SIMPLE BOX GAME

Made with UNITY Game Engine

Project Report

Diploma in
Computer Science and Technology

Submitted by
Bodhisatta Bhattacharjee
Mehedi Hassan
Swastika Sarkar
Susmita Sarkar
Uttara Ganguly

Under the guidance of
Ria Bandyopadhyay

2022

JIS School of Polytechnic
Kalyani – 741235
West Bengal, India

ACKNOWLEDGEMENT

Apart from our own efforts, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to those people who have been instrumental in the successful completion of this project.

We would like to show our greatest appreciation to Asbjørn Thirslund from his ONLINE tutorials in his Youtube channel – “Brackeys”.
Channel link -
https://www.youtube.com/channel/UCYbK_tjZ2OrIZFBvU6CCMiA

We are very thankful to Prof. RIA BANDYOPADHYAY, HOD of CST department for showing us the way during the project briefing.

Project Member Names (with Registration Number)–

Bodhisatta Bhattacharjee	D192012602
Mehedi Hassan	D192012608
Susmita Sarkar	D192012619
Swastika Sarkar	D192012620
Uttara Ganguly	D192012623

CERTIFICATE

This is to certify that this project “SIMPLE BOX GAME” in the year 2022 is bonafied record of work done at “JIS SCHOOL OF POLYTECHNIC” by:

BODHISATTA BHATTACHARJEE

MEHEDI HASSAN

SWASTIKA SARKAR

SUSMITA SARKAR

UTTARA GANGULY

Under the guidance and supervision of Assistant professor of Computer Science & Technology of JIS SCHOOL OF POLYTECHNIC.

Signature of the Project Mentor

RIA BANDYOPADHYAY

Signature of H.O.D

RIA BANDYOPADHYAY

INDEX

INTRODUCTION

For our **SIXTH SEMESTER PROJECT,** we have made an Unity Puzzle Game to be played on Windows, with very easy controls. We have used some simple textures created by myself, and downloaded some audio from the internet for free. With some knowledge of “UNITY Game Engine” and a scripting language “C# (C SHARP)”, we have made this game possible.

FEASIBILITY STUDY

Having an high-end computer is enough for using the "*Free Version of UNITY*" which can be used to make a simple Unity game. This game with all the prefabs, scripts, sound and user interfaces can be used to make more games with just a few changes. For now, the game that is made is for my own practice purpose for the "*Final Year Project of College*", but soon we will be using this basic game template to make better versions out of the game that can be played on Android and uploaded on Google Play Store.

User Requirements

Unity Editor version 2021 LTS

This section lists the minimum requirements to run the Unity Editor. Actual performance and rendering quality may vary depending on the complexity of your project.

	Operating system version	**CPU**	**Graphics API**	**Additional requirements**
Windows	Windows 7 (SP1+), Windows 10 and Windows 11, 64-bit versions only.	X64 architecture with SSE2 instruction set support	DX10, DX11, and DX12-capable GPUs	Hardware vendor officially supported drivers
macOS	High Sierra 10.13+ (Intel editor) Big Sur 11.0 (Apple silicon Editor)	X64 architecture with SSE2 instruction set support (Intel processors) Apple M1 or above (Apple silicon-based processors)	Metal-capable Intel and AMD GPUs	Apple officially supported drivers (Intel processor) Rosetta 2 is required for Apple silicon devices running on either Apple silicon or Intel versions of the Unity Editor.
Linux	Ubuntu 20.04, Ubuntu 18.04, and CentOS 7	X64 architecture with SSE2 instruction set support	OpenGL 3.2+ or Vulkan-capable, Nvidia and AMD GPUs.	Gnome desktop environment running on top of X11 windowing system, Nvidia official proprietary graphics driver or AMD Mesa graphics driver. Other configuration and user environment as provided stock with the supported distribution (Kernel, Compositor, etc.)

For all operating systems, the Unity Editor is supported on workstations or laptop form factors, running without emulation, container or compatibility layer.

Game Concept

As the project name suggests, the game concept is “Simple” indeed. The player (user) controls an “Yellow” square (cube) shaped player on top of which there is a spotlight which is highlighting the player as it moves along. The player has simple controls – “W”, “A”, “S”, “D” – for moving “forward”, “left”, “backward”, “right” respectively. There can be three kinds of objects that can be found along the way coloured green, dull red, dark red. If the player is colliding with the green coloured object, the player wins the level otherwise if the player hits the other two, the player loses that level. On losing the level the player starts again from the start of that level, and if the player wins then they go to the next level.

