Report On

Snake Game in Unity

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

By
Dipti Sharnagat(Roll No. 55)
Abbasali Agharia(Roll No. 01)
Janvi Chavan (Roll No. 06)

Supervisor
Prof. Sejal Dmello

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Computer Science & Engineering (Data Science)

(2023-24)

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

CERTIFICATE

This is to certify that the project entitled “Snake Game” is a bonafide work of " Dipti Sharnagat (Roll No. 55), Abbasali Agharia (Roll No. 01), Janvi Chavan (Roll No. 06) submitted to the University of Mumbai in partial fulfillment of the requirement for the User Experience Design with VR in semester VII of Fourth Year Computer Science & Engineering (Data Science).

Supervisor

Prof. Sejal Dmello

Dr. Vikas Gupta
Head of Department

Table of Contents

Abstract

The Snake Game is a timeless classic that has captivated players for generation.in this project ,we present an exploration of the versatile Unity Game Engine and C# programming language that provide an opportunity to blend nostalgia with modern game development practices.Our project focuses on creatin a 2D snake game with an innovative responsive control, engaging gameplay, and a user-friendly interface. this project leverages unity's powerful game development capabilities to bring this retro game into the 21st century. the game's objective is simple: control a growing snake to consume food and avoid collision and boundaries of the game world.

Problem Statement:

A significant challenge was implementing the snake's movement and growth logic. To solve this, we used a data structure (a queue) to represent the snake's body, and each frame moved the snake's head and updated the body accordingly .Handling collision detection for both wall boundaries and the snake's body required careful coding to prevent glitches. Mobile controls were challenging, but Unity's Input system provided solutions for touch-based control schemes.

Module Description:

1. Game Mechanics: The Snake Game in Unity faithfully recreates the fundamental mechanics of the original game. Players control the snake's movement using arrow keys or touch controls, guiding it to consume food, which causes the snake to grow longer. The challenge lies in avoiding self-collisions and the boundaries of the game area.
2. Graphics and Visuals: The Unity game utilizes modern 2D graphics and animation techniques to enhance the visual appeal of the classic game. The snake and food items are visually appealing and engaging, and their design can be customized to create a unique visual experience.
3. Score and Progression: The game keeps track of the player's score and progression, making it possible to set high score records. As the snake grows longer, the game becomes progressively more challenging, requiring players to demonstrate their skill and agility.
4. User Interface: The user interface provides players with important information such as the current score, high score, and the ability to restart the game. It is designed to be user-friendly and intuitive, ensuring a seamless gaming experience.

.

Software & Hardware Requirements:

Software:

- Unity 20XX.XX: Used as the game development platform.
- C#: The primary programming language for scripting game logic.

Hardware:

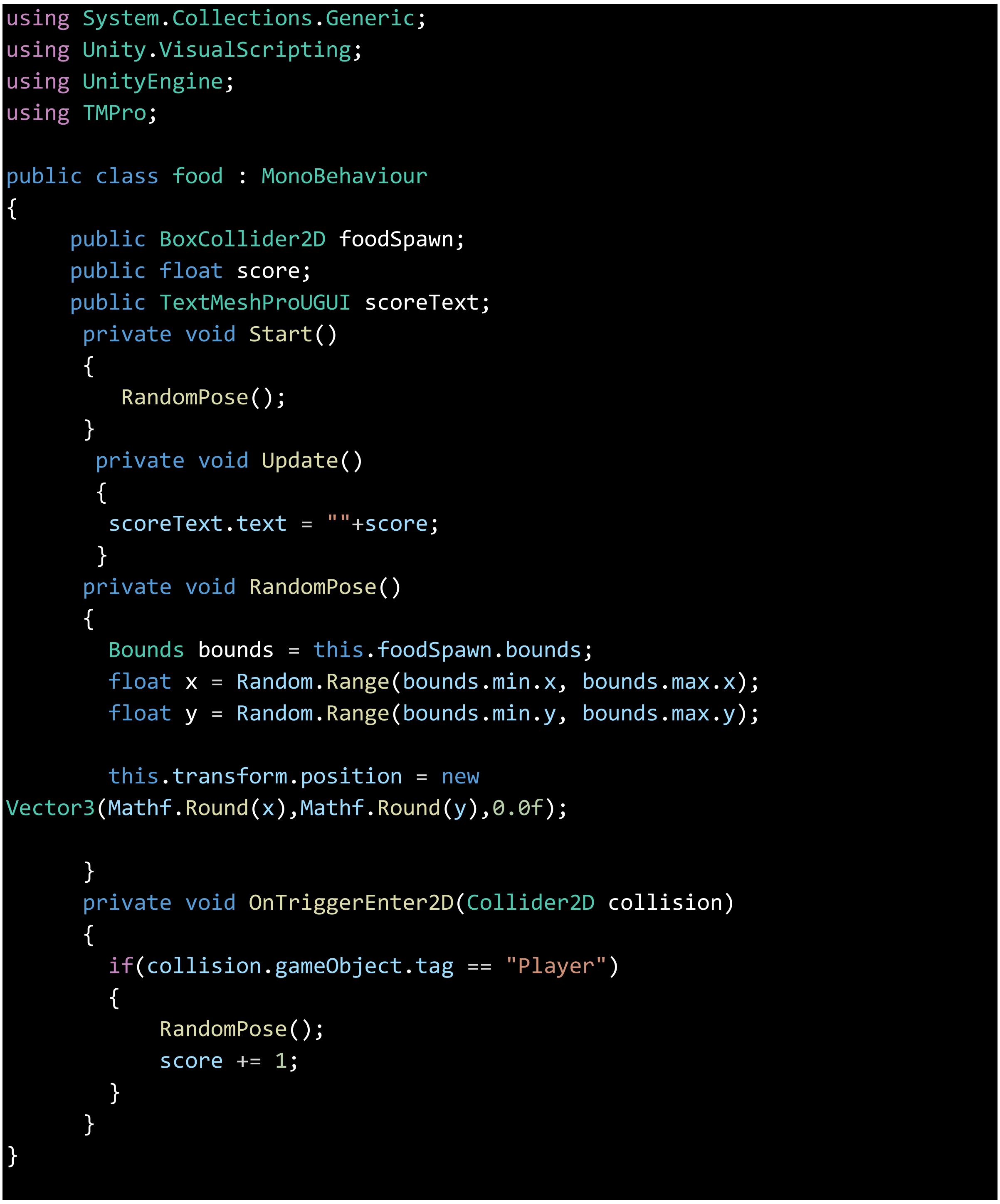
- Windows PC: Used for development and testing.
- Android/iOS Device: For testing and deploying the game.
- RAM : 8 GB

