Game Programming Project 2

Game Name – Pokémon Runner 2D

Github Link: https://github.com/sakothar/Game-Programming.git

1. Game Design

- The 2D platformer game is designed with unity assets and background images which are available for label reuse from google.
- The player starts with 10 health points and score can be as many as berries consumed.
- Score = 20 for one berry consumed.

Three enemy sprites in all the levels:

1. Low level enemy – Eats up 20 health points
2. Medium Damage enemy – Eats up 40 health points
3. High damage enemy- Eats up all the health on contact i.e. 100 Health points

The game consists of three levels as follows:

1. Level One – The player has to collect as many berries as he/she can and pass through the level without losing all 100 health points.

2. Level two – the player has to collect berries and face two level of enemies – Low and medium damage enemy.

3. Level Three- The player has to do the same as he did In the previous levels and in this level we have three doors to clear the way.

   One door takes you back to level 1.
   One door takes you back to level 2.
   One door will lead you to win the game.

2. Script Programming

- Player script

Declare speed, jump and health of the player. The rest of the code explains collision and other aspects of the player.

Game Programming Project 2

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

public class Player : MonoBehaviour
{

    public float speed;
    public float jump;
    public static int health;

    private float moveinput;

    public Animator anim;
    public Rigidbody2D rb;
    private bool faceRight = true;
    private bool isGrounded;
    //private bool isMoving;

    // Start is called before the first frame update
    void Start()
    {

        anim = GetComponent<Animator>();
        rb = GetComponent<Rigidbody2D>();
        health = 100;
    }

    // Update is called once per frame
    void Update()
    {

        moveinput = Input.GetAxis("Horizontal");
        rb.velocity = new Vector2(moveinput * speed, rb.velocity.
        if (moveinput != 0)
```

```
    if (collision.gameObject.tag.Equals("Fire"))
    {
        health -= 10;
        SoundManagerScript.PlaySound("hit");
        Destroy(collision.gameObject);
    }

    if (collision.gameObject.tag.Equals("boss"))
    {
        health -= 40;
        SoundManagerScript.PlaySound("hit");
        Destroy(collision.gameObject);
    }

    if (collision.gameObject.tag.Equals("finalboss"))
    {
        health -= 60;
        SoundManagerScript.PlaySound("hit");
        Destroy(collision.gameObject);
    }

    if (collision.gameObject.tag.Equals("Water"))
    {
        health = 0;
        SoundManagerScript.PlaySound("water");
        SoundManagerScript.PlaySound("death");
        //Destroy(gameObject);
    }
}
```

- Score Mechanism

Game Programming Project 2

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

public class ScoreScript : MonoBehaviour
{
    public static int scoreValue = 0;
    Text score;

    // Start is called before the first frame update
    void Start()
    {
        score = GetComponent<Text>();
    }

    // Update is called once per frame
    void Update()
    {
        score.text = "Score: " + scoreValue;
    }
}
```

- Sound Manager Script

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

public class SoundManagerScript : MonoBehaviour
{
    // Start is called before the first frame update
    public static AudioClip jumpSound, playerHitSound, fireSound, GameWinSound, waterSound, playerDeathSound, coinSound;
    static AudioSource audioSource;

    void Start()
    {
        jumpSound = Resources.Load<AudioClip>("Jump");
        playerHitSound = Resources.Load<AudioClip>("Hit");
        fireSound = Resources.Load<AudioClip>("");
        GameWinSound = Resources.Load<AudioClip>("GameWin");
        waterSound = Resources.Load<AudioClip>("Water");
       // playerDeathSound = Resources.Load<AudioClip>("GameOver");
        coinSound = Resources.Load<AudioClip>("Coin");

        audioSource = GetComponent<AudioSource>();
    }
```

Game Programming Project 2

```
public static void PlaySound(string clip)
{
    switch (clip)
    {
        case "jump":
            audioSource.PlayOneShot(jumpSound);
            break;
        case "win":
            audioSource.PlayOneShot(GameWinSounc
            break;
        case "fire":
            audioSource.PlayOneShot(fireSound);
            break;
        case "hit":
            audioSource.PlayOneShot(playerHitSou
            break;
        case "water":
            audioSource.PlayOneShot(waterSound);
            break;
        case "coin":
            audioSource.PlayOneShot(coinSound);
            break;

    }
}
```

- Button Level Load script- loads a new level on reference/ Creates events

```
1   using UnityEngine;
2   using System.Collections;
3   using UnityEngine.SceneManagement;
4
5   public class UIButtonLevelLoad : MonoBehaviour {
6
7       public string LevelToLoad;
8
9       public void loadLevel() {
10          //Load the level from LevelToLoad
11          SceneManager.LoadScene(LevelToLoad);
12      }
13  }
14
```

- Objective function script

Game Programming Project 2

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

public class Objective : MonoBehaviour
{

    public string levelToLoad;

    private void OnCollisionEnter2D(Collision2D collision)
    {
        if (collision.gameObject.tag.Equals("Player"))
        {
            SoundManagerScript.PlaySound("win");
            SceneManager.LoadScene(levelToLoad);
        }
    }
}
```

