

Capsule-Man 2

By: Alex Fratepietro





Introduction

- Why I decided to make a game
- Playthrough
- The creation process
- Reflection



Playthrough

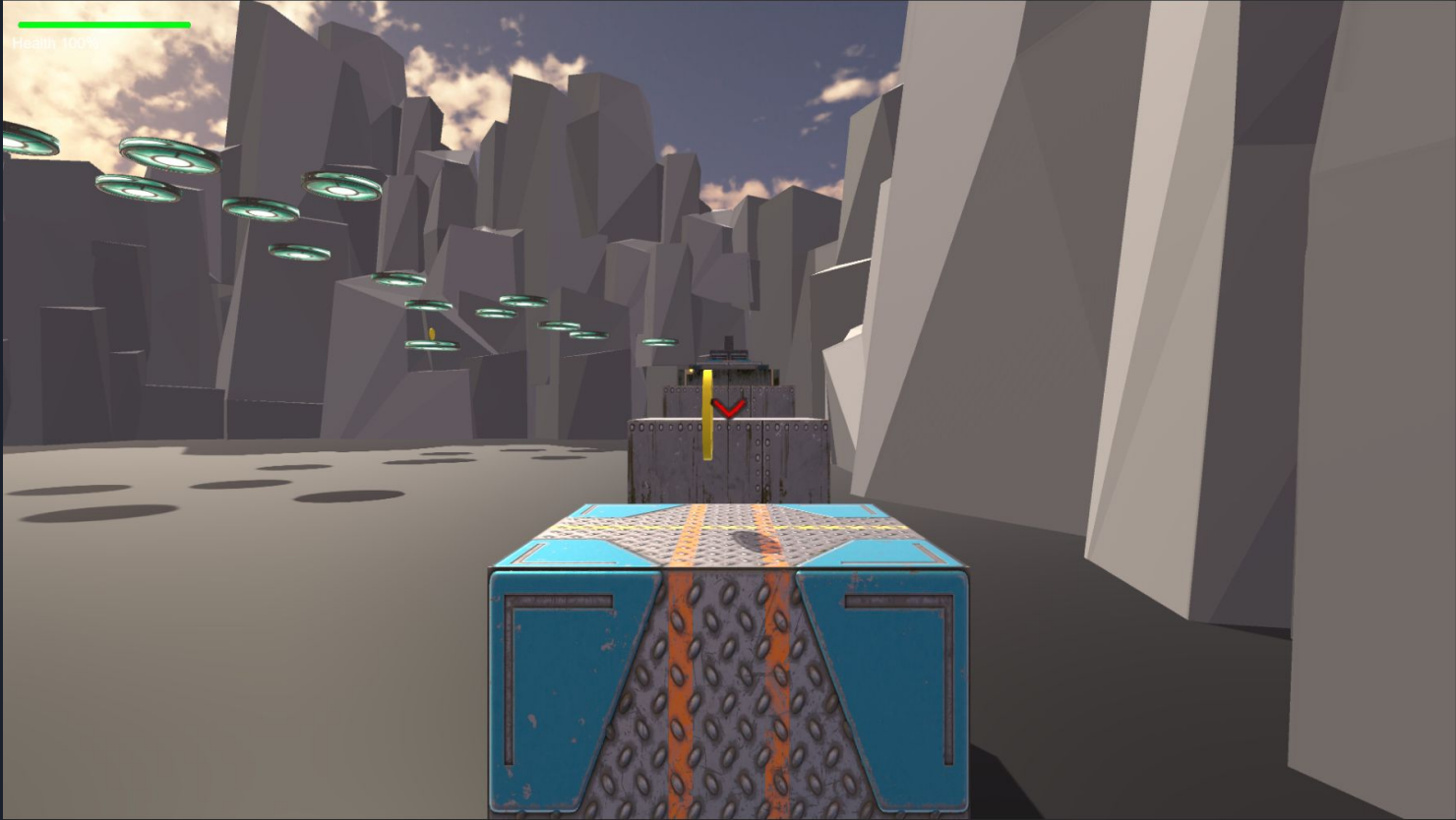




Learning Unity and C#

- The internet
- Youtube
- Unity API documentation
- Stack overflow
- Expert from expert night