Target Platforms

Currently the game is only playable on Windows 7 SP1+, 8, 10, 11 on 64-bit systems only.

In the future, some touch movement controls will be implemented and ANDROID systems will also be a part of the target platforms.

Game Mechanics

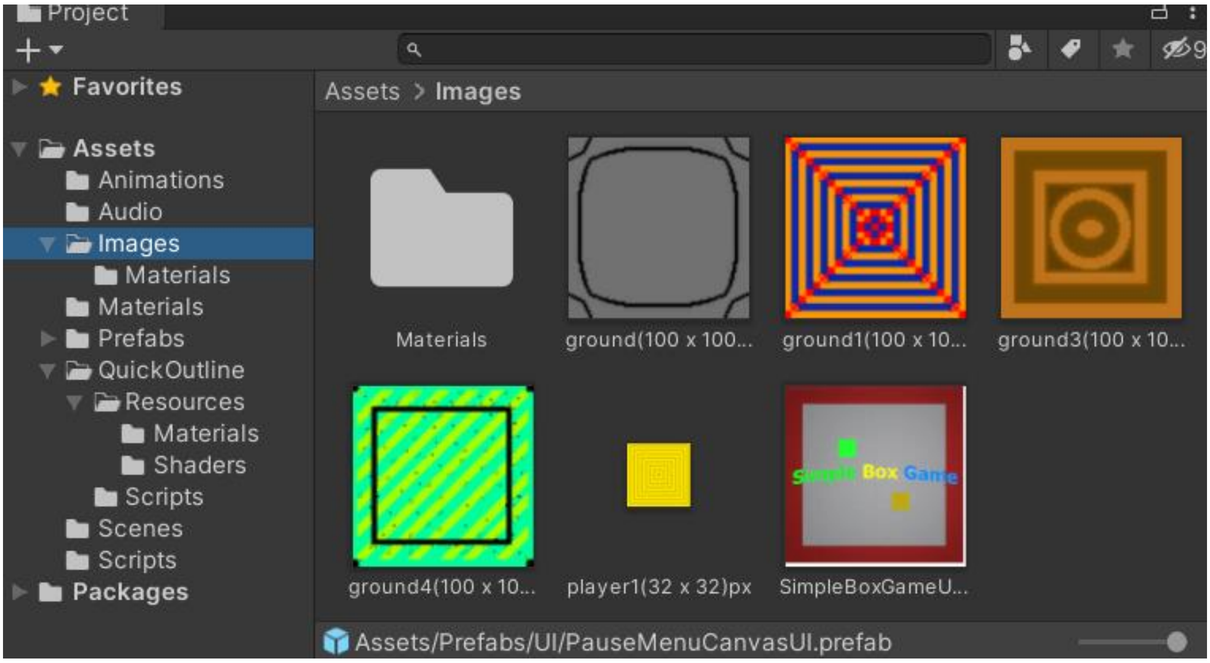
Movement	The player has simple controls – “W”, “A”, “S”, “D” – for moving “forward”, “left”, “backward”, “right” respectively.
Music	There is a background music that plays as soon as the player opens the game .exe file. And that music plays forever until the player quits the game. Even when the levels change the music does not stop nor does it start from the beginning, it keeps continuing.
Sound Effects	Whenever the player collides with any object it produces two sounds depending on of the player had won or lost.
User Interfaces	As soon as the game is being launched, there is a simple menu screen with game logo, and some buttons to start game, quit game, select level, and see credits. On any level, if the player presses “ESC” on keyboard, an overlay animation shows with some options to resume, quit, and go back to menu. These are some basic UI that a game should have to make the game more playable and interactive.

Assets

Most of the assets used in the game were created by me and some were downloaded from the internet.

1. The player, walls, rotating walls, green winning object, user interfaces – were created inside of UNITY itself.

2. The background texture (ground tiles) were created by me using an online pixel art editor (free to use) - https://www.piskelapp.com/
And the logo was created using PHOTOPEA, an image editor for free - https://www.photopea.com/



3. Some of the sounds in game were created by me using the BFXR sound effects tool (https://www.bfxr.net/) and some were downloaded from the internet (the winning tone, and the female voice saying FAIL – both free to use)

Scripts

Many scripts were used to give behaviours and other logic to the game components. Shown below …