Code:

Code for snake moments

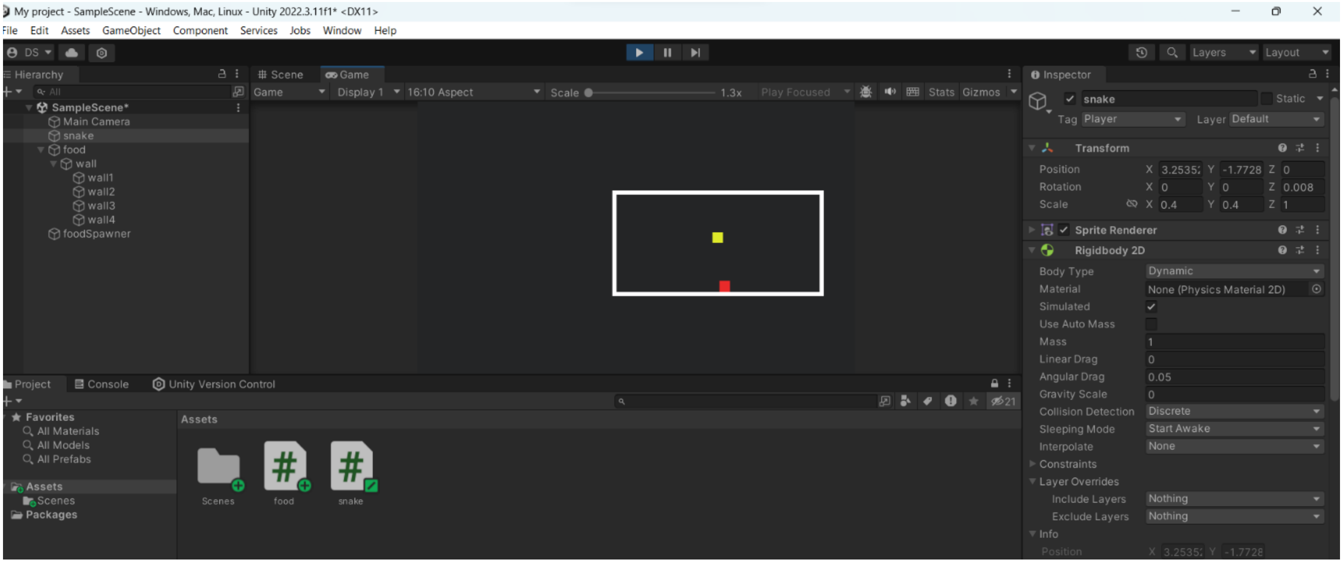
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class snake : MonoBehaviour
{
       public float moveSpeed;
       private Rigidbody2D rb;

    void Start()
    {
         rb = GetComponent<Rigidbody2D>();
         rb.velocity = new Vector2(moveSpeed, 0);
    }


    void Update()
    {
        if (Input.GetKeyDown(KeyCode.W))
        {
            rb.velocity = new Vector2(0, moveSpeed);