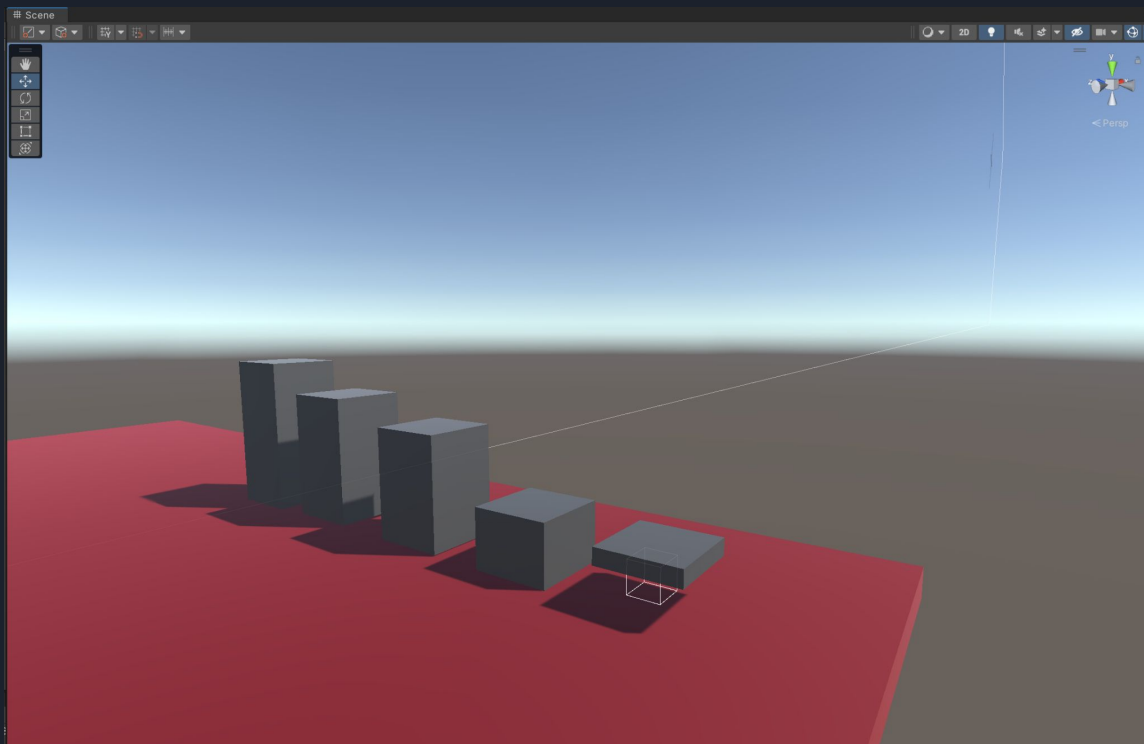
Level 1 Design Process



Start Simple

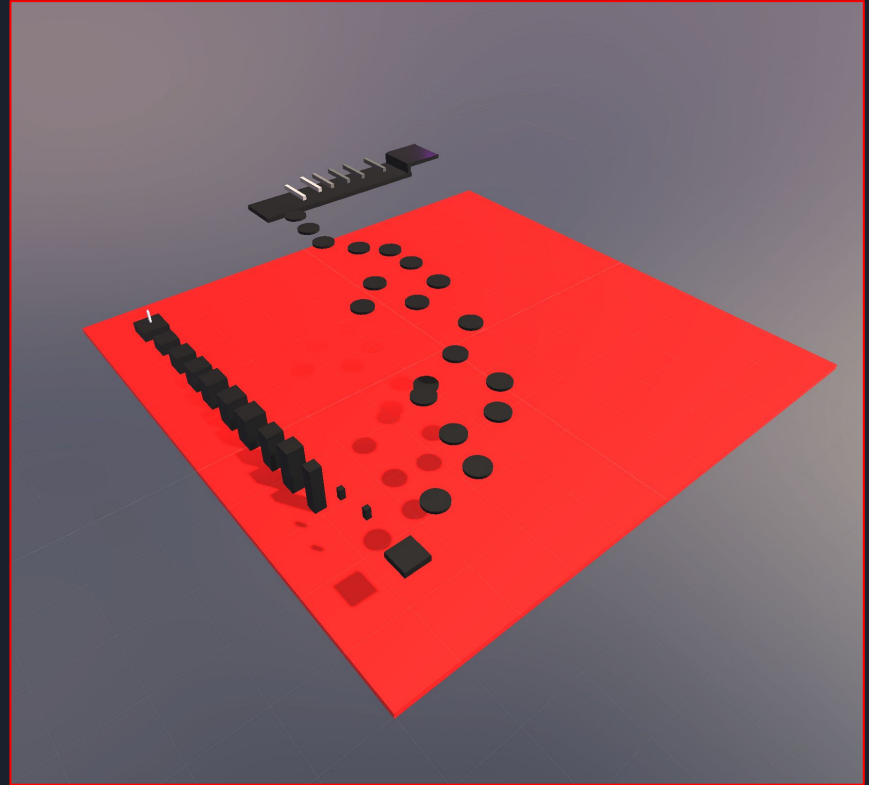
- No textures or models
- Just necessities
- Make sure everything is working

```
private void HandlePlayerMove() {  
    // Move direction directly from axes  
    float deltaX = Input.GetAxis("Horizontal") * moveSpeed;  
    float deltaZ = Input.GetAxis("Vertical") * moveSpeed;  
    _moveDirection = new Vector3(deltaX, _moveDirection.y, deltaZ);  
    // Accept jump input if grounded  
    if (_charCont.isGrounded) {  
        if (Input.GetButton("Jump")) {  
            _moveDirection.y = jumpSpeed;  
        } else {  
            _moveDirection.y = 0f;  
        }  
    }  
    ApplyMovement();  
}
```



