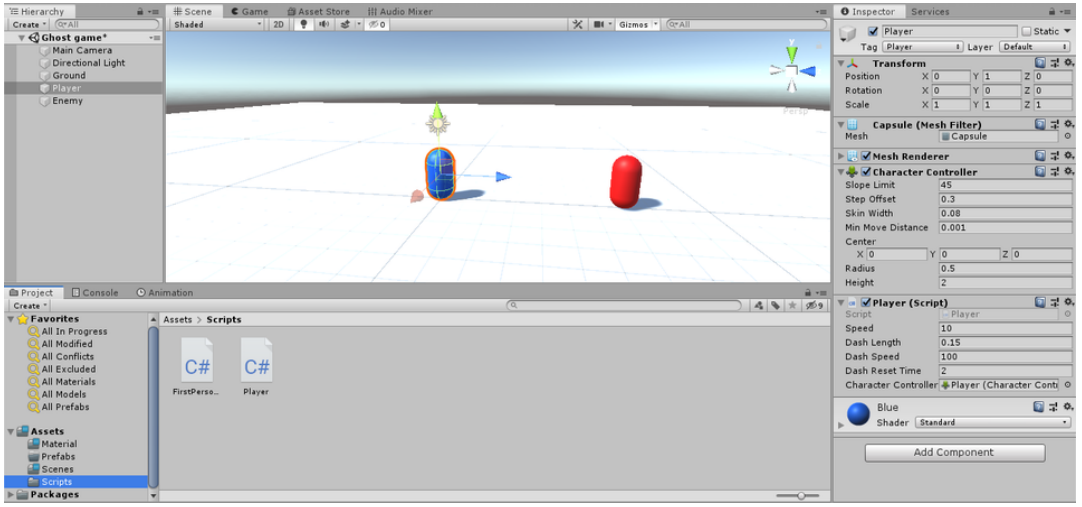
Production Journal

Diogo Constantino



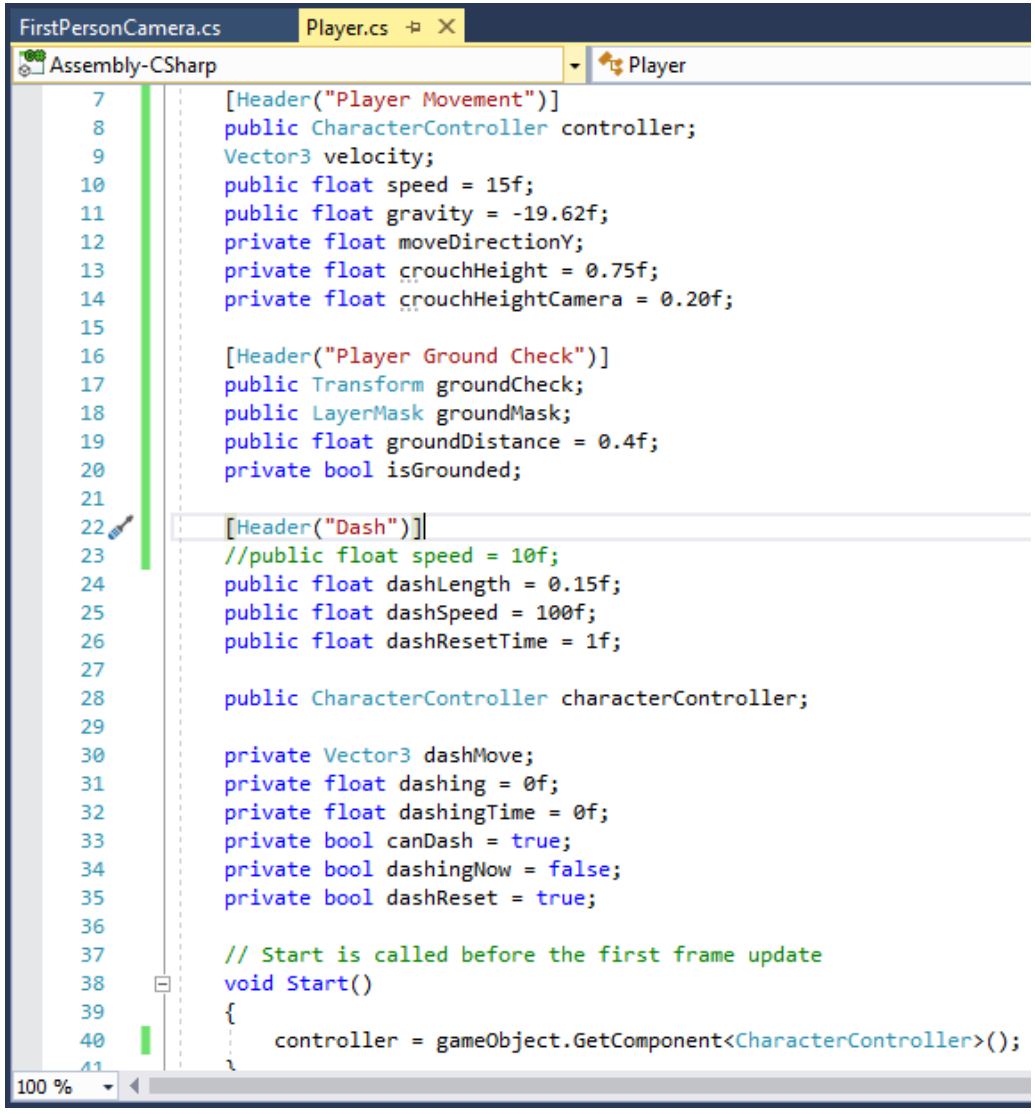
What originally my Unity and scene looked like, with what the inspector and hierarchy windows have.