1. AudioManager.cs

```
using System;
using UnityEngine;
using UnityEngine.Audio;

public class AudioManager : MonoBehaviour
{
    public Sound[] sounds;

    public static AudioManager instance;

    private void Start()
    {
        Play("Theme");
    }

    private void Awake()
    {
        if (instance == null)
        {
            instance = this;
        }
        else
        {
            Destroy(gameObject);
            return;
        }

        DontDestroyOnLoad(gameObject);

        foreach (Sound s in sounds)
        {
            s.source = gameObject.AddComponent<AudioSource>();
            s.source.clip = s.clip;
            s.source.volume = s.volume;
            s.source.pitch = s.pitch;
            s.source.loop = s.loop;
        }
    }

    public void Play(string name)
    {
        Sound s = Array.Find(sounds, sound => sound.name == name);
        if (s == null)
        {
            Debug.LogWarning("Sound : " + name + " - not found !");
            return;
        }
        s.source.Play();
    }
}
```

2. ChangeTexture.cs

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

public class ChangeTexture : MonoBehaviour
{
    public Texture[] groundTextures;

    // Start is called before the first frame update
    void Start()
    {
        try
        {
            int x = UnityEngine.Random.Range(0, groundTextures.Length); // all grounds same texture on that scene
            GameObject[] grounds =  GameObject.FindGameObjectsWithTag("Ground");
            foreach (GameObject g in grounds)
            {
                //int x = UnityEngine.Random.Range(0, groundTextures.Length*10) % groundTextures.Length; // texture not same on all grounds
                g.GetComponent<Renderer>().material.mainTexture = groundTextures[x];
            }
        }
        catch (Exception e)
        {
            Debug.Log("No Textures found / Error in getting texture.\nExcemption : " + e);
        }
    }
}
```

3. Credits.cs

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

public class Credits : MonoBehaviour
{
    public void Quit()
    {
        Debug.Log("Game has Quit");
        Application.Quit();
    }

    public void MainMenu()
    {
        SceneManager.LoadScene("MainMenu");
    }

}
```

4. Follow.cs

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

public class Follow : MonoBehaviour
{
    public Transform target;

    public float smoothSpeed = 0.125f;
    public Vector3 offset ;

    private void Start()
    {
        try
        {
            Light light = GetComponent<Light>();
            light.range = 5.6f;
            light.intensity = 35;
        }
        catch (Exception e)
        {
            Debug.Log("The gameobject : " + transform.name + " has no 'component' of type <Light> attached to it.\nException : " + e);
        }
    }

    private void LateUpdate()
    {
        GameObject scenePlayer = GameObject.Find("Player");
        // transform.position = target.position + offset;
        transform.position = scenePlayer.transform.position + offset;
    }
}
```

5. GameManager.cs

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

public class GameManager : MonoBehaviour
{
    public Light myLight;
    public GameObject player;

    public float nextLevelDelay = 2f;
    public float levelRestartDelay = 2f;

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.L) && (PauseMenu.gameIsPaused == false))
        {
            myLight.enabled = !myLight.enabled;
        }
    }

    public void LevelComplete()
    {
        Invoke("NextLevel", nextLevelDelay);
    }

    public void GameOver()
    {
        Invoke("Restart", levelRestartDelay);
    }

    private void NextLevel()
    {
        if (GameObject.Find("WinUI") == true)
        {
            GameObject.Find("WinUI").SetActive(false);
        }
        else
        {
            Debug.Log("Cant find the GUI for WIN and LOSE");

        }

        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    private void Restart()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }

}
```

6. LevelSelectMenuScript.cs

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

public class LevelSelectMenuScript : MonoBehaviour
{
    public void Back()
    {
        SceneManager.LoadScene("MainMenu");
    }

    public void Level1Button()
    {
        SceneManager.LoadScene("Level1");
    }

    public void Level2Button()
    {
        SceneManager.LoadScene("Level2");
    }
    public void Level3Button()
    {
        SceneManager.LoadScene("Level3");
    }

    public void Level4Button()
    {
        SceneManager.LoadScene("Level4");
    }

    public void Level5Button()
    {
        SceneManager.LoadScene("Level5");
    }

    public void Level6Button()
    {
        SceneManager.LoadScene("Level6");
    }
    public void Level7Button()
    {
        SceneManager.LoadScene("Level7");
    }

    public void Level8Button()
    {
        SceneManager.LoadScene("Level8");
    }

    public void Level9Button()
    {
        SceneManager.LoadScene("Level9");
    }

    public void Level10Button()
    {
        SceneManager.LoadScene("Level10");
    }

    public void CreditsButton()
    {
        SceneManager.LoadScene("Credits");
    }
}
```

7. MainMenu.cs

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

public class MainMenu : MonoBehaviour
{
    public void LevelStart()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    public void LevelSelect()
    {
        SceneManager.LoadScene("LevelSelectMenu");
    }

    public void GameQuit()
    {
        Debug.Log("Quitting the game.");
        Application.Quit();
    }
}
```