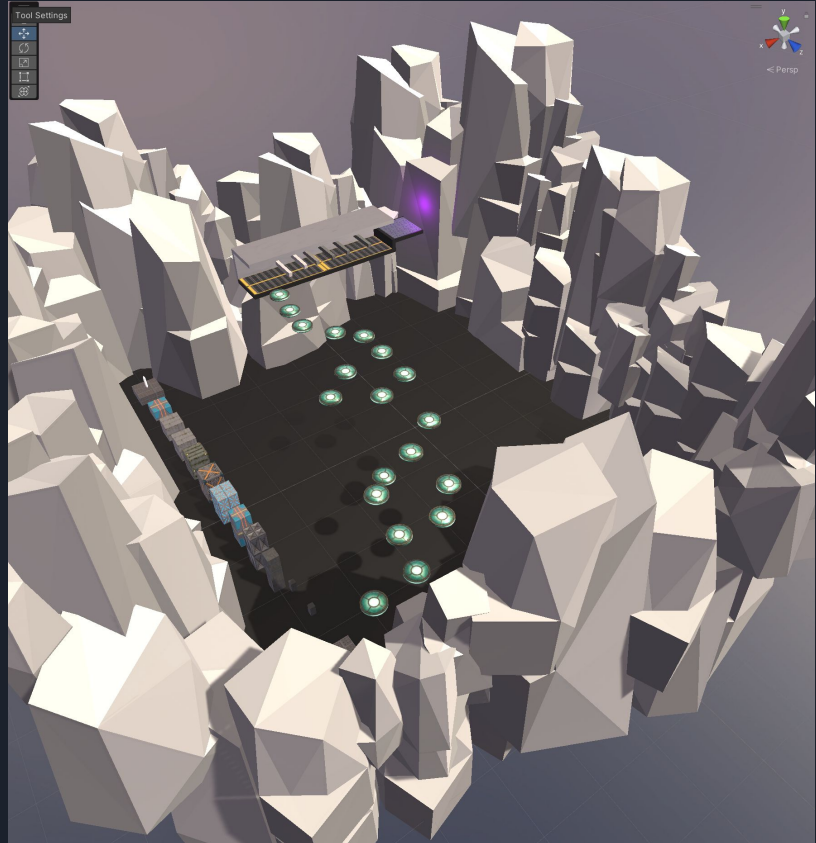
Finish Map

- Layout of full level map
- Still no textures or models
- Completion of parkour



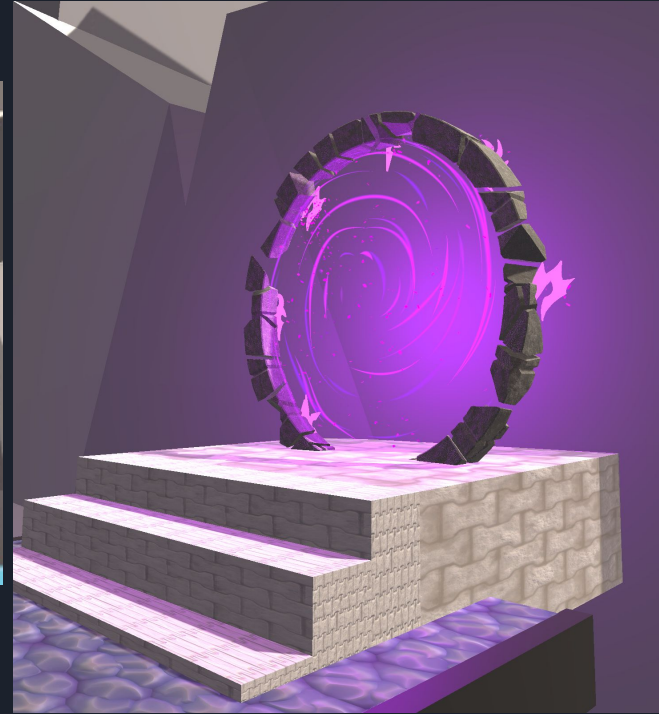
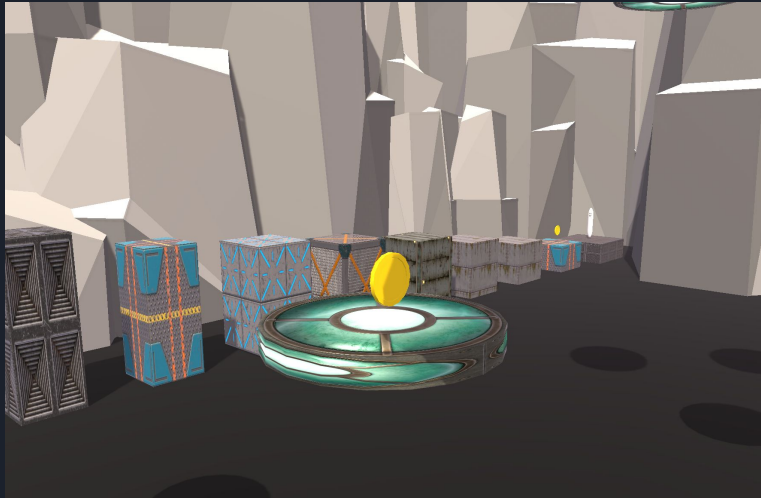
Add Textures and Map Design