FirstPersonCamera.cs | Player.cs

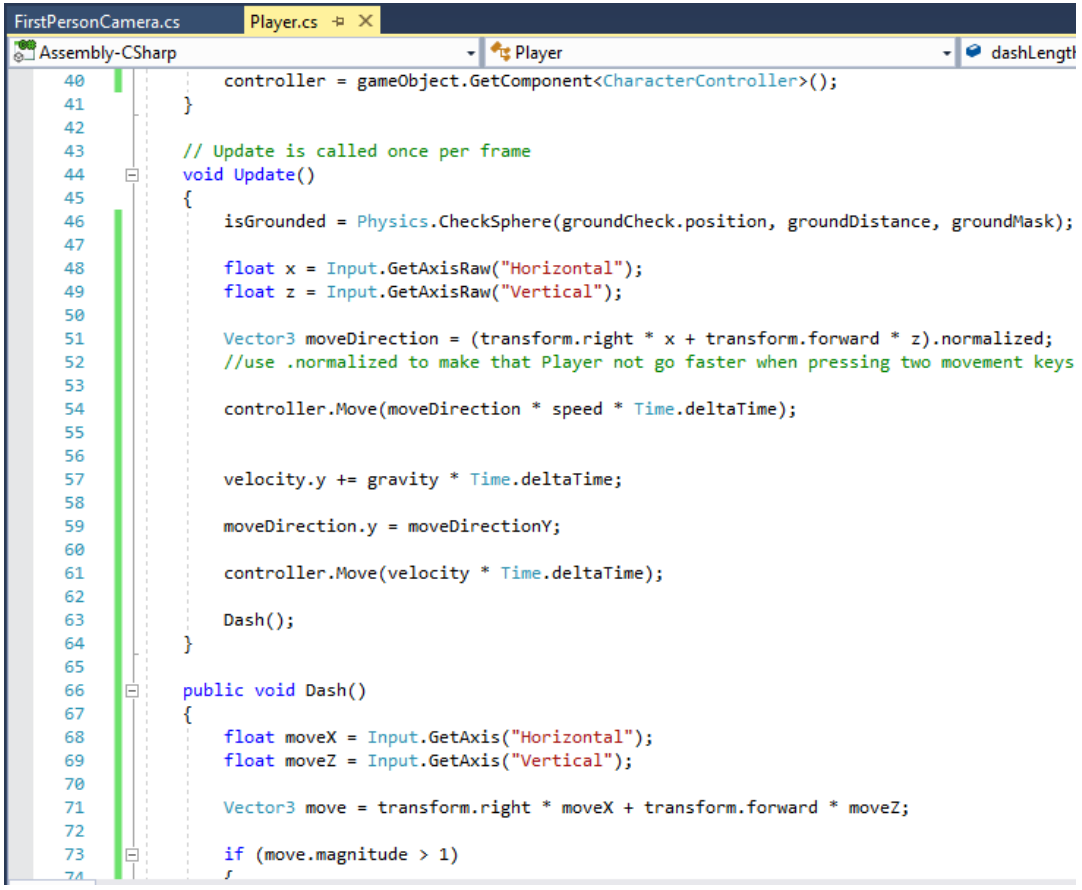
Assembly-CSharp | FirstPersonCamera

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class FirstPersonCamera : MonoBehaviour
6   {
7       [Header("First Person Camera")]
8       public float mouseSensitivity = 100f;
9       public Transform playerBody;
10      float xRotation = 0f;
11
12      // Start is called before the first frame update
13      void Start()
14      {
15          Cursor.lockState = CursorLockMode.Locked;
16      }
17
18      // Update is called once per frame
19      void LateUpdate()
20      {
21          float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
