Report On

Ballon Fight 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
Bhavik Maru   (Roll No. 31)
Sanskruti Nijai   (Roll No. 39)
Prathamesh Shinde (Roll No. 56)

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 "Ballon Fight Game" is a bonafide work of "Bhavik Maru (Roll No. 31), Sanskruti Nijai (Roll No. 39), Prathamesh Shinde (Roll No. 56) 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 Ballon Fight Game project, a product of the versatile Unity Game Engine and C# programming language, presents an immersive gaming journey set within a captivating 3D arcade environment, accessible to players on both PC and Android devices. Through a comprehensive array of assets such as meticulously crafted C# scripts, prefabs, vibrant sprite images, and dynamic animations, the game boasts stunning graphics and seamless visual transitions, elevating the overall gameplay experience. With its straightforward and user-friendly controls, Ballon Fight offers players of all levels an enticing and intuitive gaming experience, making it an appealing choice for those seeking a seamless blend of entertainment and accessibility in the world of mobile gaming.

Problem Statement:

Despite the abundance of mobile gaming options, there exists a gap for a visually appealing and user-friendly 3D arcade game that seamlessly caters to both PC and Android users. The current market lacks immersive gaming experiences with intuitive controls that can captivate players across various skill levels, prompting the development of the Ballon Fight Game project to address this void and provide a compelling and accessible gaming solution for the modern audience.

Block diagram:
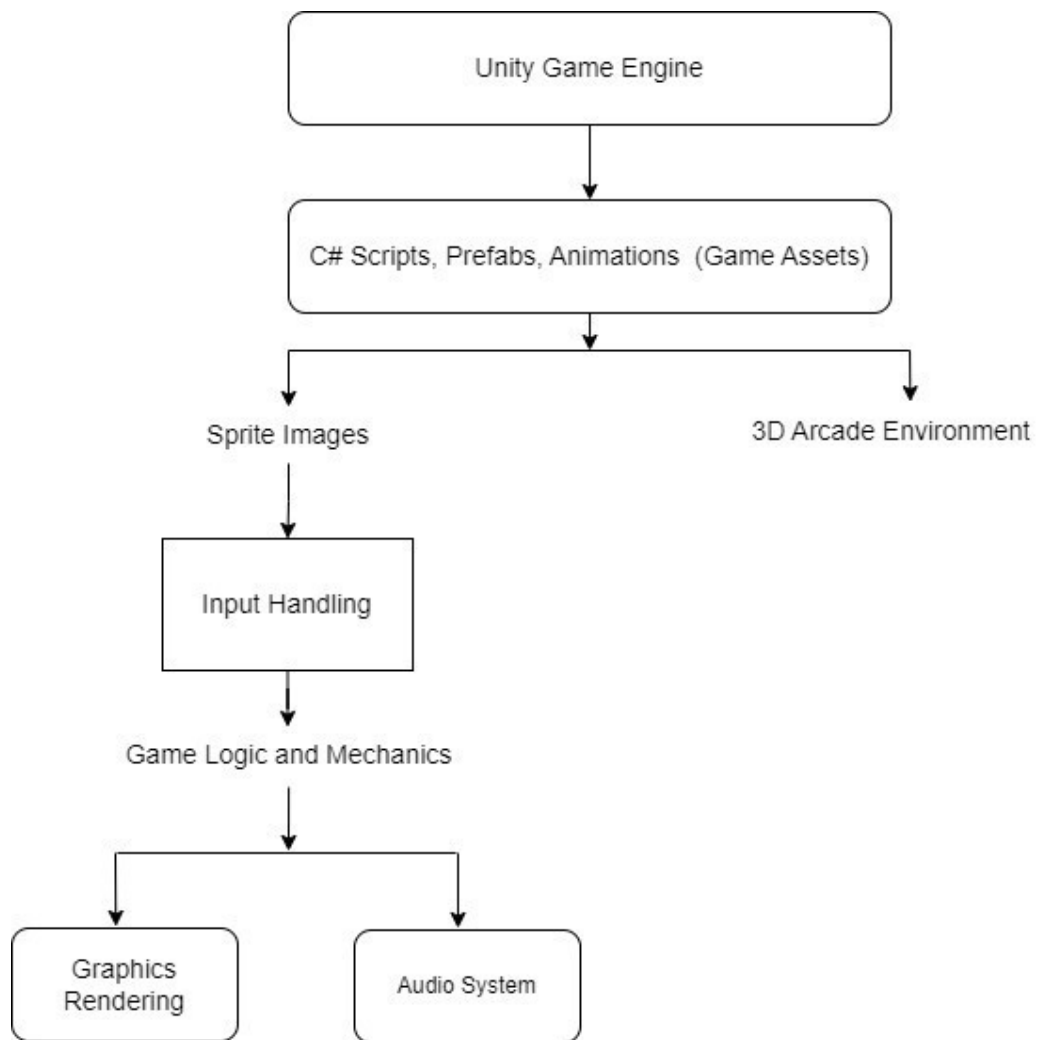
Description:

This diagram illustrates the interconnected elements involved in creating the Ballon Fight Game within the Unity environment, encompassing the Unity Game Engine, various assets, input handling, game logic, graphics rendering, and audio components. The integration of C# scripts and animations contributes to the game's mechanics and visual appeal, while the input handling ensures seamless interaction between the player and the game environment. The graphics rendering and audio system enhance the immersive experience, ultimately delivering a captivating and engaging gaming journey for players on both PC and Android platforms.

Module Description:

These modules work collaboratively to provide a cohesive and engaging gaming experience in. Each module has a specific role and is essential for the game's functionality and user enjoyment.

1. Player Input Module:
- Description: This module is responsible for handling user input. It allows players to control the direction of the rolling ball in the game. Players can interact with the game by tapping the screen or clicking the mouse. When an input event occurs, this module triggers the necessary actions to change the ball's direction.
- Functionality: Captures input events (e.g., taps or clicks) and translates them into actions to alter the ball's movement direction.
2. Game Logic Module:
- Description: The game logic module serves as the core of the Ballon Fight game. It manages various aspects of gameplay, including ball movement, obstacle generation, collision detection, scoring, and handling game over conditions.
- Functionality:
- Controls the rolling ball's forward movement.
- Generates obstacles at specific intervals and manages their behavior.
- Detects collisions between the player and obstacles.
- Updates the player's score based on successful actions.
- Monitors game over conditions, such as falling off the path or colliding with obstacles.
3. User Interface Module:
- Description: This module is responsible for creating and displaying the user interface elements of the game. It ensures that players can see their score and provides feedback when the game ends.
- Functionality:
- Displays the player's current score during gameplay.
- Presents the game over screen when the player fails.
- Offers options for the player to restart the game.

4. Game Assets Module:

- Description: The game assets module manages all the visual and audio elements that contribute to the overall look and feel of the game. It includes 3D models, textures, and sound effects.
- Functionality:
- Handles the loading and management of 3D models for the ball and obstacles.
- Manages 2D textures and graphical assets for the game's visual components.
- Integrates audio assets to enhance the player's auditory experience, including background music and sound effects.

Software & Hardware Requirements:

Software:

- Unity 20XX.XX: Used as the game development platform.
- C#: The primary programming language for scripting game logic.
- Blender: Utilized for creating 3D models and assets.
- Adobe Photoshop: Used for creating 2D textures and graphical assets.

Hardware:

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

Code:

Enemy
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Enemy : MonoBehaviour
{
    public float speed;
    public float period;
    bool ballon = true;
    Transform player;
    Animator anim;
    Rigidbody2D rb;
    SpriteRenderer sr;
    BoxCollider2D col;
    IEnumerator Start()
    {
        anim = GetComponent<Animator>();
        rb = GetComponent<Rigidbody2D>();
        sr = GetComponent<SpriteRenderer>();
        col = GetComponent<BoxCollider2D>();
        player = GameObject.FindGameObjectWithTag("Player").transform;
        anim.SetBool("Ballon", ballon);
        while (true)
        {
            if (ballon)
            {
                if (Random.value > 0.5f)
                {
                    tentativeVelocity = Random.onUnitSphere;
                    tentativeVelocity.y = tentativeVelocity.y < 0 ? 0 : tentativeVelocity.y;
                }
                else
                {
                    tentativeVelocity = player.position - transform.position;
                }
                tentativeVelocity.Normalize();

                sr.flipX = tentativeVelocity.x > 0;
                rb.AddForce(tentativeVelocity * speed, ForceMode2D.Impulse);
            }
            yield return new WaitForSeconds(period);
        }
    }
    Vector2 tentativeVelocity;
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Water"))
        {
            Destroy(gameObject);
        }
    }
    private void OnCollisionEnter2D(Collision2D collision)
```