- Add textures and models
- Rocks and jump textures



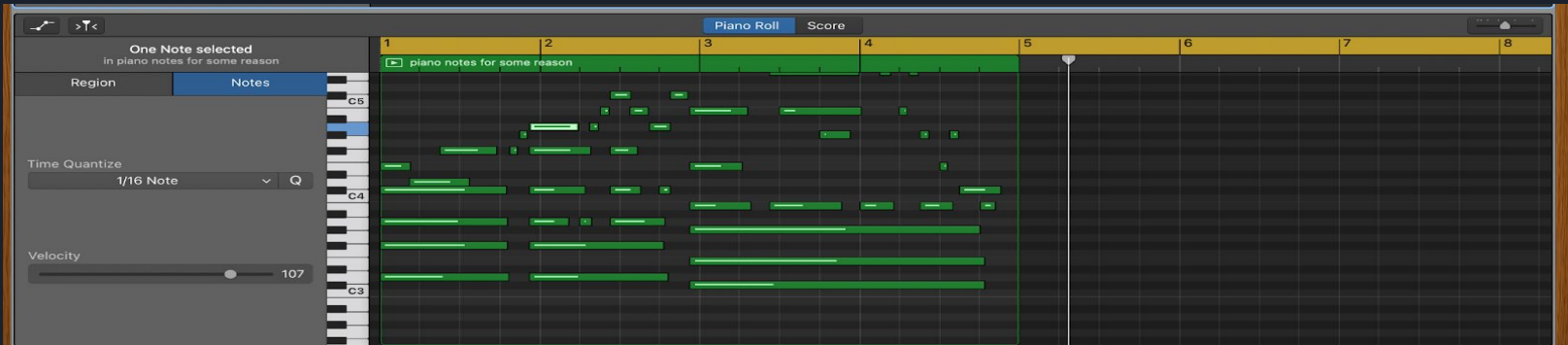
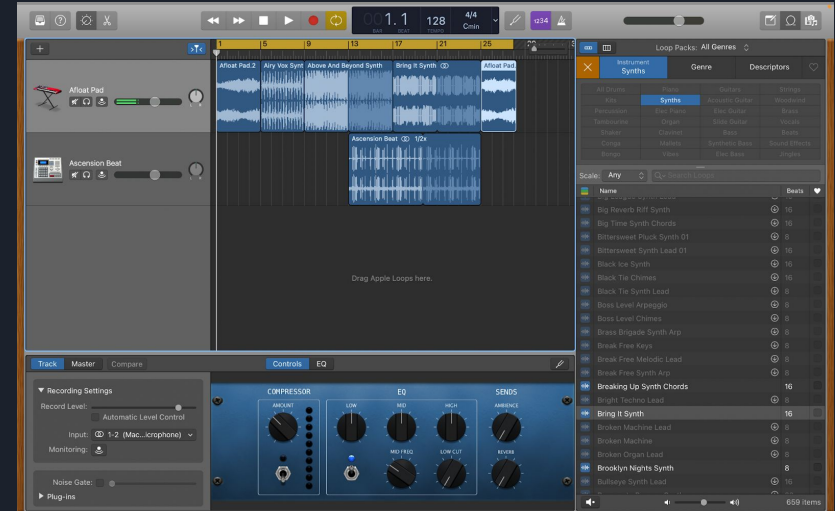
Add Details, Extras and UI