22          float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;
23
24          xRotation -= mouseY;
25          xRotation = Mathf.Clamp(xRotation, -90f, 90f);
26
27          transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
28          playerBody.Rotate(Vector3.up * mouseX);
29      }
30
31  }
32
```

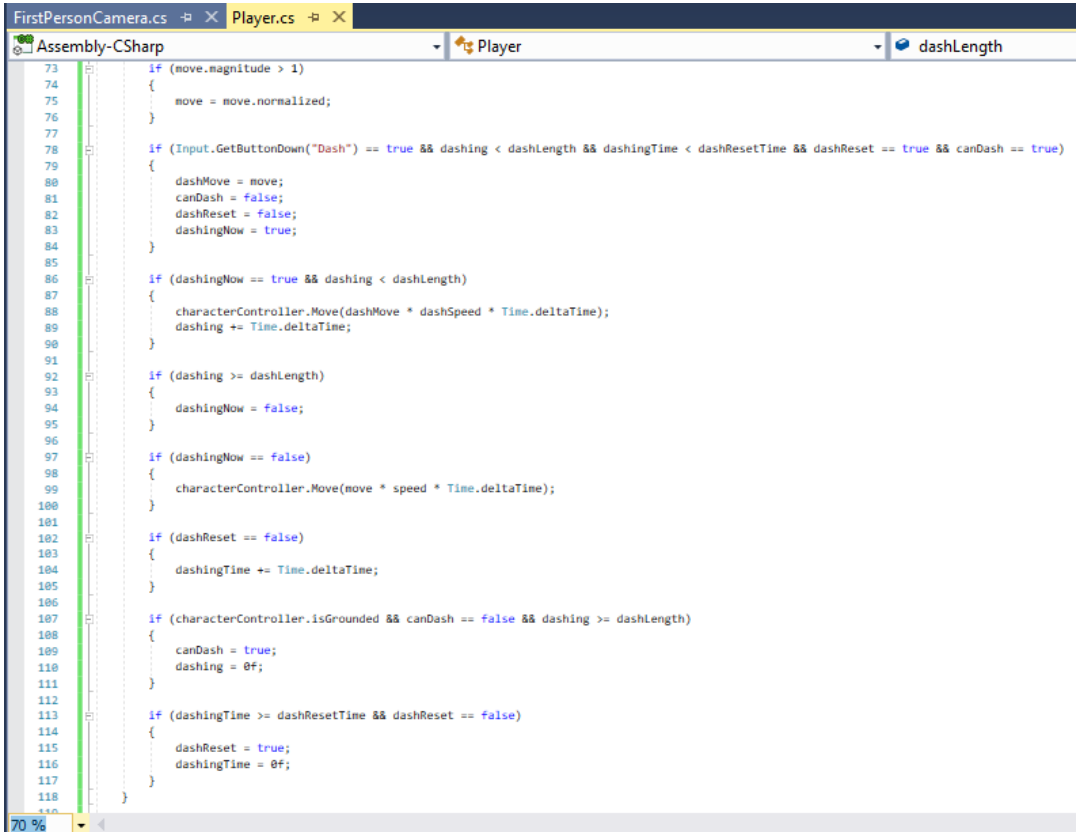
The code i copied from another project to this ghost game in the new C# script called "FirstPersonCamera".



The top part of the "Player" script, it has the code i copied from a different project.

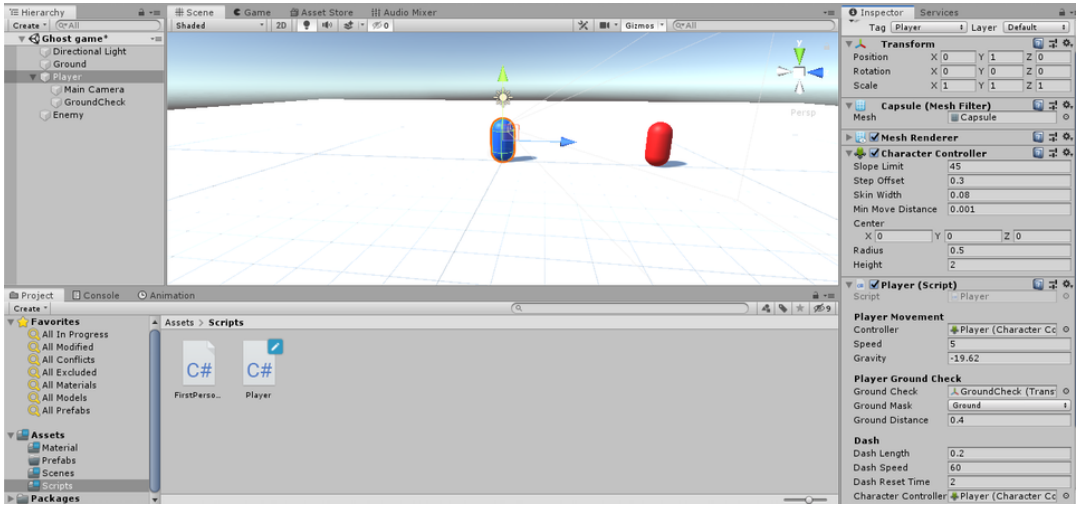


The middle part of the "Player" script, it has the code i copied from a different project.

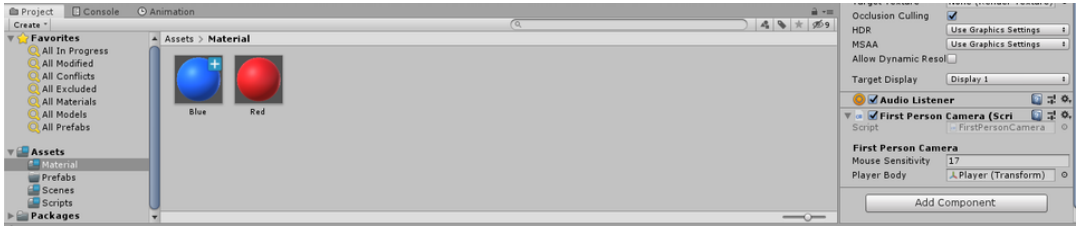



