Professional Practice in IT Project August

Title of Project:

3D Platformer Game

Names of Group Members:

Faustas Tamulis (G00373028)

Garreth Ward (G00374946)

Supervisors Name:

Daniel Cregg

GitHub Repository Link:

https://github.com/faustastamulis/PPITAugustProject.git

YouTube Screencast Link:

https://youtu.be/8TXEmOj_ack

One Drive Link to Unity Source Code (Files were too big to upload to GitHub):

https://galwaymayoinstitute-my.sharepoint.com/:f:/g/personal/g00373028_gmit_ie/EkVJkUsAquxIteUjyPXIhkQBfnukEnSgLCEBYoo90cmpLA?e=JASNeN

Overview:

Project Requirements:

To develop a 3D Platformer Game using Unity

Technologies Used:

Unity 2019

Visual Studio 2017

Methodology:

We learned how to make games with unity from a module in college called Mobile Applications Development, but it was a 2D platformer game we made in the labs. We never tried making a 3D game, so we decided to learn how to make a 3D version of a platformer game. We wanted to involve the same principals as a regular platformer game.

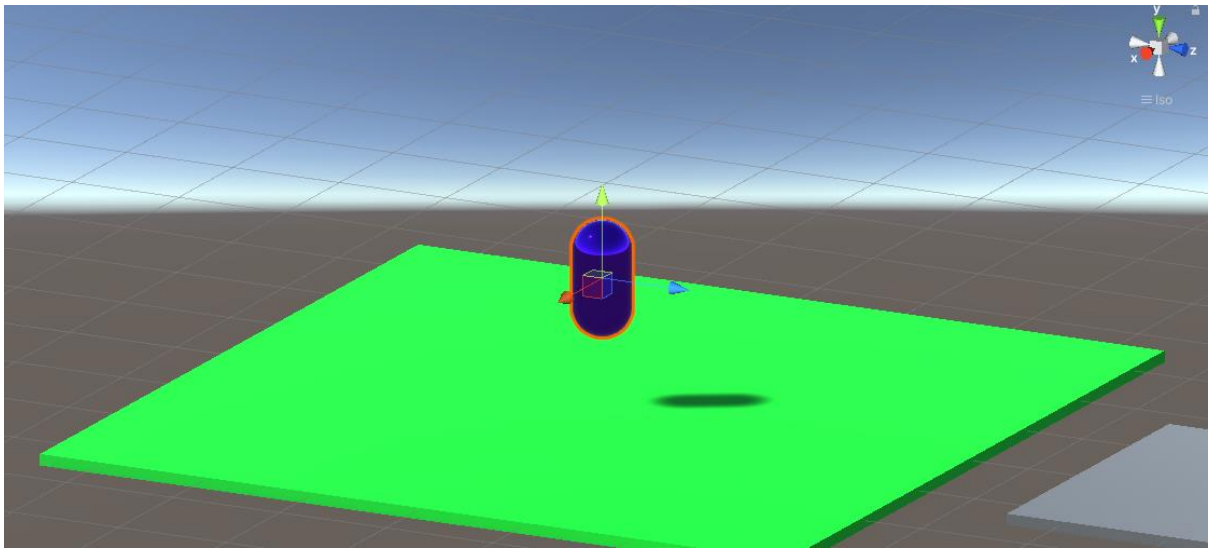
Architecture:

We broke down the making of the game into 10 Easy Steps:

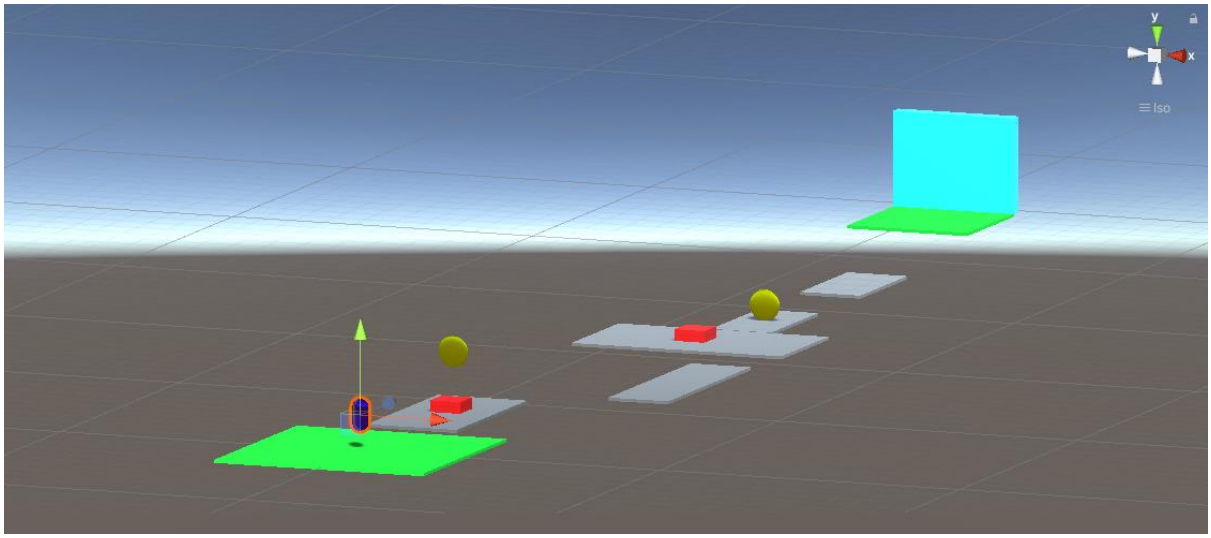
1. **Setting up Unity and all the project settings.**

   Here we downloaded unity from their website and created a new project called 3D Game. When we done this, we set up a few settings that was needed for the project.

   After we finished setting everything up it was time to start making the game. We started off with making a player and the floor the player was standing on and made sure to use appropriate names for these.



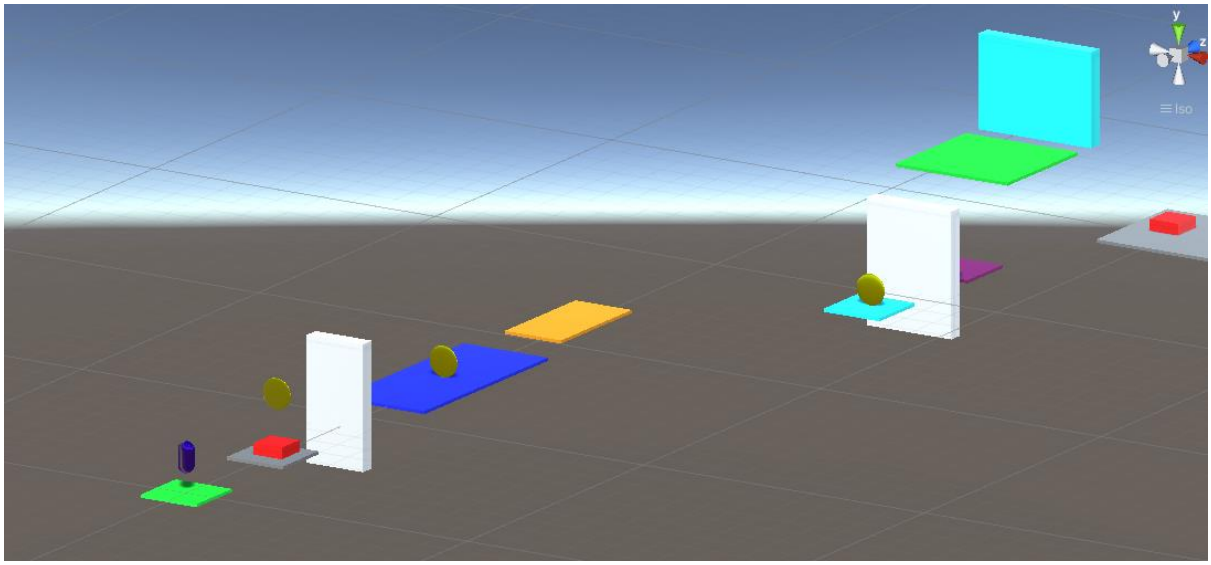
2. **Making the first level.**
   After we made the player and some floors it was ready to set up our first level, we decided that at the start it didn't really matter where anything went because we were going to design the levels properly at the end when we had everything done. So, we just set up a basic level.