```csharp
    {
        if (collision.collider.CompareTag("Player"))
        {
            if (collision.collider.transform.position.y > transform.position.y)
            {
                ballon = false;
                anim.SetBool("Ballon", ballon);
                rb.gravityScale = 1;
                return;
            }

            if (!ballon)
            {
                col.enabled = false;
                Global.singleton.AddScore();
                Destroy(gameObject, 3);
            }
        }
    }
}
```

Global.cs
```csharp
 using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
public class Global : MonoBehaviour {
    public static Global singleton;
    public Text scoreText;
    public GameObject gameOverGameObject;
    int score;
    private void Start()
    {
        if (singleton != null)
        {
            Destroy(gameObject);
            return;
        }
        singleton = this;
        scoreText.text = "Score: " + score.ToString();
    }
    public void AddScore()
    {
        score++;
        scoreText.text = "Score: " + score.ToString();
    }
    public void GameOver()
    {
        StartCoroutine(IEGameOver());
    }
    IEnumerator IEGameOver()
```

```csharp
        {
            yield return new WaitForSeconds(3);
            gameOverGameObject.SetActive(true);
            yield return new WaitForSeconds(3);
            SceneManager.LoadScene(0);
        }
}
```

Player.cs:
```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Player : MonoBehaviour
{
    public Transform groundPosition;
    public Vector2 groundSize;
    public float speed;
    public float jumpSpeed;
    public int ballon = 3;
    Animator anim;
    Rigidbody2D rb;
    SpriteRenderer sr;
    BoxCollider2D col;
    bool CheckGround
    {
        get
        {
            var ground = Physics2D.BoxCast(groundPosition.position, groundSize, 0,
Vector2.zero);

            if (ground.collider == null)
                return false;

            return true;
        }
    }
    private void Start()
    {
        anim = GetComponent<Animator>();
        rb = GetComponent<Rigidbody2D>();
        sr = GetComponent<SpriteRenderer>();
        col = GetComponent<BoxCollider2D>();

        anim.SetFloat("Ballons", ballon);
        rb.freezeRotation = true;
    }
    float horizontal = 0;
    bool jumpRequest;
    private void Update()
    {
        horizontal = Input.GetAxis("Horizontal");
        var g = CheckGround;

        if (Input.GetKeyDown(KeyCode.Space))
```

```csharp
        {
            jumpRequest = true;
        }
        if (horizontal == 0)
        {
            if(g)
                anim.speed = 0;
        }
        else
        {
            sr.flipX = horizontal > 0;
            anim.speed = 1;
        }
        anim.SetBool("Ground", g);
    }
    private void FixedUpdate()
    {
        if (jumpRequest)
        {
            jumpRequest = false;
            rb.AddForce(Vector2.up * jumpSpeed, ForceMode2D.Impulse);
        }

        rb.velocity = new Vector3(horizontal * speed, rb.velocity.y);
    }
    private void OnDrawGizmosSelected()
    {
        Gizmos.DrawWireCube(groundPosition.position, groundSize);
    }
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Water"))
        {
            Die();
        }
    }
    private void OnCollisionEnter2D(Collision2D collision)
    {
        if (collision.collider.CompareTag("Enemy"))
        {
            if (collision.collider.transform.position.y > transform.position.y)
            {
                ballon--;
                anim.SetFloat("Ballons", ballon);