```
        if (move.magnitude > 1)
        {
            move = move.normalized;
        }

        if (Input.GetButtonDown("Dash") == true && dashing < dashLength && dashingTime < dashResetTime && dashReset == true && canDash == true)
        {
            dashMove = move;
            canDash = false;
            dashReset = false;
            dashingNow = true;
        }

        if (dashingNow == true && dashing < dashLength)
        {
            characterController.Move(dashMove * dashSpeed * Time.deltaTime);
            dashing += Time.deltaTime;
        }

        if (dashing >= dashLength)
        {
            dashingNow = false;
        }

        if (dashingNow == false)
        {
            characterController.Move(move * speed * Time.deltaTime);
        }

        if (dashReset == false)
        {
            dashingTime += Time.deltaTime;
        }

        if (characterController.isGrounded && canDash == false && dashing >= dashLength)
        {
            canDash = true;
            dashing = 0f;
        }

        if (dashingTime >= dashResetTime && dashReset == false)
        {
            dashReset = true;
            dashingTime = 0f;
        }
    }
```

The bottom part of the "Player" script, it has the code i copied from a different project.



The changes i made to the new scripts, the Player object and it's components of the Player script.



The new materials i added and the "FirstPersonCamera" script component with some changes.

The test i did to see how the Player moves and dashes.

29/11/2020 was the date that i started to create the new 3D FPS (first person shooter) that involves ghost and you are a ghost hunter scientist in a sort of abandon west town or something of the like.

I set up my new ghost unity project and added a simple plane, called "Ground", so my players and enemies don't fall and interact with each other. I also made three new scripts one for the Player, the camera and the other for the Enemy.

I added code to then and tested it out, the camera wasn't working so well, i could only look up and down. Not only that but i couldn't make the Player move at all, not even dash.

