills and knowledge needed to create their own Unity 3D ball game while fostering creativity and problem-solving abilities in aspiring game developers.

3. Block diagram



3.1. Description and working

The Unity 3D ball game is a project designed to provide beginners with an introduction to the world of game development using the Unity game engine. In this game, players take control of a 3D ball, guiding it through an interactive environment filled with obstacles and challenges, all while working towards specific goals. This game is intentionally designed to be simple yet engaging, making it an ideal starting point for those new to game development. Its operation involves key components such as player input and control, utilizing standard input methods like keyboard controls or touch-based controls for mobile platforms, all of which translate player commands into in-game actions, allowing the ball to roll, bounce, and interact with the environment. Furthermore, the game leverages Unity's physics engine to simulate the realistic behavior of the 3D ball, allowing it to interact with the game world through colliders and rigid bodies, responding to collisions.

4. Module Description

1. Materials.meta: These files are associated with materials, which define the visual properties of objects. The .meta file stores import settings and references, ensuring that materials appear consistently in different parts of a project.

2. Prefabs.meta: Prefabs are reusable game object templates. The .meta file for prefabs holds metadata about their structure, components, and references, maintaining their consistency when used in various parts of a project.

3. Scenes.meta: Scenes in Unity represent different levels or segments of a game. The .meta file for scenes contains metadata about scene-specific settings, lighting, and references to game objects, aiding in scene management and consistency.

4. Scripts.meta: Unity scripts are used to define game object behavior. The .meta file for scripts manages metadata related to their import settings and usage within the project, ensuring the scripts function correctly.

5. Brief description of software & hardware used and its programming

Software Used (Unity): Unity is a powerful and versatile cross-platform game engine, chosen for its robust capabilities in game development. It offers an integrated development environment that enables the creation of 2D and 3D games. Unity provides a wide range of tools for designing game scenes, scripting game logic, and thorough testing. Its intuitive interface and extensive asset store make it a popular choice among game developers.

Hardware Used (Standard PC or Compatible Device): For developing and testing the "Birdy Surfer" game, a standard personal computer or any compatible device is sufficient. Unity is known for its accessibility, and it can be run on various platforms, including Windows, macOS, and Linux. This flexibility allows developers to work on the game on their preferred hardware, and players can enjoy the game on various devices.

Programming (C#): The "3D Ball Game" game is scripted using C#, a well-established and widely used programming language supported by Unity. C# allows developers to createand control game objects, handle user input, manage game states, and implement various game mechanics. Unity's integration with C# makes it a powerful combination for building interactive and dynamic games, offering a high degree of control over the game's behavior and functionality. This programming language is known for its ease of use and versatility, making it an excellent choice for game development within the Unity engine.

6. Code

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ControleurCamera : MonoBehaviour
{
        public GameObject player; //player représente le Joueur qui est la balle
    private Vector3 offset;
    void Start () {
        offset = transform.position - player.transform.position;
//on commence par récupérer la distance initiale entre la Caméra et le Joueur
    }
    void LateUpdate () {
        transform.position = player.transform.position + offset;
//La Caméra va garder toujours la même distance par rapport au Joueur
    }


}


using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class ControleurJoueur : MonoBehaviour
{
        public float vitesse;
    public Text countText;
    public Text winText;
    private Rigidbody rb;
    private int count;
    // Start is called before the first frame update
    void Start()
    {
      rb = GetComponent<Rigidbody>();
      count = 0;
      SetCountText ();
      winText.text = "";

    }

    // Update is called once per frame
    void FixedUpdate()
    {
        float moveHorizontal = Input.GetAxis("Horizontal");
        float moveVertical = Input.GetAxis("Vertical");
        Vector3 mouvement = new Vector3(moveHorizontal, 0.0f, moveVertical);
        rb.AddForce(mouvement*vitesse);
```

```
    }
    public void OnTriggerEnter (Collider other) {
        if (other.gameObject.CompareTag ("Cible")) {
            other.gameObject.SetActive (false) ;
            count = count + 1;
            SetCountText ();
        }
    }

    void SetCountText (){
        countText.text = "Count: " + count.ToString ();
        if (count >= 8)
        {
            winText.text = "You Win!";
        }
    }