- Health script

```
Miscellaneous Files
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using UnityEngine.UI;
5
6   public class HealthScript : MonoBehaviour
7   {
8
9       Text health;
10
11      // Start is called before the first frame update
12      void Start()
13      {
14          health = GetComponent<Text>();
15      }
16
17      // Update is called once per frame
18      void Update()
19      {
20          health.text = "Health: " + Player.health;
21      }
22  }
```

Game Programming Project 2

- Camera follow player script

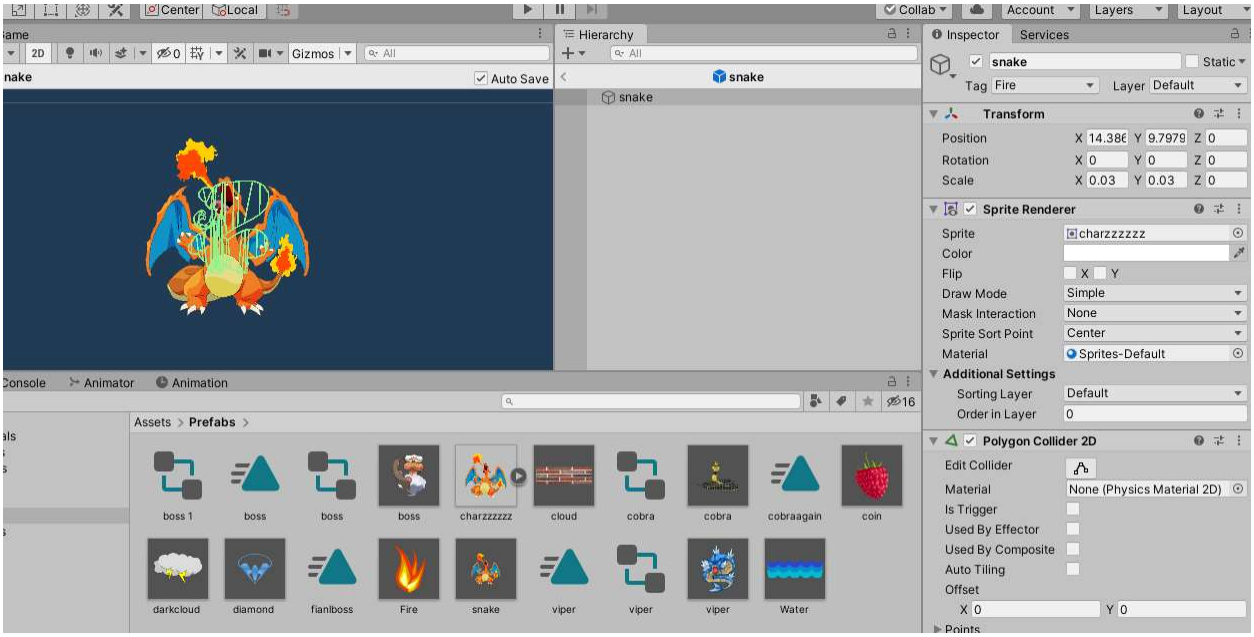
```
using UnityEngine;

public class CameraFollow : MonoBehaviour
{
    public Transform target;
    public float smoothspeed = 0.125f;
    public Vector3 offset;

    private void FixedUpdate()
    {
        Vector3 desiredPosition = target.position + offset;
        Vector3 smoothPosition = Vector3.Lerp(transform.position,desiredPosition,smoothspeed);
        transform.position = smoothPosition;

        //transform.LookAt(target);
    }
}
```

Asset Design

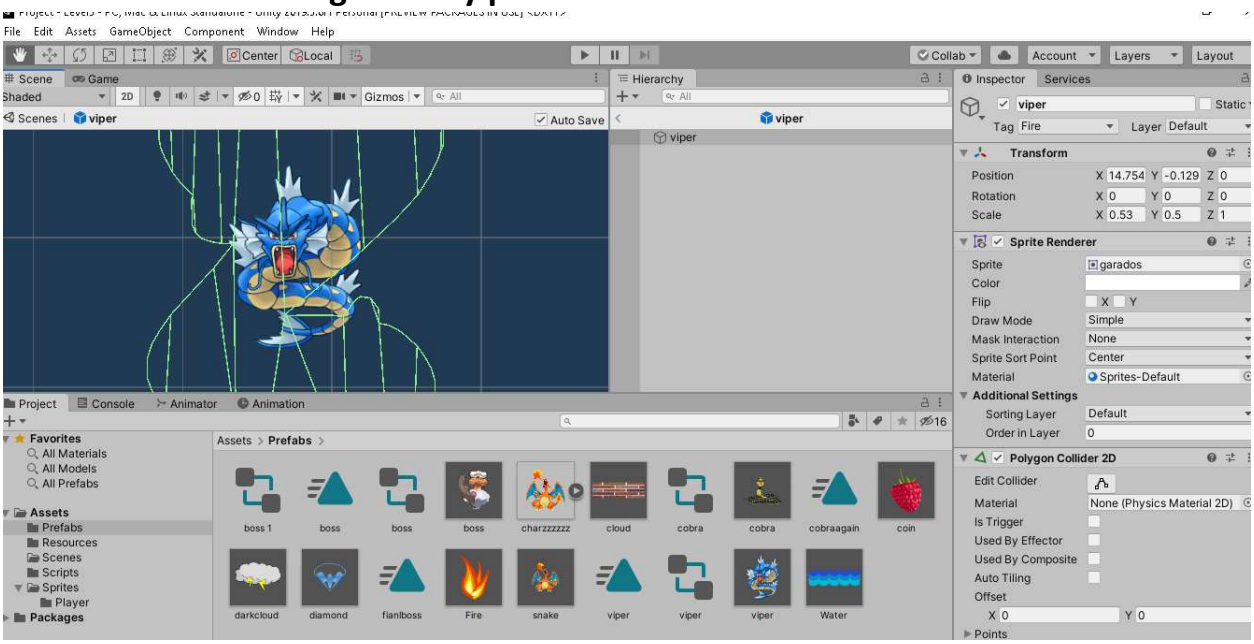
1. **Asset Sprite - Sliced the sprite sheet using Unity's sprite editor**