        }

        if (Input.GetKeyDown(KeyCode.S))
        {
            rb.velocity = new Vector2(0, -moveSpeed);

        }

        if (Input.GetKeyDown(KeyCode.D))
        {
            rb.velocity = new Vector2(moveSpeed, 0);

        }

        if (Input.GetKeyDown(KeyCode.A))
        {
            rb.velocity = new Vector2(-moveSpeed, 0);

        }

    }
}
```

Code for food and score:-



```
using System.Collections.Generic;
using Unity.VisualScripting;
using UnityEngine;
using TMPro;

public class food : MonoBehaviour
{
     public BoxCollider2D foodSpawn;
     public float score;
     public TextMeshProUGUI scoreText;
      private void Start()
      {
         RandomPose();
      }
       private void Update()
       {
        scoreText.text = ""+score;
       }
      private void RandomPose()
      {
        Bounds bounds = this.foodSpawn.bounds;
        float x = Random.Range(bounds.min.x, bounds.max.x);
        float y = Random.Range(bounds.min.y, bounds.max.y);

        this.transform.position = new
Vector3(Mathf.Round(x),Mathf.Round(y),0.0f);

      }
      private void OnTriggerEnter2D(Collider2D collision)
      {
        if(collision.gameObject.tag == "Player")
        {
            RandomPose();
            score += 1;
        }
      }
}
```

Result:



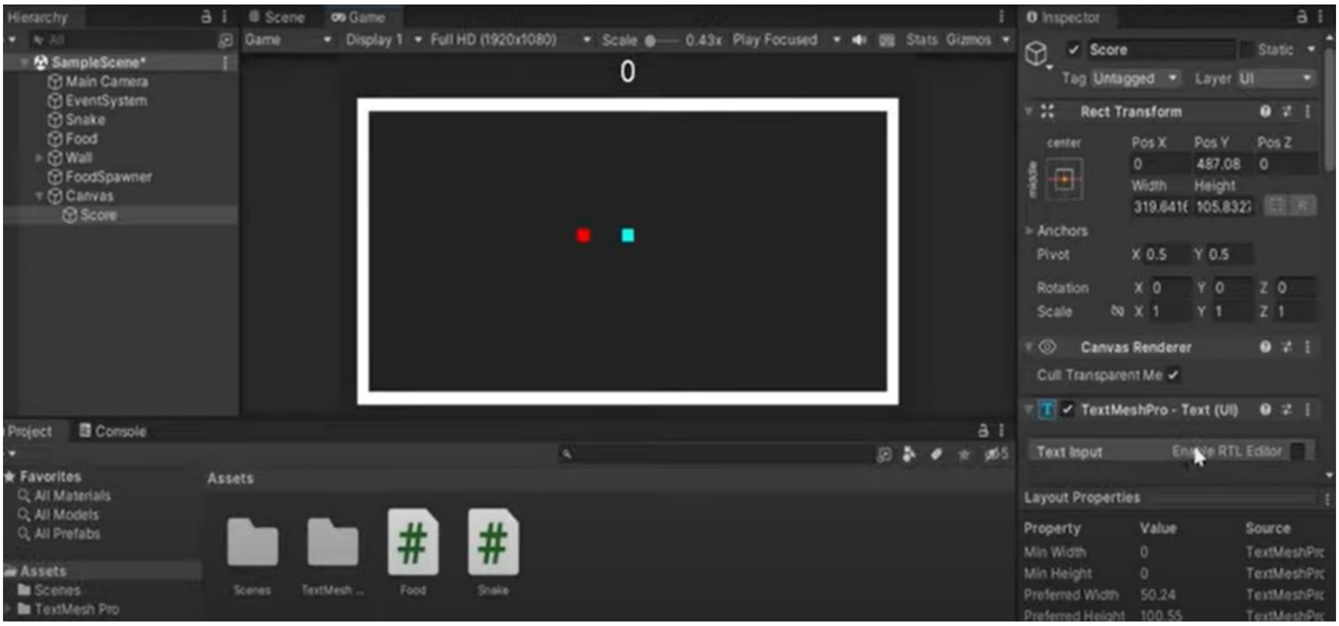
My project - SampleScene - Windows, Mac, Linux - Unity 2022.3.11f1* <DX11>

snake Static
Tag Player Layer Default

Transform
Position X 3.2535 Y -1.7728 Z 0
Rotation X 0 Y 0 Z 0.008
Scale X 0.4 Y 0.4 Z 1

Sprite Renderer

Rigidbody 2D
Body Type Dynamic
Material None (Physics Material 2D)
Simulated ✓
Use Auto Mass