8. Movement.cs

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

public class Movement : MonoBehaviour
{
    public float moveSpeed = 5f;

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey(KeyCode.RightArrow) || Input.GetKey(KeyCode.D))
        {
            transform.Translate(moveSpeed * Vector3.right * Time.deltaTime);
        }
        if (Input.GetKey(KeyCode.LeftArrow) || Input.GetKey(KeyCode.A))
        {
            transform.Translate(moveSpeed * Vector3.left * Time.deltaTime);
        }
        if (Input.GetKey(KeyCode.UpArrow) || Input.GetKey(KeyCode.W))
        {
            transform.Translate(moveSpeed * Vector3.forward * Time.deltaTime);
        }
        if (Input.GetKey(KeyCode.DownArrow) || Input.GetKey(KeyCode.S))
        {
            transform.Translate(moveSpeed * Vector3.back * Time.deltaTime);
        }

    }
}
```

9. PauseMenu.cs

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

public class PauseMenu : MonoBehaviour
{
    public static bool gameIsPaused = false;

    public GameObject pauseMenuUI;

    private void Update()
    {
        if (Input.GetKeyDown(KeyCode.Escape) || Input.GetKeyDown(KeyCode.P))
        {
            if (gameIsPaused)
            {
                Resume();
            }
            else
            {
                Pause();
            }
        }
    }

    public void Resume()
    {
        pauseMenuUI.SetActive(false);
        Time.timeScale = 1f;
        gameIsPaused = false;
    }

    void Pause()
    {
        pauseMenuUI.SetActive(true);
        Time.timeScale = 0f;
        gameIsPaused = true;
    }

    public void LoadMenu()
    {
        Time.timeScale = 1f;
        SceneManager.LoadScene("MainMenu");
        gameIsPaused = false;
    }

    public void QuitGame()
    {
        Debug.Log("Quitting the game");
        Application.Quit();
    }
}
```

10. Rotator.cs

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

public class Rotator : MonoBehaviour
{
    public bool rotateClockWise = true;
    public float turnSpeed = 100f;

    // Update is called once per frame
    void FixedUpdate()
    {
        if (rotateClockWise == true)
        {
            transform.Rotate(Vector3.up, turnSpeed * Time.deltaTime);
        }
        else transform.Rotate(Vector3.up, -turnSpeed * Time.deltaTime);
    }
}
```

11. Sound.cs

```
using UnityEngine;
using UnityEngine.Audio;

[System.Serializable]
public class Sound
{
    public string name;

    public AudioClip clip;

    [Range(0.0001f, 1f)]
    public float volume;
    [Range(0.1f, 3f)]
    public float pitch;

    public bool loop;