2. **Low damage enemy Sprite prefab**

Game Programming Project 2

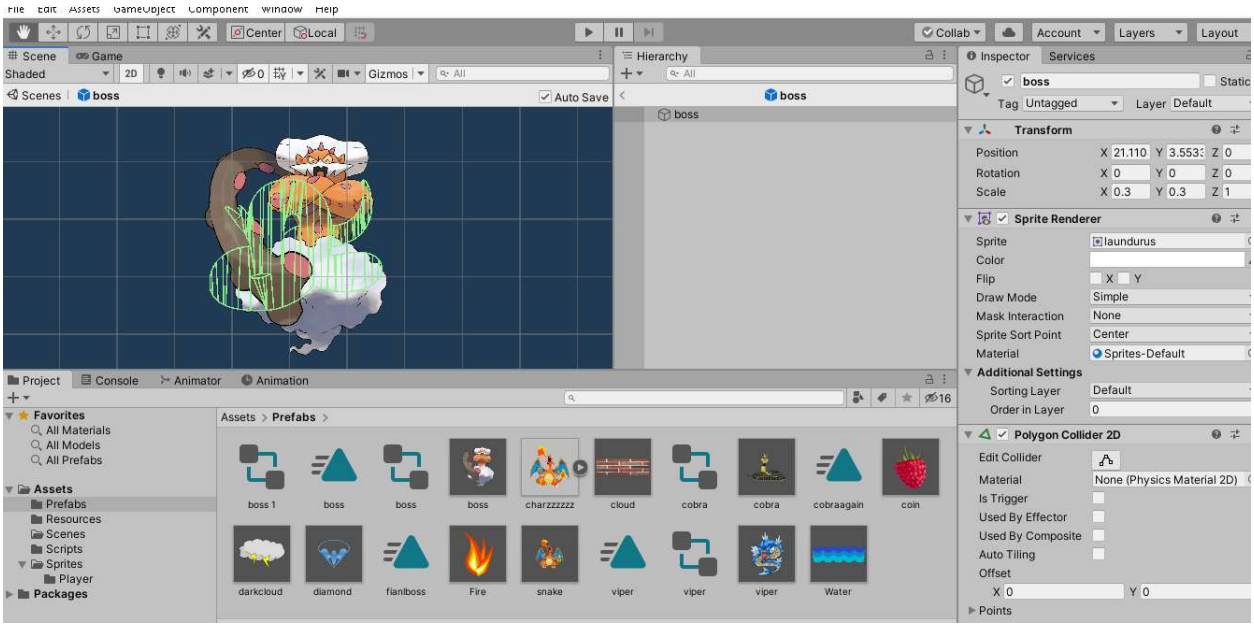


3. Medium Damage enemy prefabs

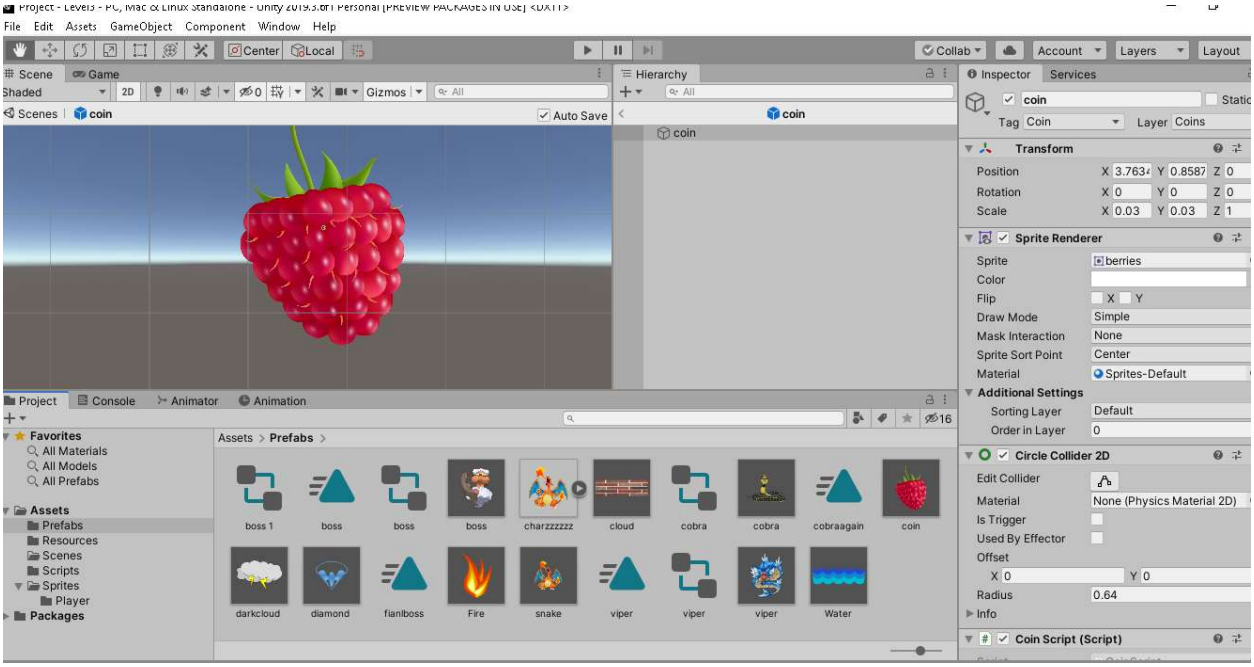


4. High Damage