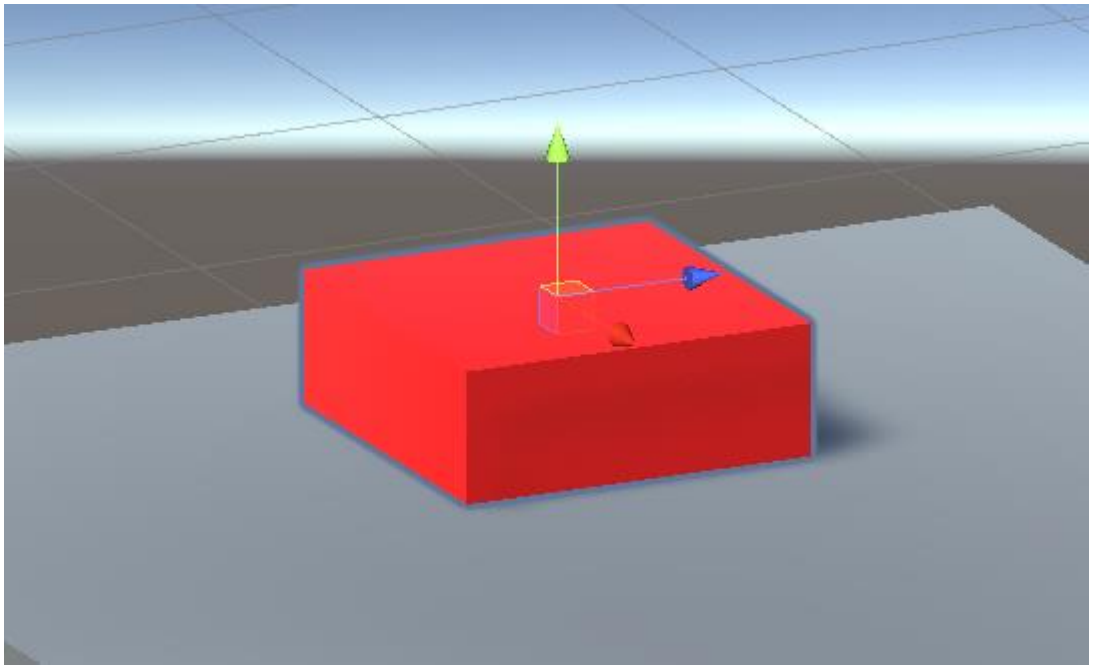
3. **Adding Player Movement.**
   Once we had the player movement setup, we had to focus on adding player movement for a smooth user experience this involved writing some C# script files to tell the program what we wanted the player to do. We started off with making an organised scripts folder where we kept all the scripts we have written. Thankfully majority of the code we had to write was mostly the same as the 2d platformer game we made with a few adjustments we got a proper player movement script.

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

public class PlayerMovement : MonoBehaviour
{
    Rigidbody rb;
    [SerializeField] float movementSpeed = 6f;
    [SerializeField] float jumpForce = 5f;

    [SerializeField] Transform groundCheck;
    [SerializeField] LayerMask ground;

    [SerializeField] AudioSource jumpSound;

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

    // Update is called once per frame
    void Update()
    {
        float horizontalInput = Input.GetAxis("Horizontal");
        float verticalInput = Input.GetAxis("Vertical");

        rb.velocity = new Vector3(horizontalInput * movementSpeed, rb.velocity.y, verticalInput * movementSpeed);

        if (Input.GetButtonDown("Jump") && IsGrounded())
        {
            Jump();
        }
    }

    void Jump()
    {
        rb.velocity = new Vector3(rb.velocity.x, jumpForce, rb.velocity.z);
        jumpSound.Play();
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("Enemy Head"))
        {
            Destroy(collision.transform.parent.gameObject);
            Jump();
        }
    }