    [HideInInspector]
    public AudioSource source;
}
```

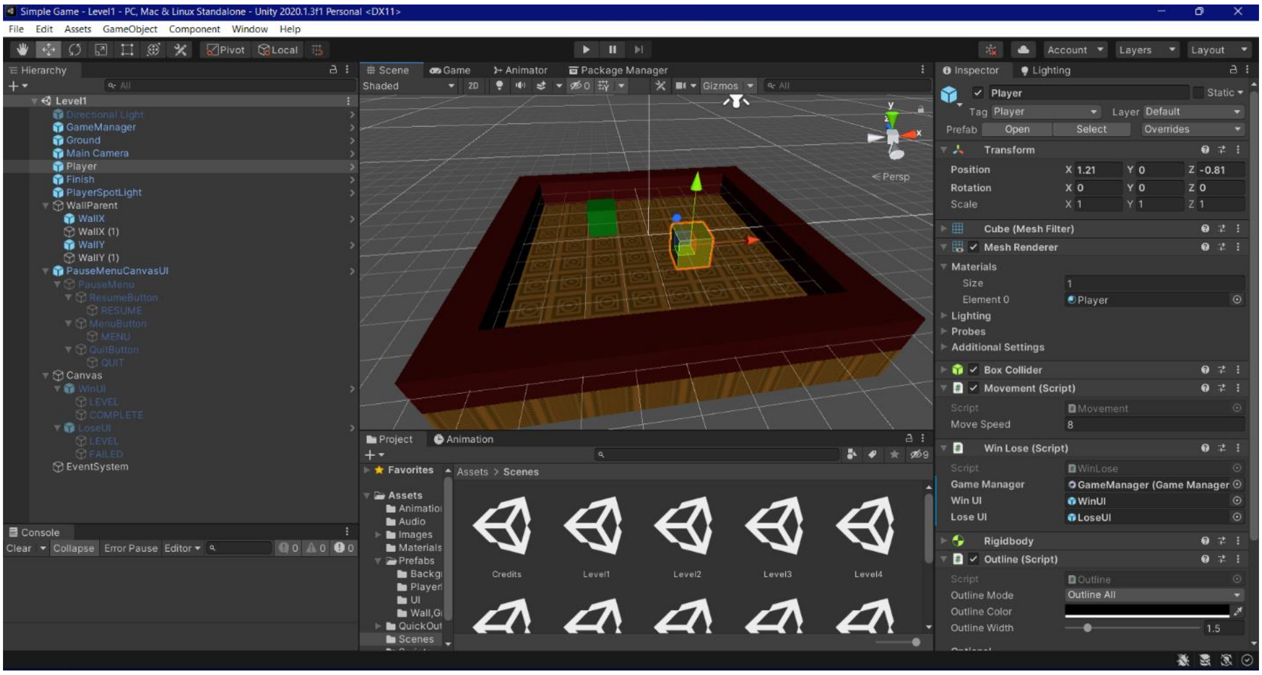
12. WinLose.cs

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

public class WinLose : MonoBehaviour
{
    public GameManager GameManager;
    public GameObject WinUI;
    public GameObject LoseUI;

    private GameObject a;

    private bool isWin = false;
    private bool isLose = false;
    private bool soundPlaying = false;

    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag("Finish") && (isLose != true) && (soundPlaying == false))
        {
            FindObjectOfType<AudioManager>().Play("LevelComplete");
            soundPlaying = true;
            WinUI.SetActive(true);
            isWin = true;
            GameManager.LevelComplete();
            // isWin = false;
        }
        else if (other.gameObject.CompareTag("Wall") && (isWin != true) && (soundPlaying == false))
        {
            FindObjectOfType<AudioManager>().Play("HitWall");
            FindObjectOfType<AudioManager>().Play("LevelFailed");
            soundPlaying = true;
            LoseUI.SetActive(true);
            isLose = true;
            GameManager.GameOver();
            // isLose = false;
        }
    }
}
```

Scenes

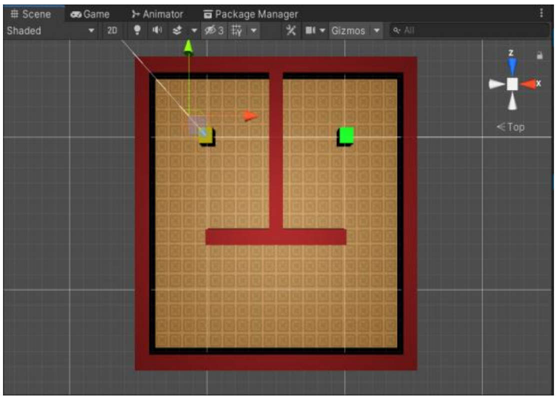
Till now we have added 10 scenes to my game. The screenshots of the top view of all the scenes are as follows

Scene 1

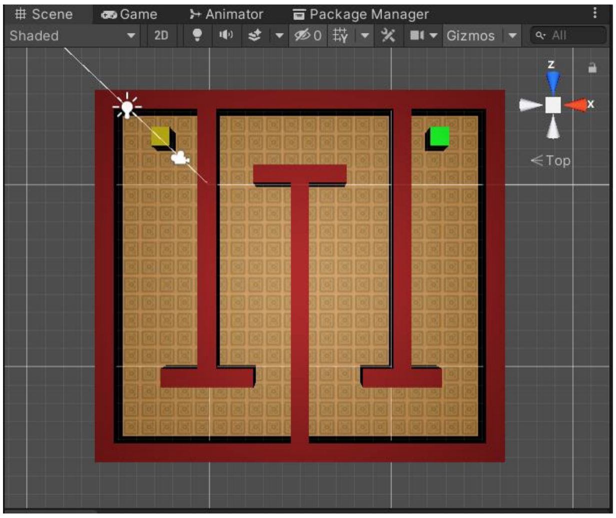


Player Static
Tag Player Layer Default
Prefab Open Select Overrides
Transform
Position X 1.21 Y 0 Z -0.81
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Cube (Mesh Filter)
Mesh Renderer
Materials
Size 1
Element 0 Player
Lighting
Probes
Additional Settings
Box Collider
Movement (Script)
Script Movement
Move Speed 8
Win Lose (Script)
Script WinLose
Game Manager GameManager (Game Manager
Win UI WinUI
Lose UI LoseUI
Rigidbody
Outline (Script)
Script Outline
Outline Mode Outline All
Outline Color
Outline Width 1.5