                if (ballon <= 0)
                {
                    Die();
                }
            }
        }
    }
    void Die()
    {
        col.enabled = false;
```
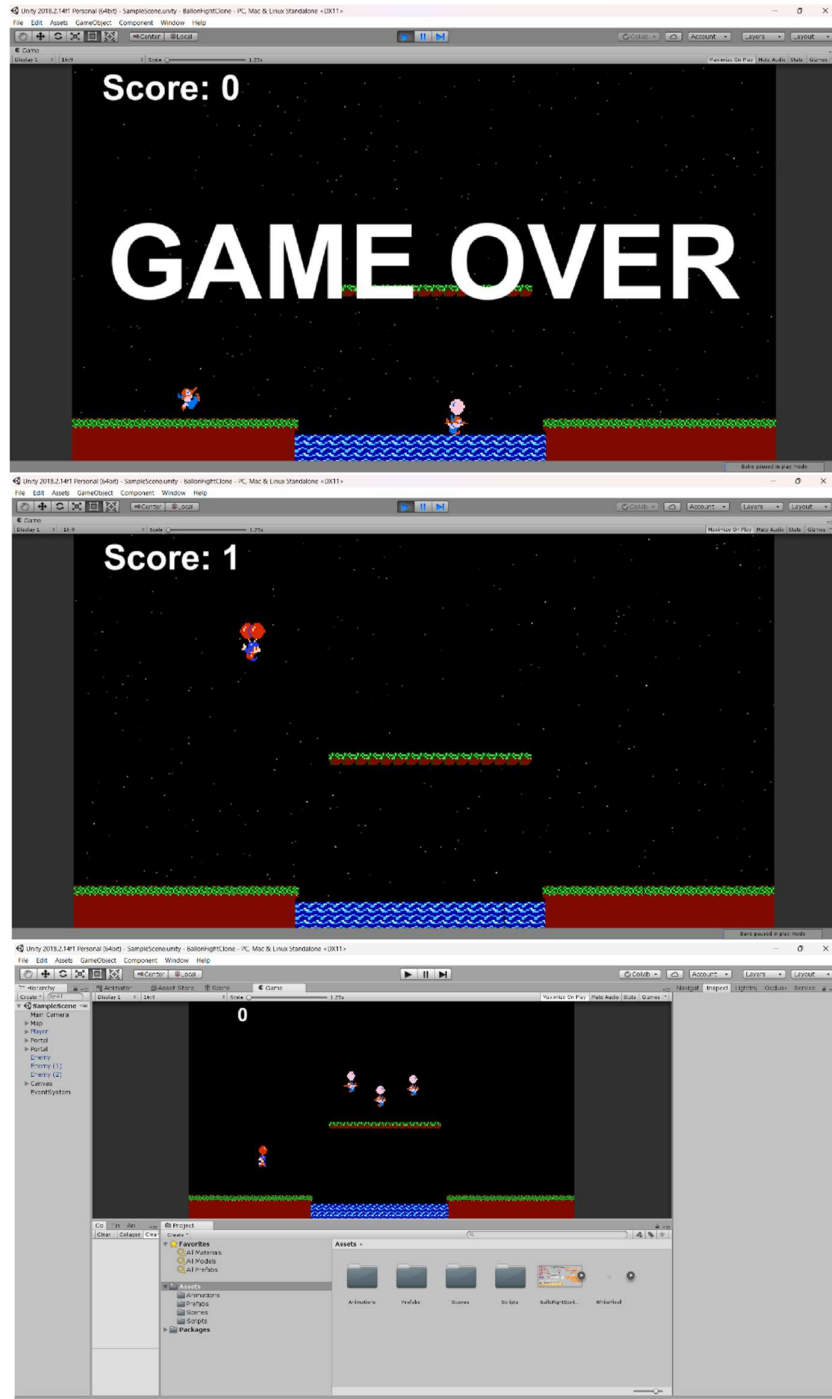
```
        rb.velocity = Vector2.zero;
        rb.gravityScale = 1;
        Global.singleton.GameOver();
        this.enabled = false;
    }
}
```

Result:

Conclusion:

In conclusion, the Ballon Fight Game project represents a successful fusion of the Unity Game Engine's flexibility and the capabilities of the C# programming language, delivering an engaging 3D arcade experience for PC and Android players. With meticulous attention to detail in its assets, including scripts, prefabs, animations, and sprite images, the game achieves a visually captivating and immersive environment. Its intuitive controls and seamless gameplay transitions further solidify its appeal to a broad audience, setting a standard for accessible and enjoyable gaming in the mobile gaming sphere.

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] Learn Everything. "Learn Unity 3D for Absolute Beginners," Udemy, [Online]. Available: https://www.udemy.com/course/learnunity3d/. [Accessed 21 May 2022].

[4] Brackeys, "Unity Beginner Tutorials," Youtube, [Online]. Available: https://www.youtube.com/watch?v=j48LtUkZRjU&list=PLPV2Kylb3jR5QFsefuO2R IA gWEz6EvVi6&index=1. [Accessed 21 May 2022].

[5] Unity, "Unity Asset Store," Unity, https://assetstore.unity.com/. [Accessed 21 May 2022]. [Online]. Available: https://assetstore.unity.com/. [Accessed 21 May 2022].