Game Programming Project 2

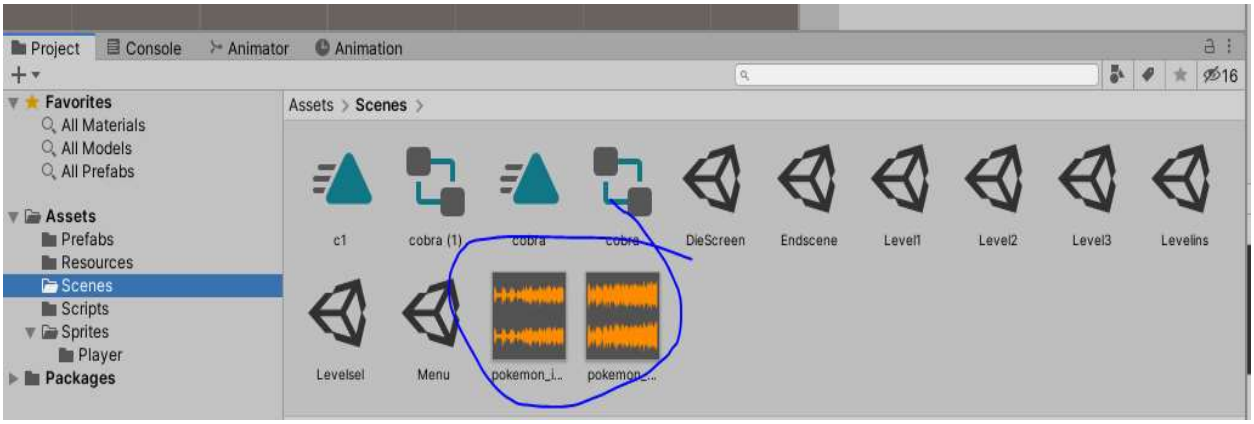


5. Cheery prefab



6. Audio Files used in the game

Game Programming Project 2

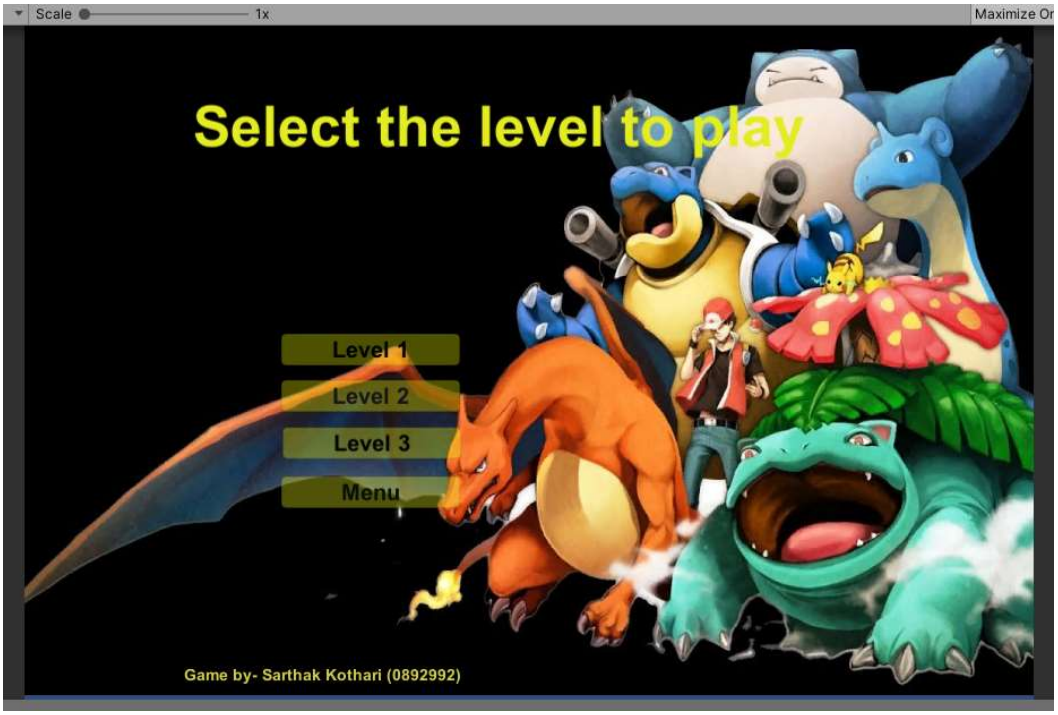


3. Game Run Time Screenshots