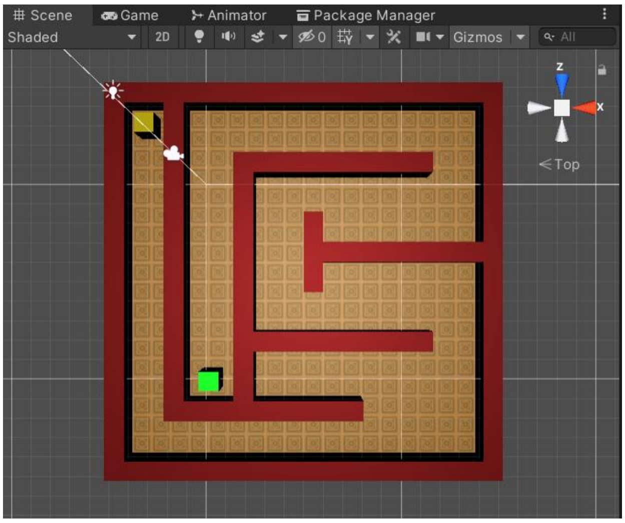
Scene 2



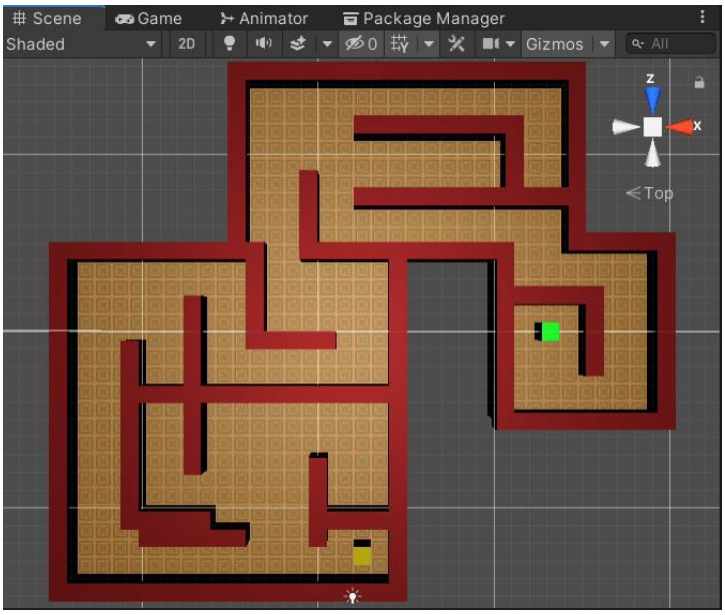
Scene 3



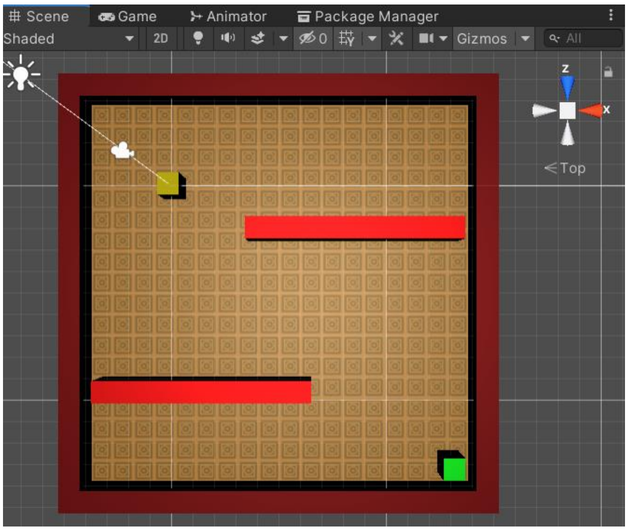
Scene 4



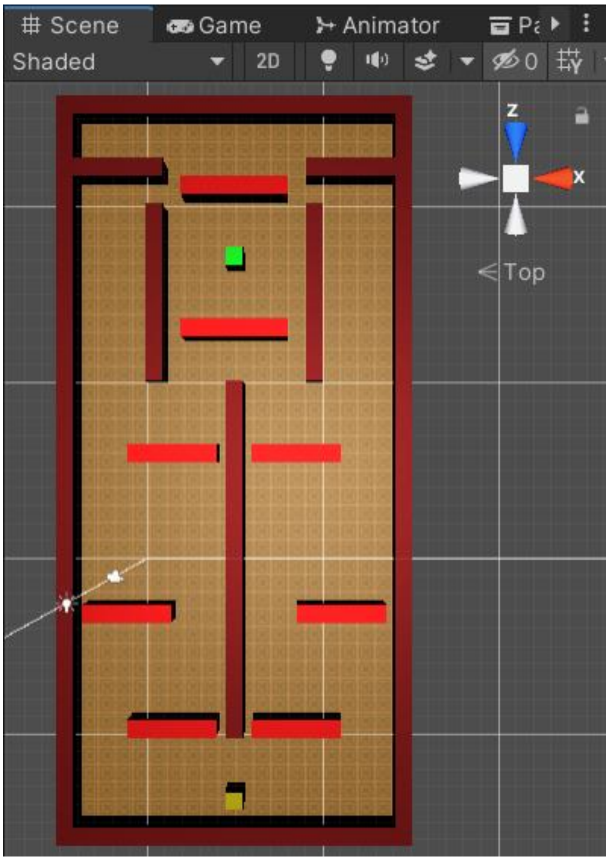
Scene 5



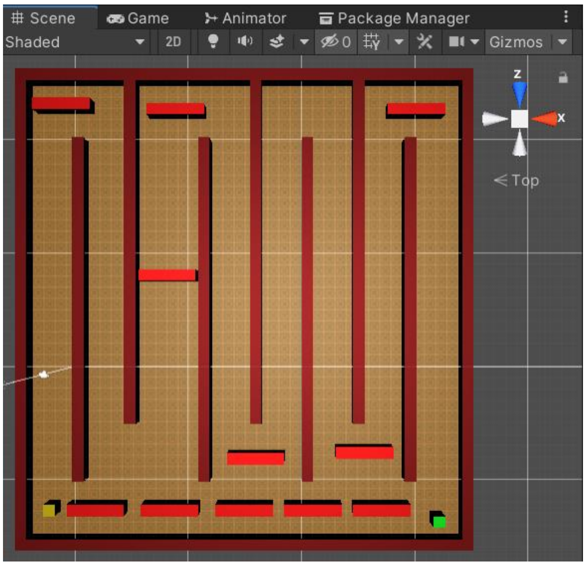
Scene 6



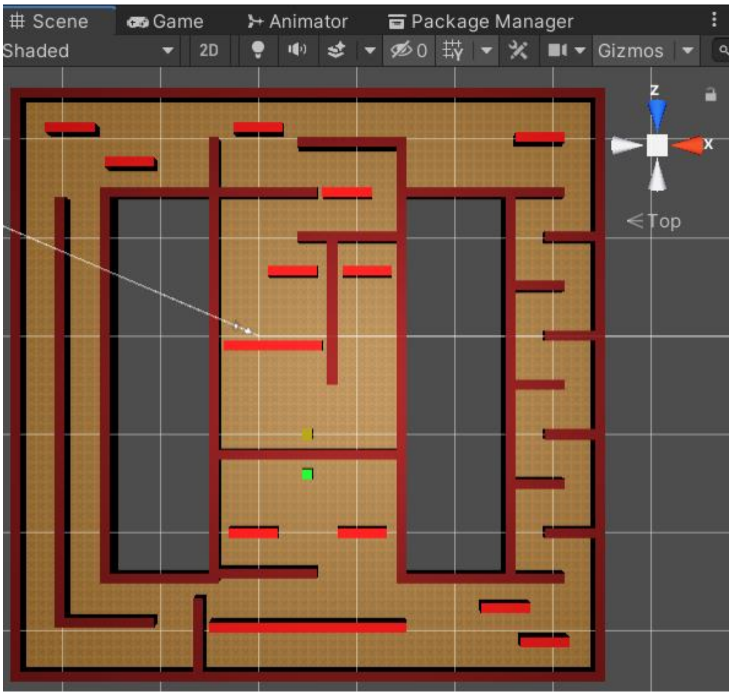
Scene 7



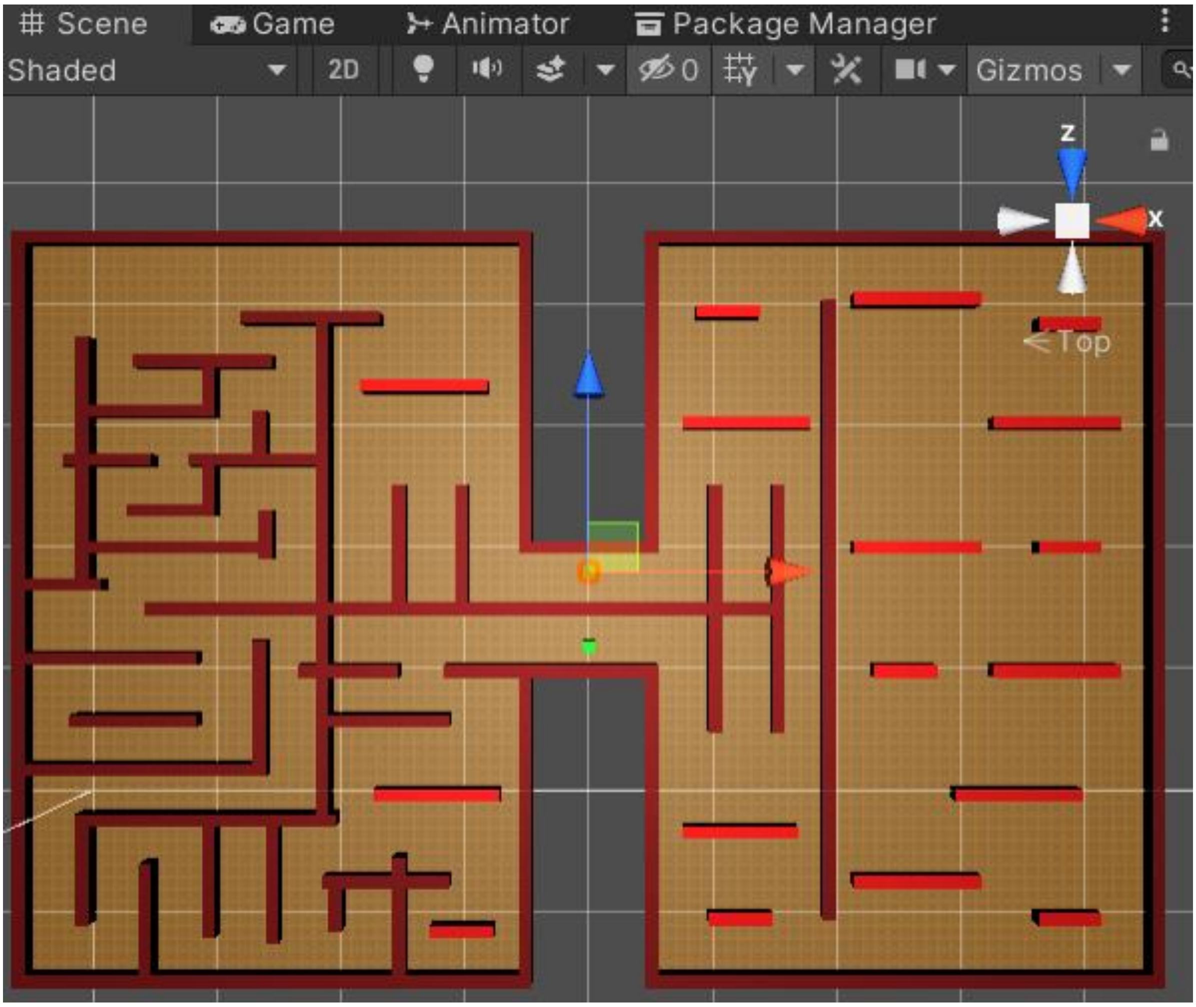
Scene 8



Scene 9



Scene 10



Future Scope

The game can be made more fun by –
1. adding coins,
2. a time counter,
3. saving high scores for each level,
4. change player texture,
5, adding animations

Other than the improvements, touch controls can be implemented and an ANDROID version for the app can be made, which can be later uploaded for GOOGLE PLAY STORE

Other than that, AD space can be created to support Google AdSense for generating money from Google Play Store.

Conclusion

To conclude the project, we have made a completely working 2D game with some features to make it playable and make it bug free. The looks and feels are made simple but nothing is left incomplete with errors.

Bibliography

Project Mentor - Ria Bandyopadhyay

Unity User Manual 2021.3 (LTS) – https://docs.unity3d.com/Manual/index.html

“Brackeys” Youtube Channel - https://www.youtube.com/channel/UCYbK_tjZ2OrIZFBvU6CCMiA

“CodeMonkey” Youtube Channel – https://www.youtube.com/c/CodeMonkeyUnity/featured

“Blackthornprod” Youtube Channel – https://www.youtube.com/channel/UC9Z1XWw1kmnvOOFsj6Bzy2g

“Dani” Youtube Channel – https://www.youtube.com/channel/UCIabPXjvT5BVTxRDPCBBOOQ

“Quick Outline Feature” by Chris Nolet – https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488#description