

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Formerly SRM University, Under section 3 of UGC Act, 1956)

S.R.M. NAGAR, KATTANKULATHUR –603 203

COLLEGE OF ENGINEERING & TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF DATASCIENCE AND BUSINESS SYSTEMS



GAME DESIGN REPORT

Course Code: 18CSE337J

Course Name: Game Design, Prototyping and Development

Name :

Reg.no :

Class : B.Tech– Computer Science and Engineering

Year /Semester: III / V

Specialization: Gaming Technology

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203



BONAFIDE CERTIFICATE

Certified that this is a bonafide record of practical work done by
_____, Reg. No. _____
_____ pursuing Third year B.tech, Computer Science and
Engineering with Gaming Technology Specialization in the Subject **Game
Design, Prototyping and Development (18CSE337J)** during the End
Semester University Examinations , November 2022.

Faculty in-charge

Dr.P.Saravanan

Head of the Department

Dr.M.Lakshmi

Dept. of DSBS

Submitted for the University Examination held on _____

Internal Examiner-1

Internal Examiner-2

GAME IDEAS

1. Paint Brawls
2. Pixel Rush.

GAME Design Document

Game 1 - Paint Brawls

Tag: Painting Game, Multiplayer, Art

Team Detail:

1. Jalaj Shah – Developer
2. Anushree Bajaj – Designer
3. Aryan Maurya - VFX leads
4. Chirag Khurana – Side Developer

Concept

Paint Brawls is a single and multiplayer game which can be played online/offline in which the player has to paint the white screen and the player which has the most area covered with his color wins the game.

Target Audience

Underage Kids are the major targeted audience as this is a basic fun game which can be played and enjoyed by them. Also, a major Advantage of these game is it does not involve any sought of violence which can be used as a marketing point.

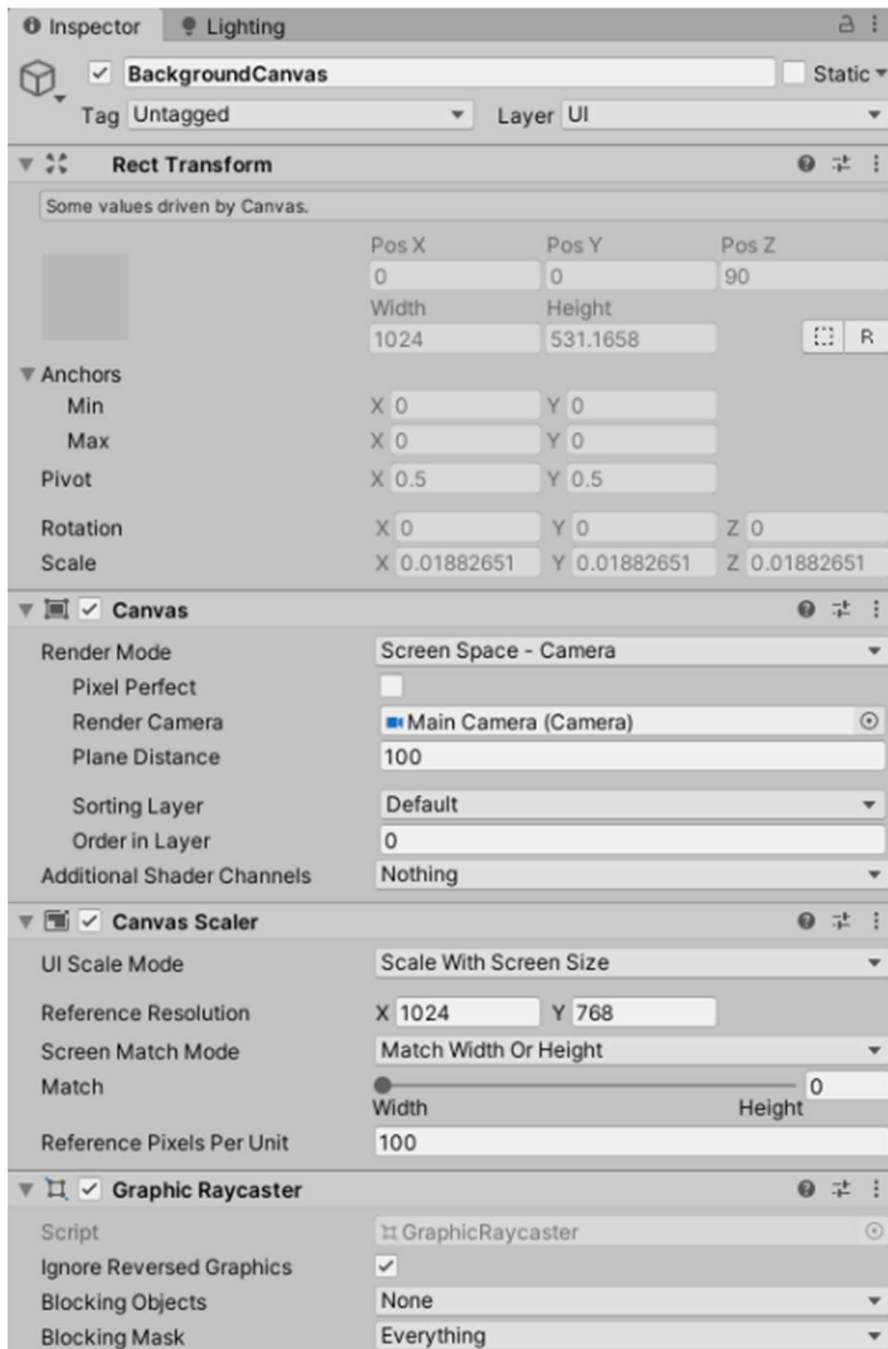
Scene join Code

```
// Load an assetbundle which contains Scenes.  
// When the user clicks a button the first Scene in the assetbundle is  
// loaded and replaces the current Scene.  
  
using UnityEngine;  
using UnityEngine.SceneManagement;  
  
public class LoadScene : MonoBehaviour  
{  
    private AssetBundle myLoadedAssetBundle;  
    private string[] scenePaths;  
  
    // Use this for initialization  
    void Start()  
    {  
        myLoadedAssetBundle = AssetBundle.LoadFromFile("Assets/AssetBundles/scenes");  
        scenePaths = myLoadedAssetBundle.GetAllScenePaths();  
    }  
  
    void OnGUI()  
    {  
        if (GUI.Button(new Rect(10, 10, 100, 30), "Change Scene"))  
        {  
            Debug.Log("Scene2 loading: " + scenePaths[0]);  
            SceneManager.LoadScene(scenePaths[0], LoadSceneMode.Single);  
        }  
    }  
}
```