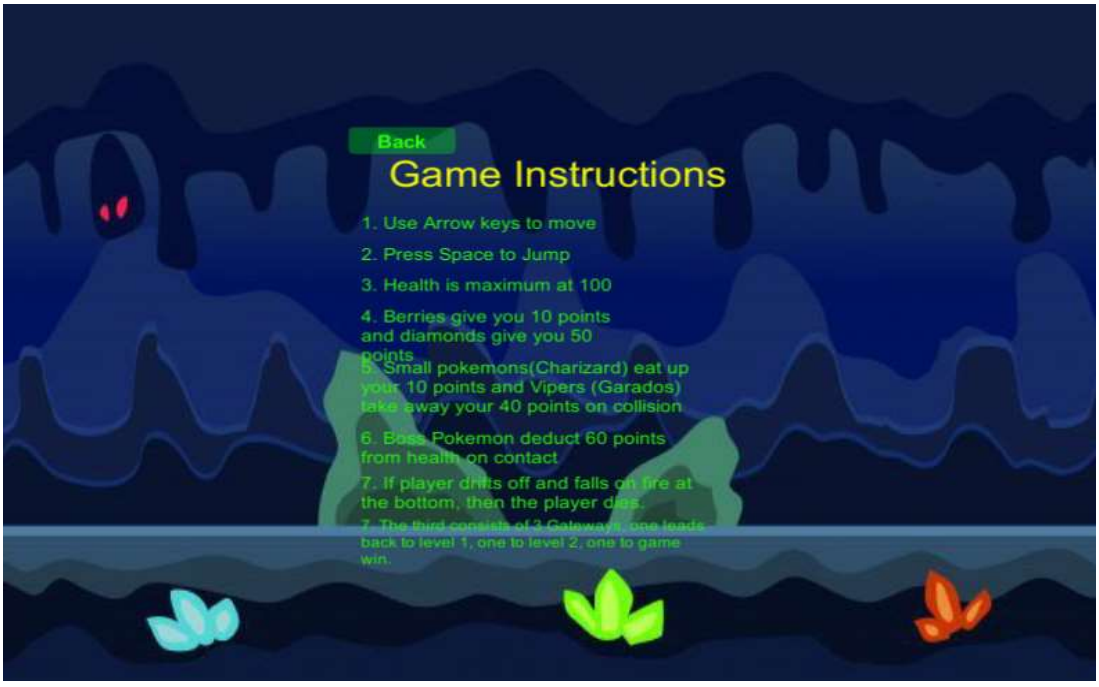
- **Main Menu**

- **Level Selection**

Game Programming Project 2

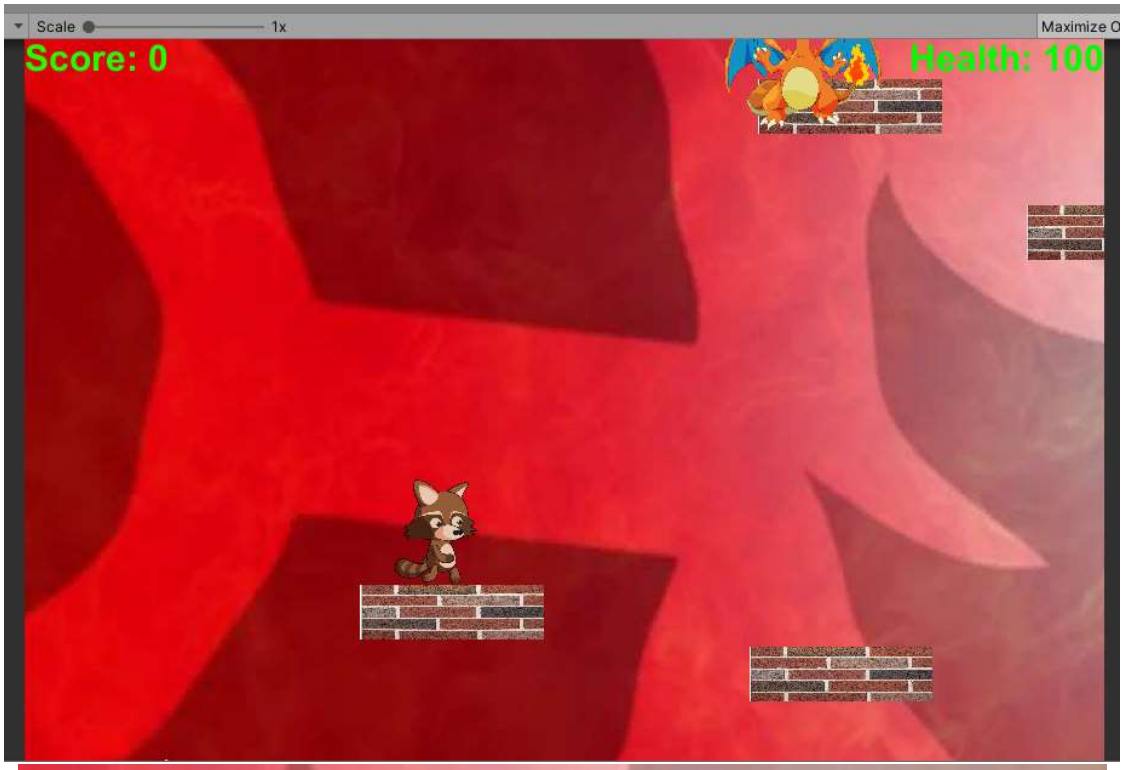


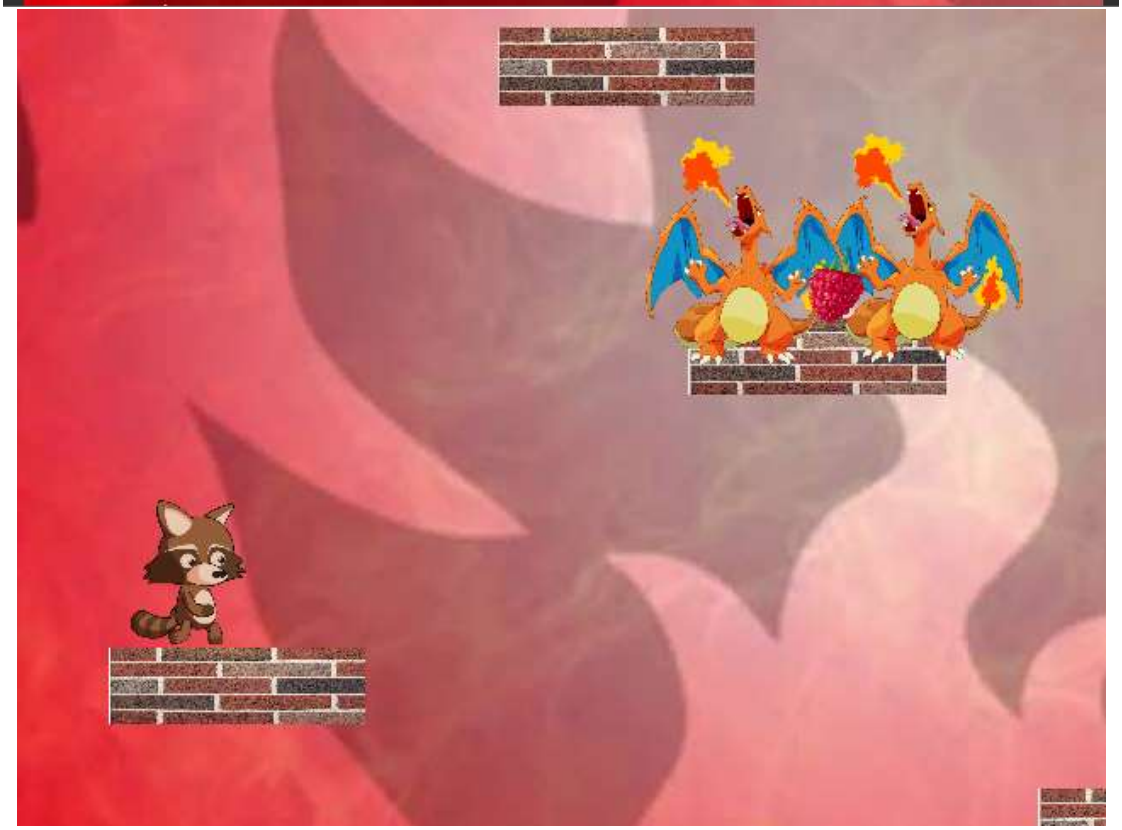