- Link to other levels
- Crosshair User Interface (UI)
- Coins



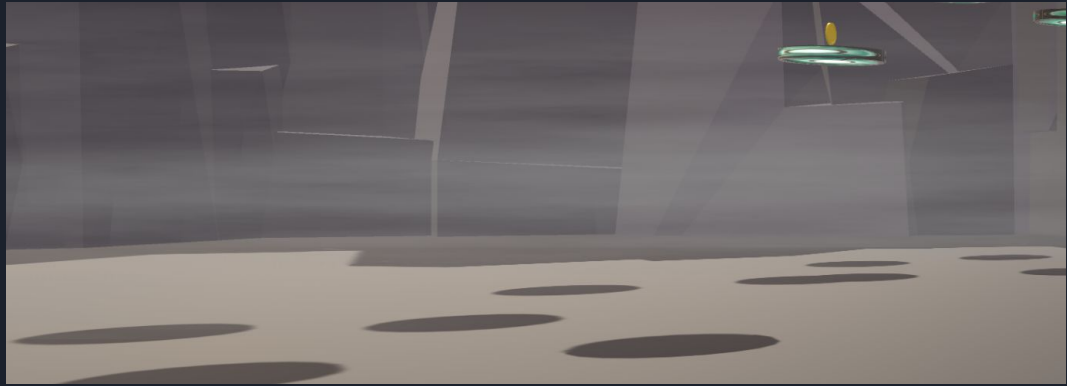
Music

- Music done by Joe Jackson
- Different genres for each level
- Menu theme



Particle Packages

- Add interest to visual effects
 - Fog
 - Snow



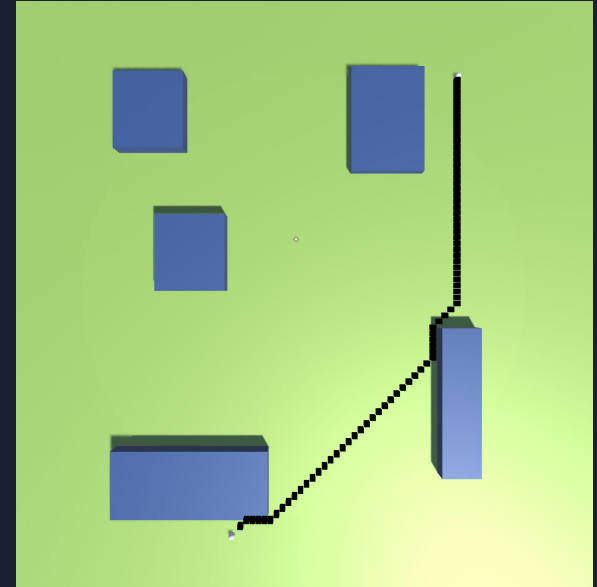
Enemy Pathfinding Process

-Three numbers:

- G cost
- H cost
- F cost

-Nav-Mesh

```
foreach (Node neighbour in grid.GetNeighbours(currentNode)){  
    if (!neighbour.walkable || closedSet.Contains(neighbour)){  
        continue;  
    }  
    int newMovementCostToNeighbour = currentNode.gCost + GetDistance(currentNode, neighbour)  
    if (newMovementCostToNeighbour < neighbour.gCost || !openSet.Contains(neighbour)){  
        neighbour.gCost = newMovementCostToNeighbour;  
        neighbour.hCost = GetDistance(neighbour, targetNode);  
        neighbour.Parent = currentNode;  
  
        if (!openSet.Contains(neighbour))  
            openSet.Add(neighbour);  
    }  
}
```

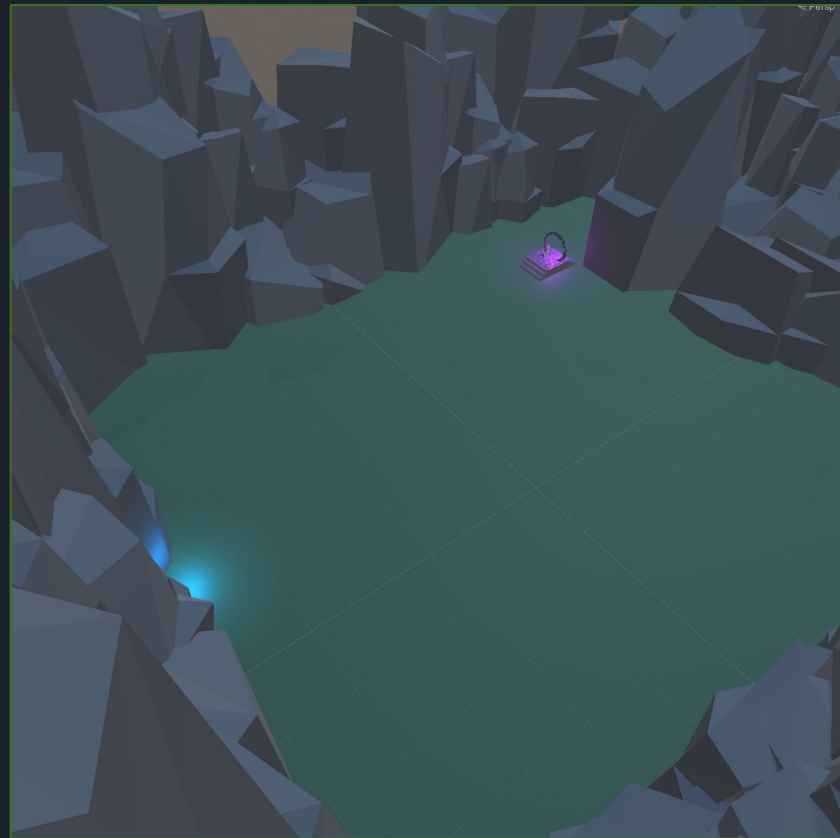


Level 2 Design Process



Use Same Starting Steps

- Same movement
- Similar map designs
- Change in lighting and scenery



Add Weapon and Firing Capabilities

- Required learning about instantiation
- Display ammo in quicktime
- Check for collision with enemies