    bool IsGrounded()
    {
        return Physics.CheckSphere(groundCheck.position, .1f, ground);
    }
}
```

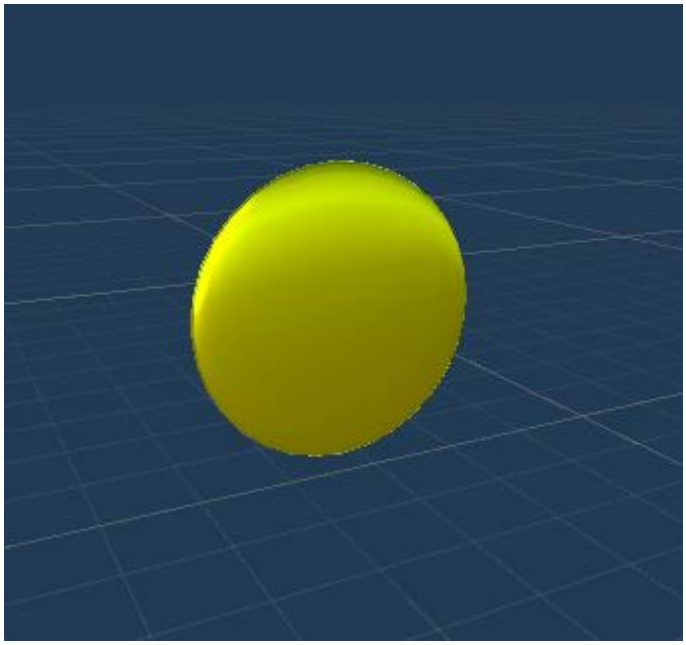
4. **Adding Platform Movement.**
   We wanted this platformer game to be challenging so we made some moving platforms for the player to jump on we tried to figure out how to make the game more challenging for the user, so we came up with the ability to fall off the map certain obstacles like moving platforms and walls.



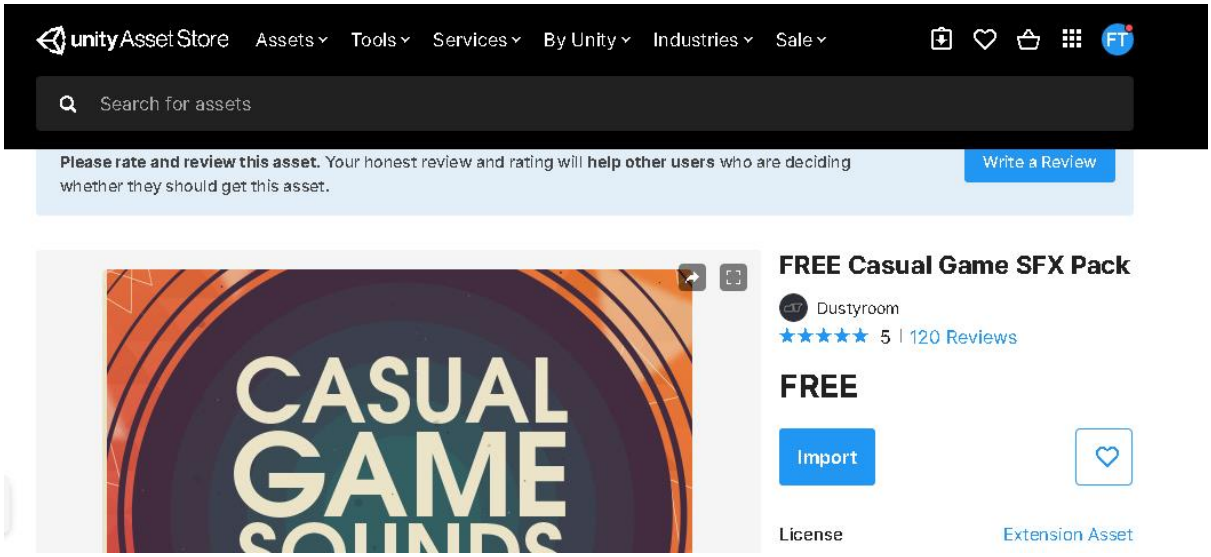
5. **Adding Enemy Movement.**
   Another challenge we decided to make is to add enemies and they would have the ability to move around. So, if the player hit into them, they would go back to the start. We done this by adding two waypoints for the enemy to move back and forth on.



6. **Adding Collectible Items.**
   We thought we needed to add some sort of collectible items also, so we decided to add coins to the game. They also can spin around.



7. **Adding Sounds and Music.**