- **In-Game Instructions Screen**



- **Level 1**

Game Programming Project 2



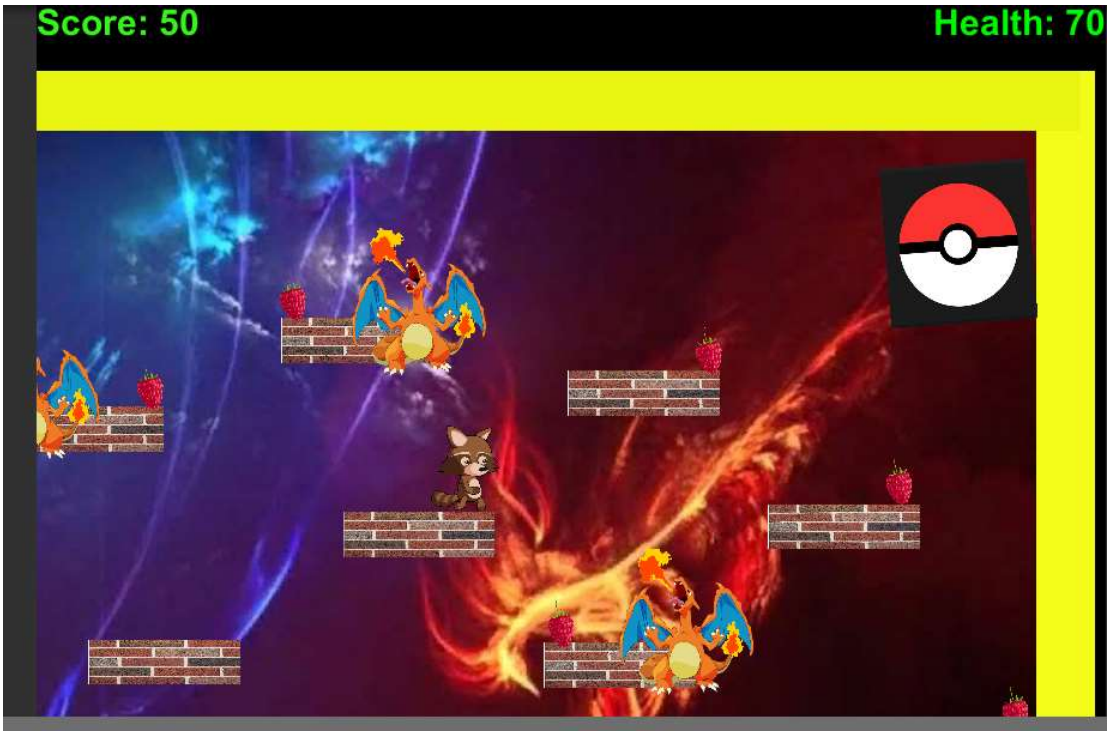


Game Programming Project 2

Level 2

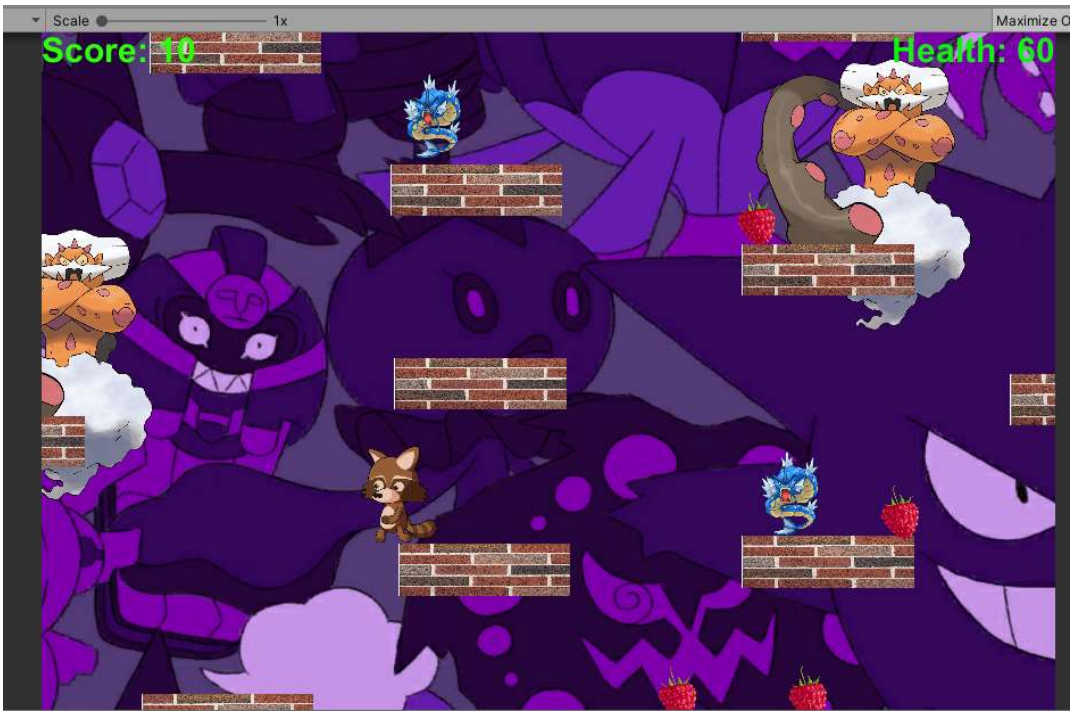