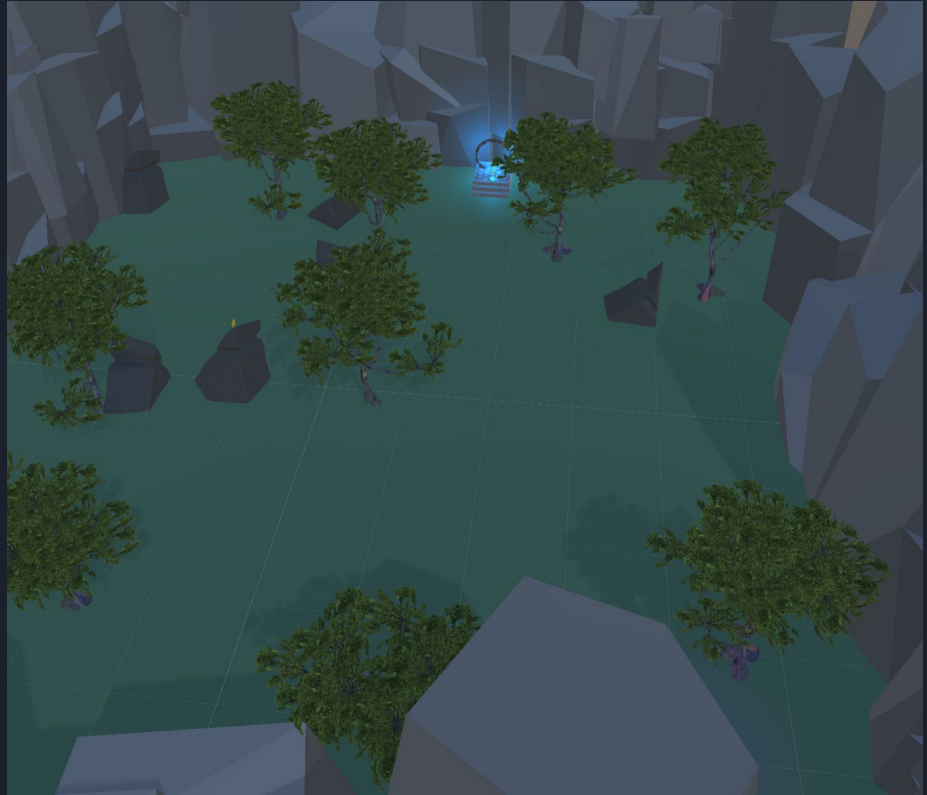
Enemies

- Designed in Blender
- Tracking stats for each enemy
- Pathfinding
- Attack / damage to player



Add Additional Details

- Grenade option
- Extra obstacles (trees and rocks)
- Sky
- Coins
- Music

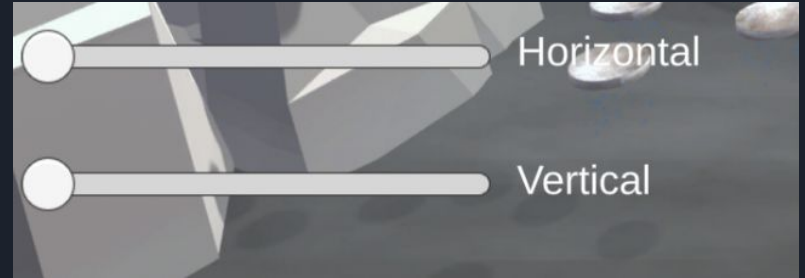


Menu Design Process



Receive Inputs

- Sensitivity inputs
- Button returns



```
public void ChangeH (float val) {  
    PlayerStat.sensH = val;  
}  
public void ChangeV (float val) {  
    PlayerStat.sensV = val;  
}
```



Tips and Stats

- Counters for different stats
- Tips on how to survive

If you are low on health
stand by a medkit to
regain 10 health



Deaths:

KILLS:



Coins:

```
private void Update() {  
    txt.text = "DEATHS: " + PlayerStat.deaths.ToString();  
    stat.text = "KILLS: " + PlayerStat.kills.ToString();  
    coins.text = "COINS: " + PlayerStat.coinCollect.ToString();  
}
```

Putting the Game Online

- Using Simmer.io
- Game address:

https://simmer.io/@_Frate_/capsule



Software Used



blender

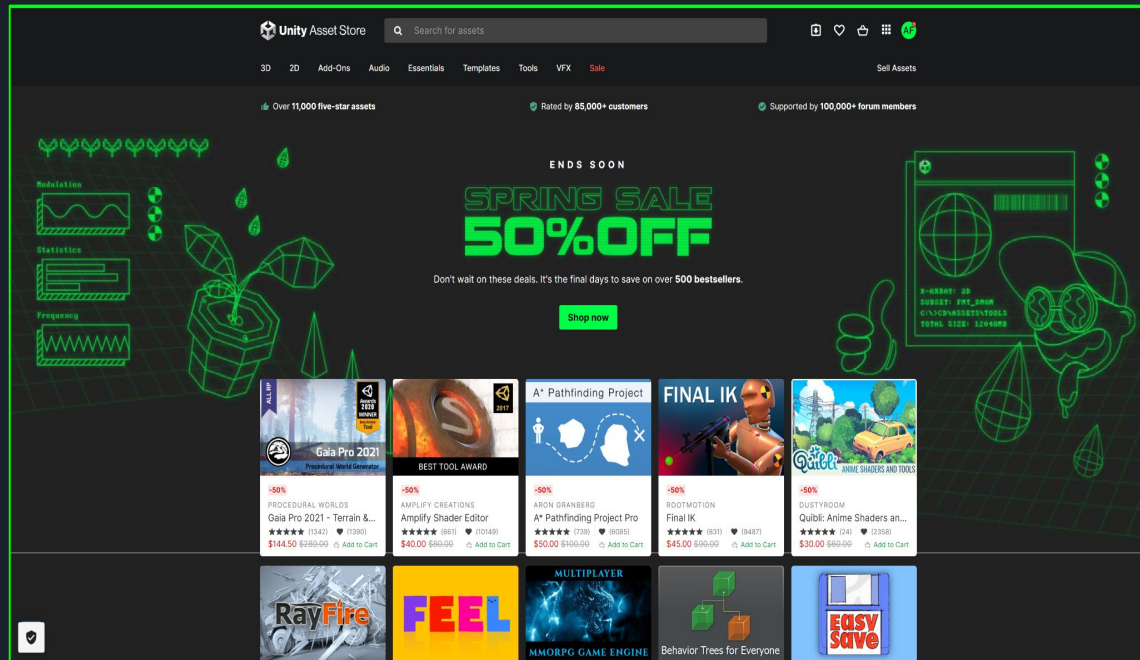


Visual Studio Code




Unity Asset Store

Place to download useful textures
and models



Visual Studio Code

How the game was coded

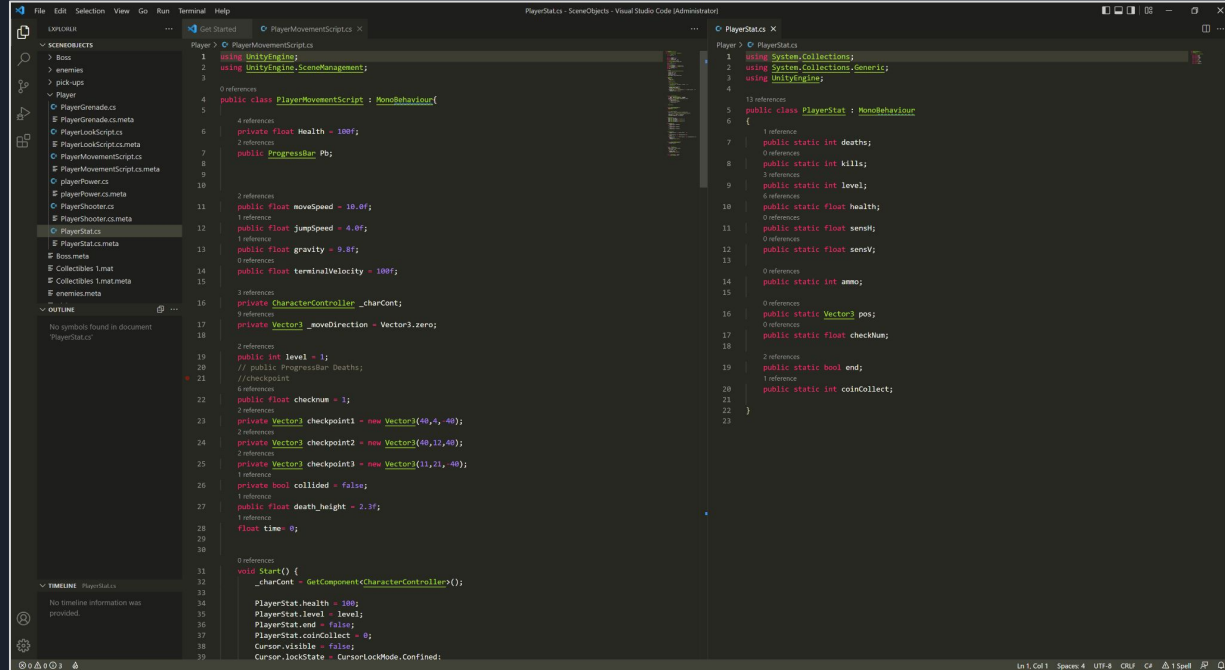


Basic Extension Pack for Unity

Bean Machine Studios | 2,069 | ★★★★★

Tools to make Unity development easier

This extension is enabled globally.



```
using UnityEngine;
using UnityEngine.SceneManagement;

public class PlayerMovementScript : MonoBehaviour
{
    private float Health = 100f;
    public ProgressBar Pb;

    private float moveSpeed = 10.0f;
    public float jumpSpeed = 4.0f;
    public float gravity = 9.8f;
    public float terminalVelocity = 100f;

    private CharacterController _charCont;
    private Vector3 _moveDirection = Vector3.zero;

    public int level = 1;
    // public ProgressBar Deaths;
    // Checkpoint
    public float checknum = 1;
    private Vector3 checkpoint1 = new Vector3(40, 4, 40);
    private Vector3 checkpoint2 = new Vector3(40, 12, 40);
    private Vector3 checkpoint3 = new Vector3(11, 21, 40);