   We played the game a few times and realised the game isn't much fun without any music and sound effects, so we decided to look up some music and sounds on the asset store in unity. We found some background music that fits the platformer style game and added a death sound coin sound and jump sound to the game.



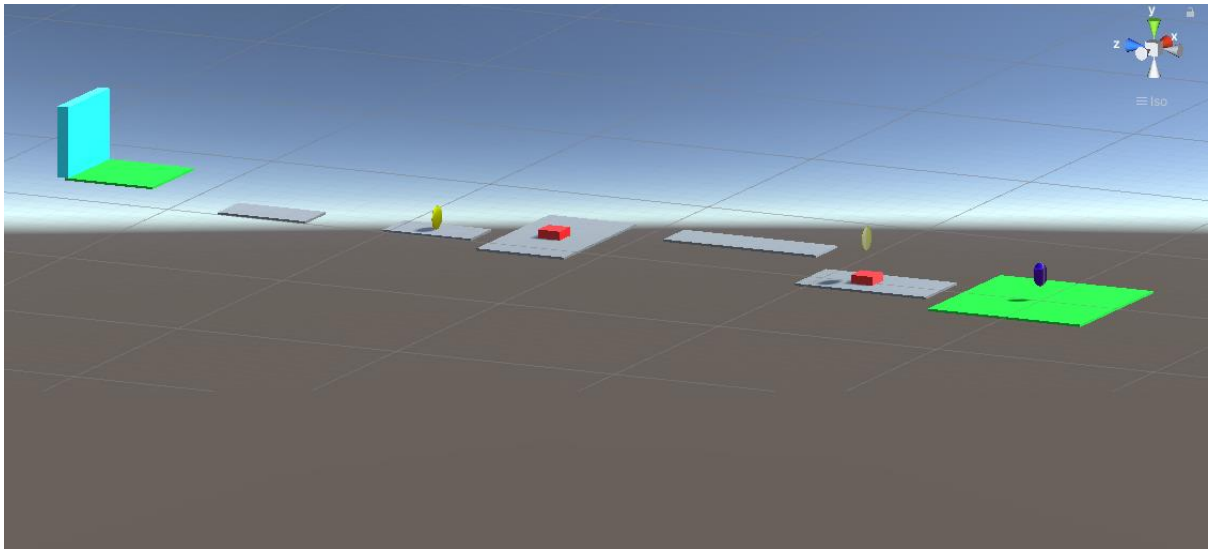
8. **Planning the levels.**

   At this point we pretty much had everything set up in prefabs ready to use to make up our levels, but we didn't want to just free style it we wanted to plan out the levels we decide we should make three levels ranging from easy to difficult. The first level was just some basic platforms and enemies. The second level consisted of all moving

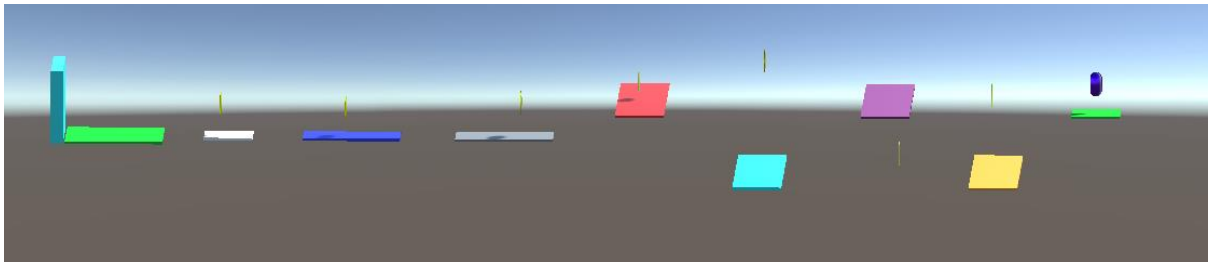
platforms. The last level had pretty much everything walls enemies' coins moving platforms.

9. **Putting the levels together.**

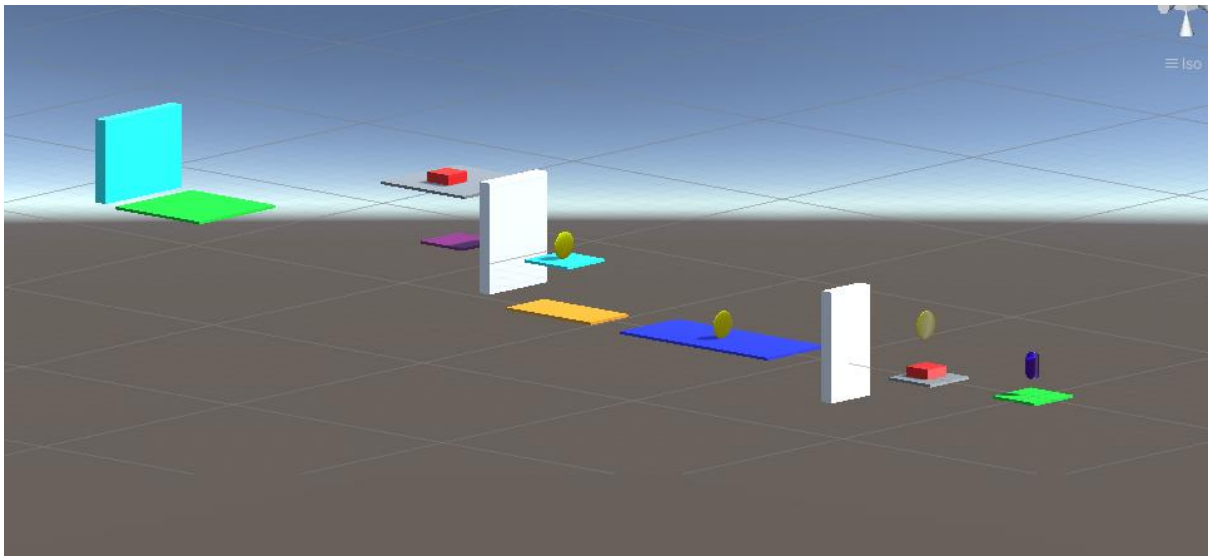
Level 1:



Level 2:



Level 3:



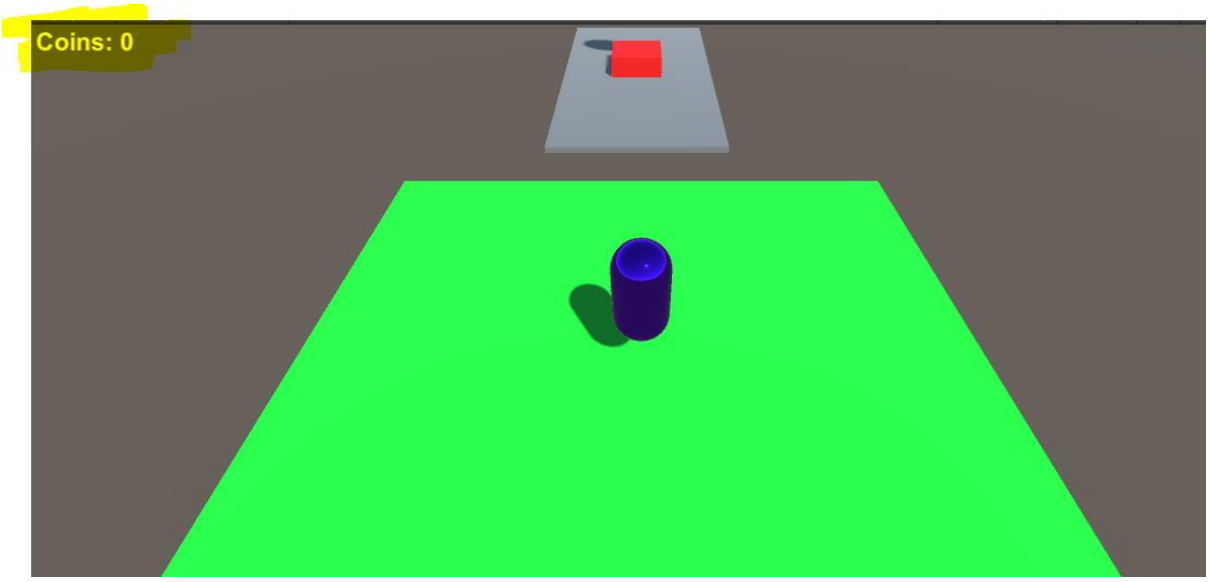
10. **Adding the Start and Ending Menu's.**

We needed to add UI elements to the game like a start menu and an ending menu. So, we made a basic Canvas for the two menus. We also added a counter for the coins in the top left-hand side of the screen.

Start of the game:

End of the game:

The coin counter:



Test Plans and Bugs:

We ran into a lot of numerous bugs while doing this project like the count messing up and the player sliding off the map and falling over. When doing game development there are a lot of bugs you will face when coding these types of games but trying to break it down into steps and looking up online for solutions helped us tremendously.

Conclusion:

Overall, this was a very basic project which we could of added a lot more to and the possibilities with game development is endless we had ideas of adding proper sprites, scoreboard more levels more enemies, abilities and so much more. But we never designed a 3D game before, and we really wanted to learn how to do it so we looked up tutorials and used some of our own knowledge we learned from college and put it together to make this project. In other words, we really enjoyed taking on this task and we are happy with the game we produced.