Game Programming Project 2

Game Programming Project 2

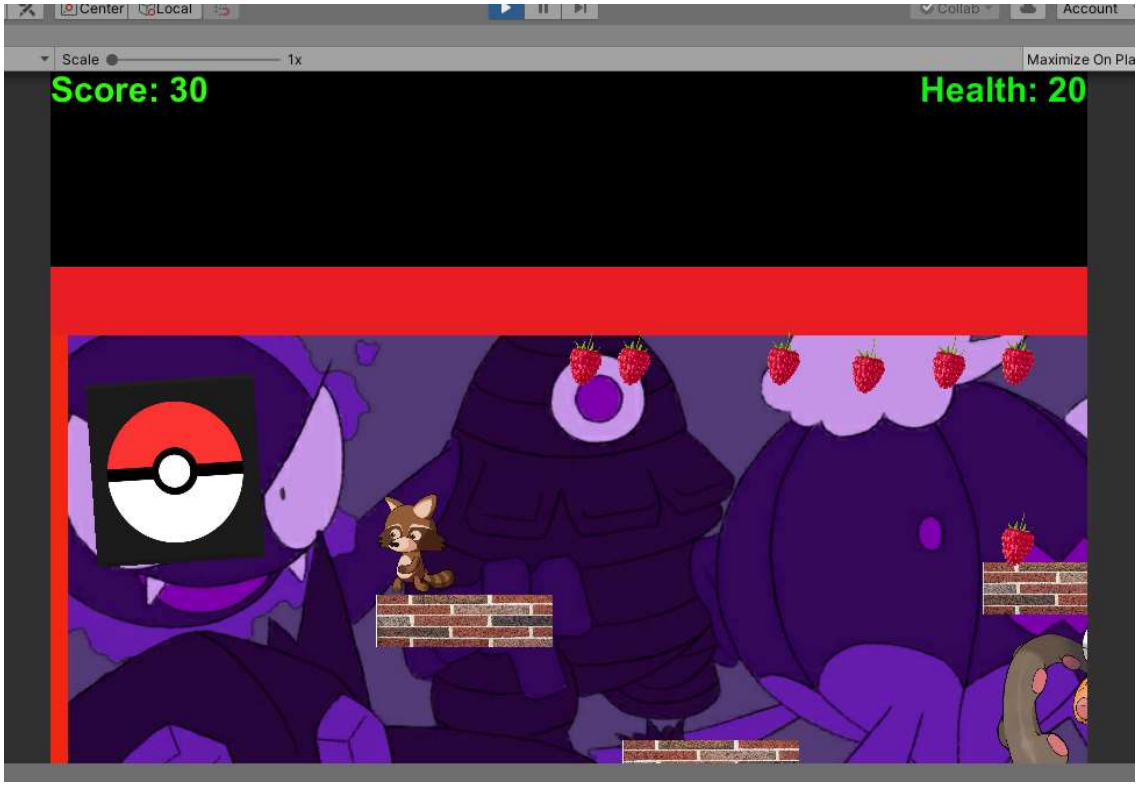


Level 3

Game Programming Project 2

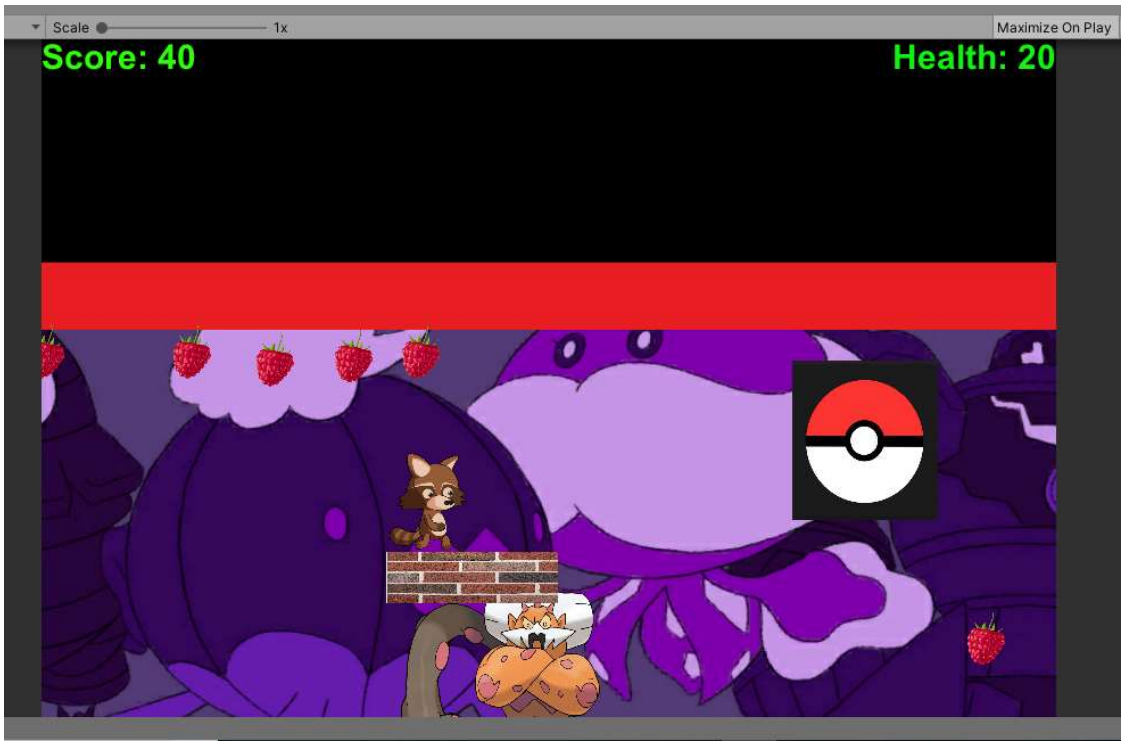


Objective door 1- Takes you back to level 1



Objective – Takes you back to level 2

Game Programming Project 2



Objective door 3 – Takes you to 'Win' screen / Endgame Screen

EndGame Screen

Game Programming Project 2

Player- Dead Screen