    private bool collided = false;
    private float death_height = 2.3f;
    float time = 0;

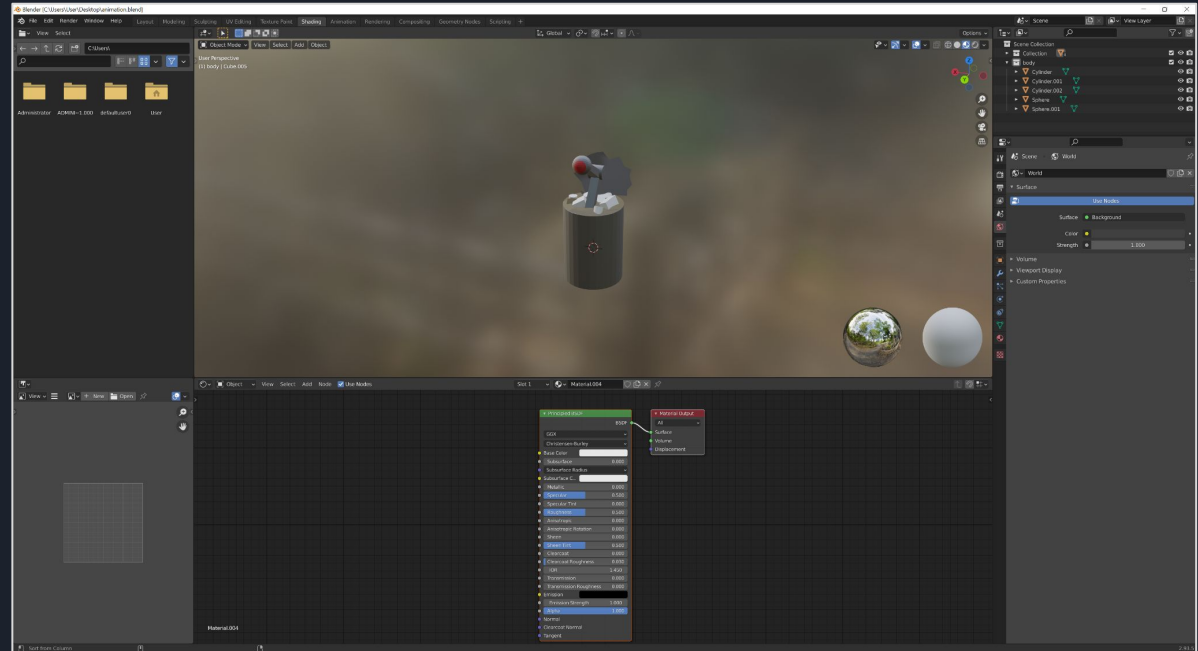
    void Start()
    {
        _charCont = GetComponent<CharacterController>();
        PlayerStat.health = 100;
        PlayerStat.level = level;
        PlayerStat.end = false;
        PlayerStat.coinCollect = 0;
        Cursor.visible = false;
        Cursor.lockState = CursorLockMode.Confined;
    }
}
```

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

public class PlayerStat : MonoBehaviour
{
    public static int deaths;
    public static int kills;
    public static int level;
    public static float health;
    public static float sensu;
    public static float sensu;
    public static int ammo;
    public static Vector3 pos;
    public static float checkNum;
    public static bool end;
    public static int coinCollect;
}
```

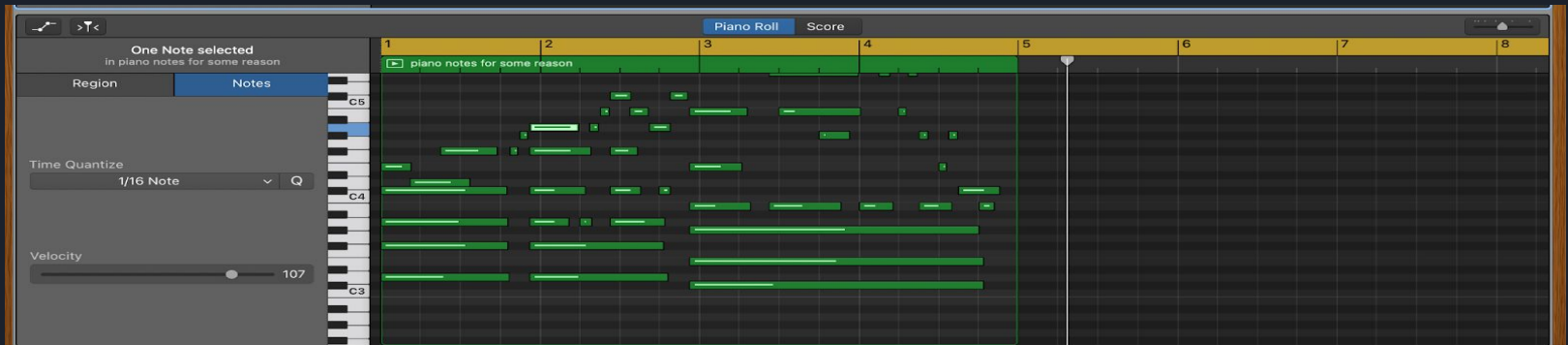

Blender

How models were created, edited
and textured



Garage Band

- Easy to use
- Knew the basics



With More Time...

- Final Boss Level
- Rendering settings / graphics
- Database stats such as High Score