Mass 1
Linear Drag 0
Angular Drag 0.05
Gravity Scale 0
Collision Detection Discrete
Sleeping Mode Start Awake
Interpolate None
Constraints
Layer Overrides
Include Layers Nothing
Exclude Layers Nothing
Info
Position X 3.2535 Y -1.7728



Score Static
Tag Untagged Layer UI

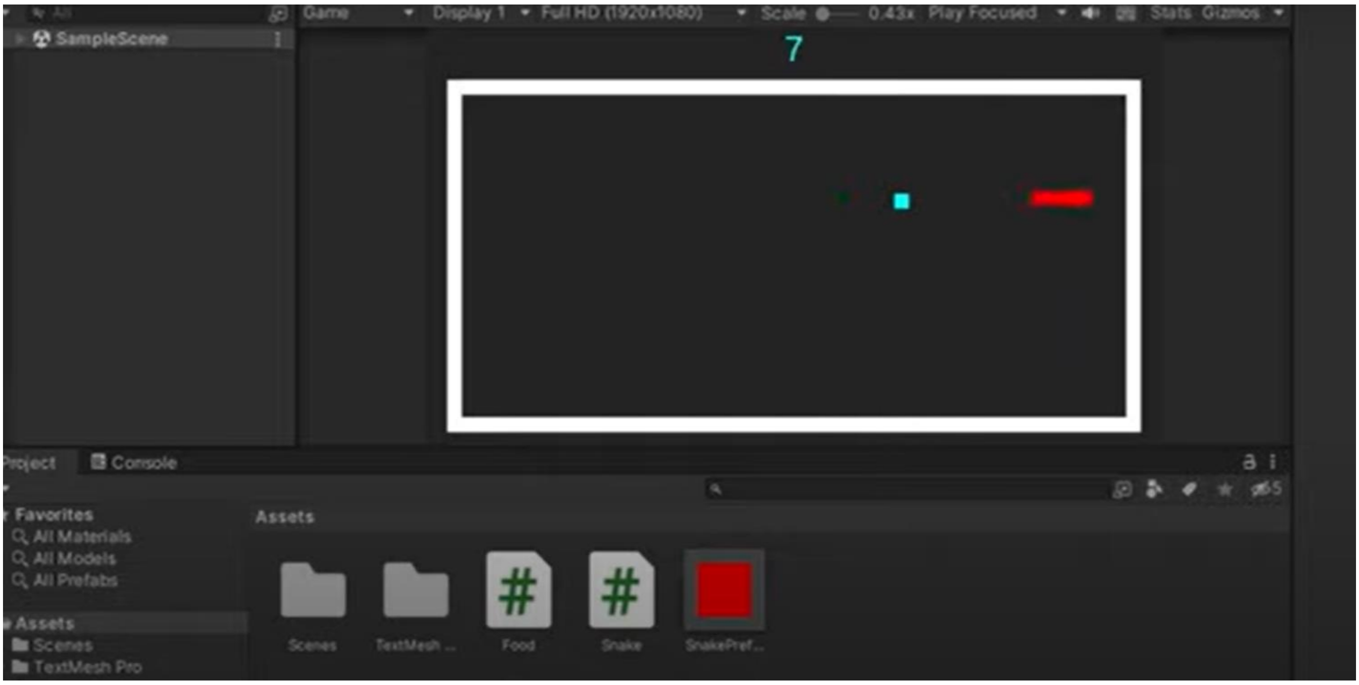
Rect Transform
center middle
Pos X 0 Pos Y 487.08 Pos Z 0
Width 319.641(Height 105.832
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Canvas Renderer
Cull Transparent Me ✓

TextMeshPro - Text (UI)
Text Input Enable RTL Editor

Layout Properties

Property	Value	Source
Min Width	0	TextMeshPro
Min Height	0	TextMeshPro
Preferred Width	50.24	TextMeshPro
Preferred Height	100.55	TextMeshPro



Conclusion:

Creating a Snake game in Unity has been a valuable learning experience. It showcases the flexibility and capabilities of the Unity game engine for developing 2D games. The project involved multiple development stages, including game design, coding, and optimization. It also required problem-solving skills for implementing core game mechanics like snake movement and collision detection.This project showcases the power of Unity in creating classic games for contemporary audiences while retaining the essence of the original gameplay.

References:

[1] Unity, "Unity User Manual 2021.3 (LTS)," Unity, [Online]. Available: https://docs.unity3d.com/Manual/index.html. [Accessed 21 May 2022].

[2] C. W. C. a. C. G. Marvin T. Chan, "Development of a Car Racing Simulator Game
Using Artificial Intelligence." in International Journal of Computer Games Technology,
2015.

[3] https://youtu.be/Iz22-o7l6bc?si=HV1XsA7_chJpN-kn