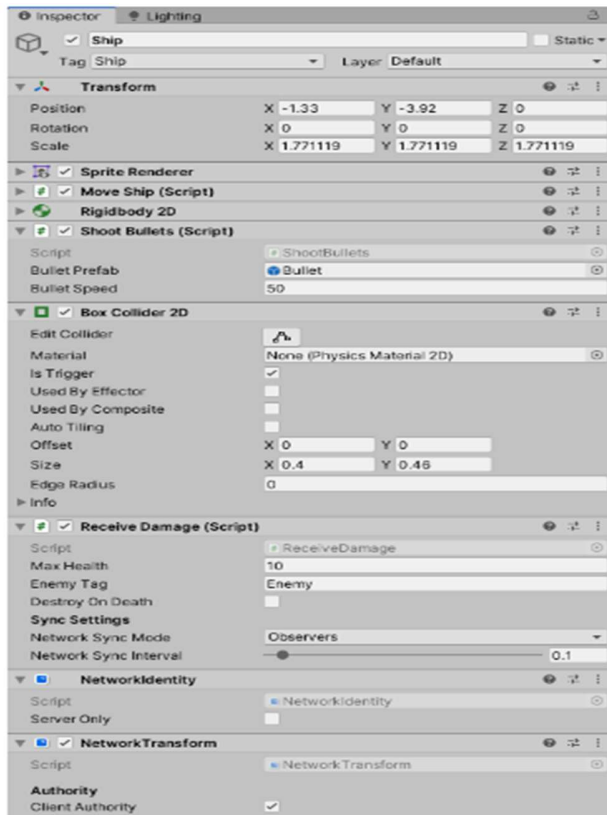
Exit code

```
void QuitGame () {  
    Application.Quit ();  
    Debug.Log("Game is exiting");  
    //Just to make sure its working  
}
```

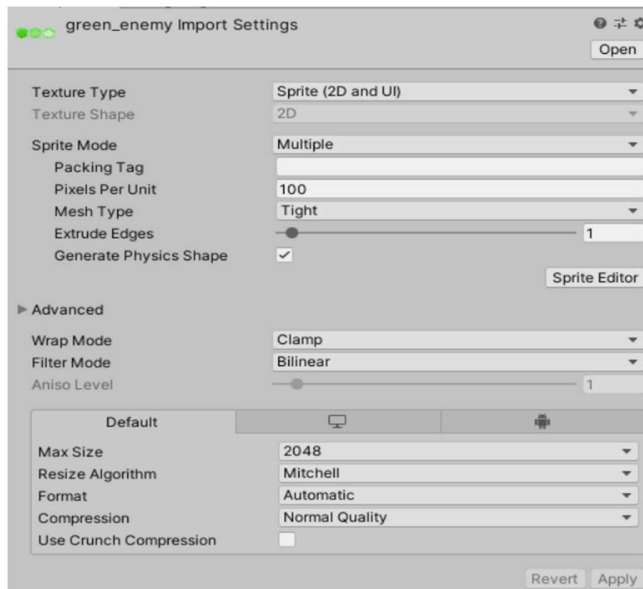
Set Background



Brush Movement



Multiplayer Score Card Setting



Pictorial View



Game 2 - Pixel Rush

Tag: Running Game, Survival

Team Detail:

1. Jalaj Shah – Developer
2. Anushree Bajaj – Designer
3. Aryan Maurya - VFX leads
4. Chirag Khurana – Side Developer

Concept

This is the pixelate runner game in which there are several levels in which a player has to clear each level to go ahead in the game it is a single player game which a gamer can enjoy any time

Target Audience

This Game also target under age kids and also doesn't involve violence. But this type of game is generally interesting and are popular in all age group.

Scene Join Code

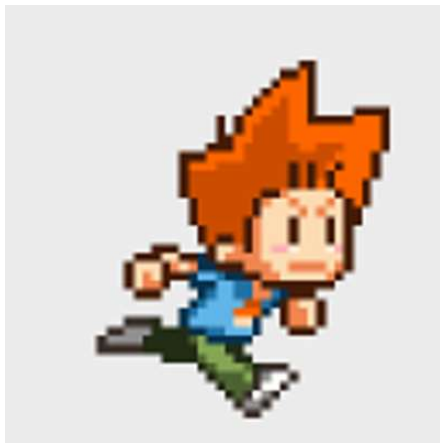
```
// Load an assetbundle which contains Scenes.  
// When the user clicks a button the first Scene in the assetbundle is  
// loaded and replaces the current Scene.  
  
using UnityEngine;  
using UnityEngine.SceneManagement;  
  
public class LoadScene : MonoBehaviour  
{  
    private AssetBundle myLoadedAssetBundle;  
    private string[] scenePaths;  
  
    // Use this for initialization  
    void Start()  
    {  
        myLoadedAssetBundle = AssetBundle.LoadFromFile("Assets/AssetBundles/scenes");  
        scenePaths = myLoadedAssetBundle.GetAllScenePaths();  
    }  
  
    void OnGUI()  
    {  
        if (GUI.Button(new Rect(10, 10, 100, 30), "Change Scene"))  
        {  
            Debug.Log("Scene2 loading: " + scenePaths[0]);  
            SceneManager.LoadScene(scenePaths[0], LoadSceneMode.Single);  
        }  
    }  
}
```


Exit Code

```
void QuitGame () {  
    Application.Quit ();  
    Debug.Log("Game is exiting");  
    //Just to make sure its working  
}
```

Sample Character

1.



2.



Character Movement

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

public class Example : MonoBehaviour
{
    private CharacterController controller;
    private Vector3 playerVelocity;
    private bool groundedPlayer;
    private float playerSpeed = 2.0f;
    private float jumpHeight = 1.0f;
    private float gravityValue = -9.81f;

    private void Start()
    {
        controller = gameObject.AddComponent<CharacterController>();
    }

    void Update()
    {
        groundedPlayer = controller.isGrounded;
        if (groundedPlayer && playerVelocity.y < 0)
        {
            playerVelocity.y = 0f;
        }

        Vector3 move = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical"));
        controller.Move(move * Time.deltaTime * playerSpeed);

        if (move != Vector3.zero)
        {
            gameObject.transform.forward = move;
        }

        // Changes the height position of the player..
        if (Input.GetButtonDown("Jump") && groundedPlayer)
        {
            playerVelocity.y += Mathf.Sqrt(jumpHeight * -3.0f * gravityValue);
        }

        playerVelocity.y += gravityValue * Time.deltaTime;
        controller.Move(playerVelocity * Time.deltaTime);
    }
}
```

Jump Control

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerController : MonoBehaviour
6  {
7      private Rigidbody2D rigidBody2D;
8      private CircleCollider2D circleCollider2D;
9      [SerializeField] private LayerMask groundLayer;
10     [Range(0, 10f)] [SerializeField] private float speed = 0f;
11
12     float horizontal = 0f;
13     float lastJumpY = 0f;
14     private bool isFacingRight = true;
15     bool jump = false, jumpHeld = false;
16
17     [Range(0, 5f)] [SerializeField] private float fallLongMult = 0.85f;
18     [Range(0, 5f)] [SerializeField] private float fallShortMult = 1.55f;
19
20     void Start()
21     {
22         ...
23     }
24
25     void Update()
26     {
27         ...
28     }
29
30     void FixedUpdate()
31     {
32         ...
33     }
34
35     private void flipSprite()
36     {
37         ...
38     }
39
40     private bool isOnGround()
41     {
42         ...
43     }
44 }
45
```

Pictorial View