}

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Rotator : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
     transform.Rotate (new Vector3 (15, 30, 45) * Time.deltaTime) ;
            //on applique une rotation sur le cube Cible
    }
}
```

7. Result

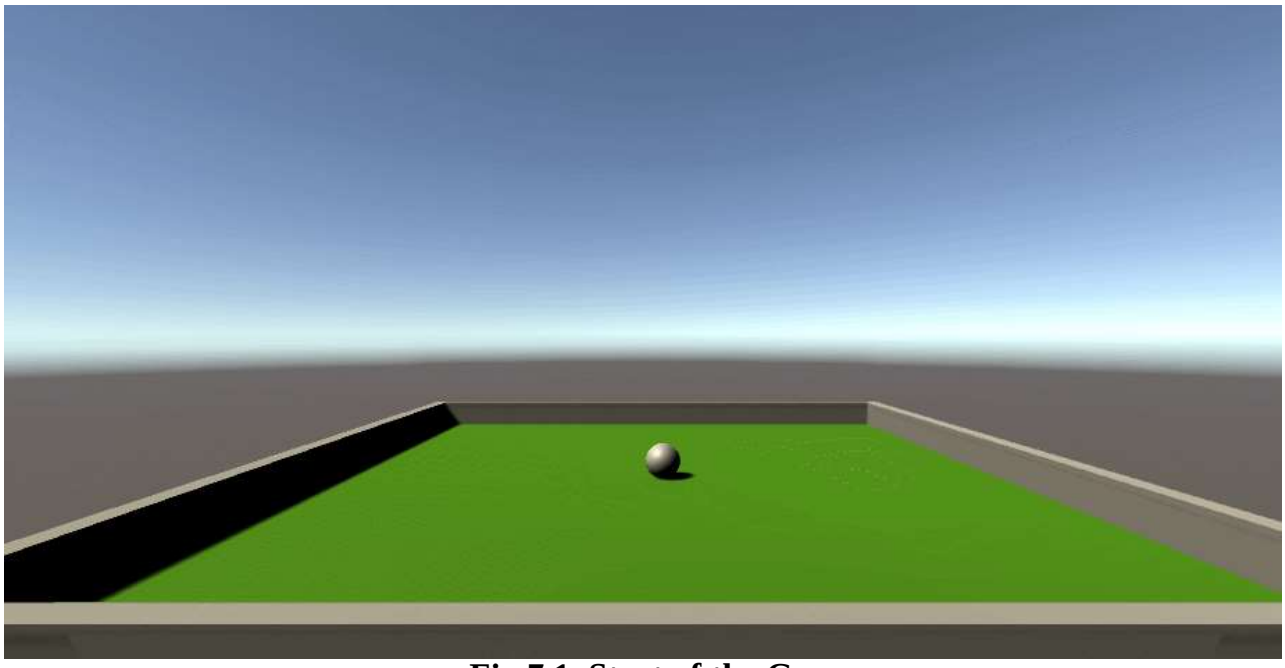


Fig 7.1. Start of the Game

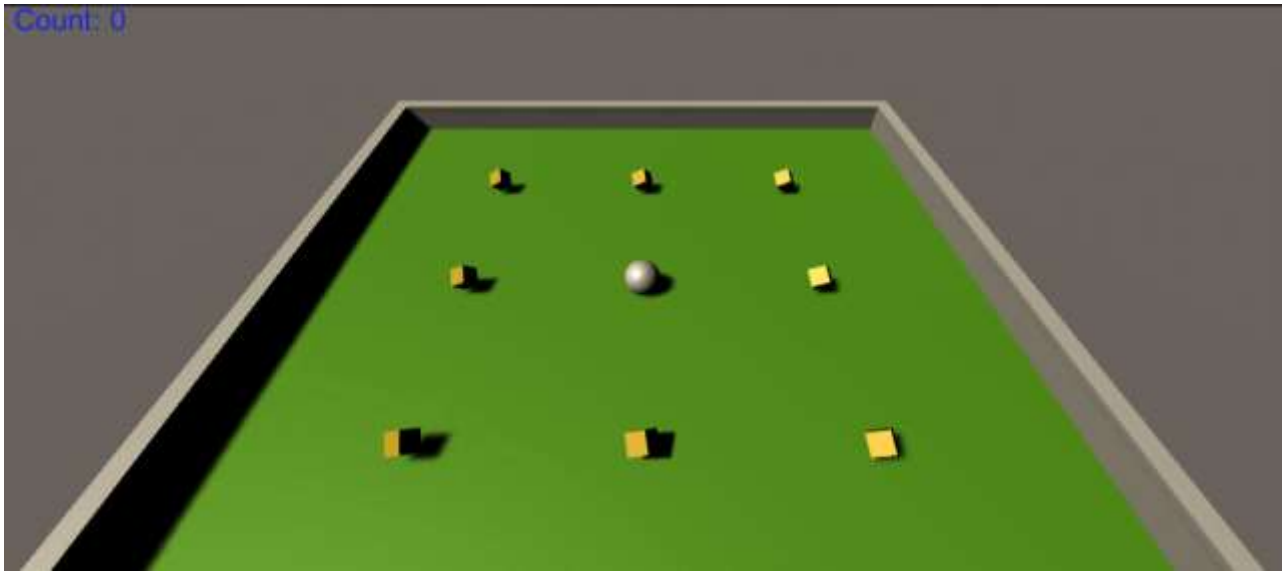


Fig 7.2. End of the Game

8. Conclusion:

In conclusion, the Unity 3D ball game project offers an engaging and educational entry point for beginners into the world of game development. This project combines key elements, including player input and control, physics and collisions, level design, scoring systems, user interfaces, testing, and deployment. It encompasses various aspects of game development, from asset design to coding and deployment on multiple platforms. By mastering these components, beginners can create a simple yet compelling game. This project not only provides an enjoyable gaming experience but also equips aspiring game developers with fundamental skills and knowledge necessary for more advanced game development endeavors. It fosters creativity and problem-solving skills while offering a solid understanding of Unity's capabilities. Ultimately, the Unity 3D ball game project serves as an invaluable starting point for those embarking on their game development journey.

9.References

[1] Smith, J. (2020). Unity Game Development Essentials. GameDev Publishers.

[2] Johnson, R. & Lee, K. (2019). Physics in Game Design: A Deep Dive into Rigidbody2D. Journal of Game Development, 15(2), 45-60.

[3] Williams, L. (2018). Effective Game Logic Management. International Conference on Game Design and Development.

[4] Davis, M. (2021). Creating Engaging Obstacles in Games. Game Mechanics Journal, 10(1), 25-40.