The code i added to the camera and player script were code that i made thanks to some YouTube videos that i then added to a personal project i was doing over the summer holidays. I have made multiple scripts but the main ones i want to copy from to add to this ghost project were the "MouseLook", "PlayerMovement" and "TracerDash", the name itself says what it's design to do.

The "MouseLook" makes so the Player can look in a first person fashion, "PlayerMovement" makes the Player move, jump and crouch using the character controller. Although i won't have any jump or crouch mechanic in the Ghost Town project, just some simple movement.
The "TracerDash" makes the Player dash like Tracer from Overwatch but a bit more simple and just accelerates the Player instead of a teleportation.

After doing all of that, i saved my game and exited it. But when i came back later and tried to open it from the files i have, the save file, it asked me if i wanted to upgrade it and i had to because then i can't open my game properly with accepting it. So i did that and then my unity looked a bit different which isn't my cup of tea, but what i really didn't like was that i could open my scripts, not even new ones that i created, how can i make a game if i can't make code in scripts. This still happened even when i open my game through the Unity hub.

I had no other choice but to make a new game which will basically have the same name. I then deleted the old ghost game and started working on this one, luckily i didn't do much so not a lot of progress was lost.

I added the 3D GameObjects and the scripts for my Ghost Town game, after doing that i added the code to the scripts.

Here is the "FirstPersonCamera" C# script i created for the main camera. This script will make it possible for the Player to be able to look around in a 3D first person perspective. The mouse will disappear, the Player will also not be able to look beyond vertically 90 degrees up and down.

But even after adding the code and dragging the script to the main camera the camera was still not working properly, so to figure the problem and find a solution i went back and watched a video i used to know how to make a first person camera, the video was "FIRST PERSON MOVEMENT in Unity - FPS Controller" by Brackeys.

After re-watching the video, i later learned that i needed to make the main camera the child of the Player. Then, after i did that everything works now, the camera moves normally now, the only slight thing is that the sensitivity is a bit high but i can adjust that to what is the best.

Later i went to my "PlayerMovement" and "TreacerDash" scripts from my other project and added it to my "Player" script in this project. Since all i need to do is copy and paste the code into this script i think it will be easy to do, i just need to remove some unnecessary lines of code and make some adjustments.

I won't be needing the jump or crouch mechanics so i removed them. After making sure the code is nice and clean, i tested it out and everything seems okay. The Player moves at a speed i like, the dash is fine, if i don't like how my player moves or dashes i can always adjust them later or even ask my peers for feedback and what they think is the best for the Player.

The image to the left is basically the top part of the "Player" script, with the variables and such.

Here is the middle of "Player" script that encompasses how the Player moves.

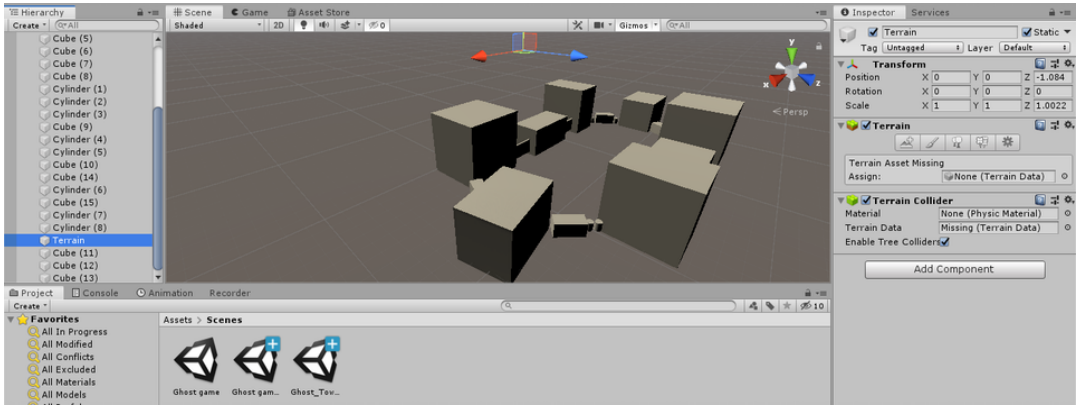
Here is the bottom part of the script that is inside the Dash() method, this shows you how the Player dashes, the time before it can dash, the speed, the length and more involving the dash.

I later went to the Unity editor and made some changes here and there, mostly to the Player and Camera in the inspector window. I also made sure that the Player had the Main Camera as a child and also a GroundCheck, it may not be important but it may be useful in case the Player some how goes above the ground, with out the GroundCheck and some GroundCheck code the Player would just stay at the air and not drop down.

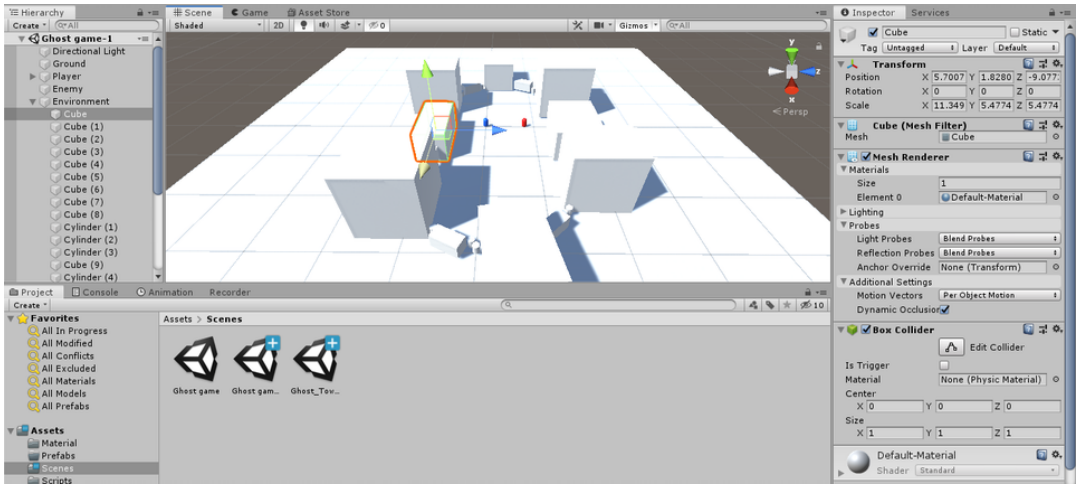
I also made some simple materials so i would know who is the Player and who is the Enemy.

After everything was done i made one final test and i even downloaded the Unity Recorder so i can record my game to show how my Player moves and the likes. The Player moves and dashes the way i like it, it probably could be improved but this is just fine the way it is.

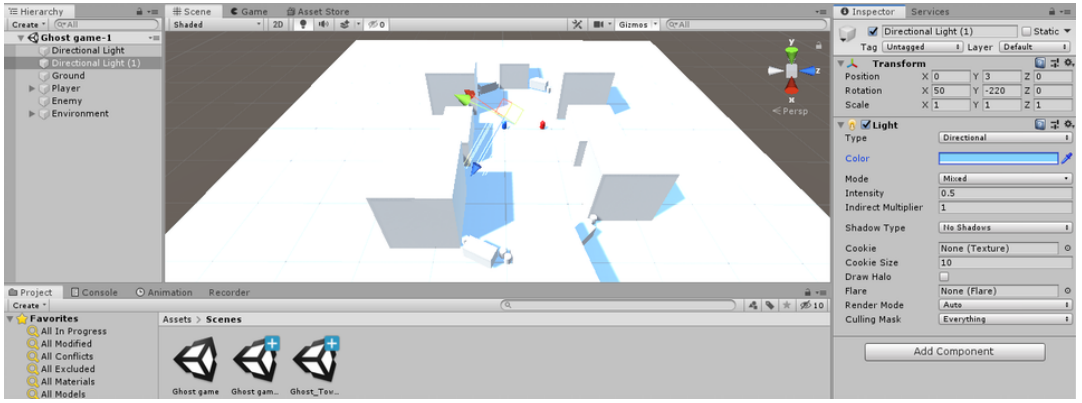
The enemy (capsule in red) doesn't do anything yep, it's just there as a referent point, to see how fast the Player moves.



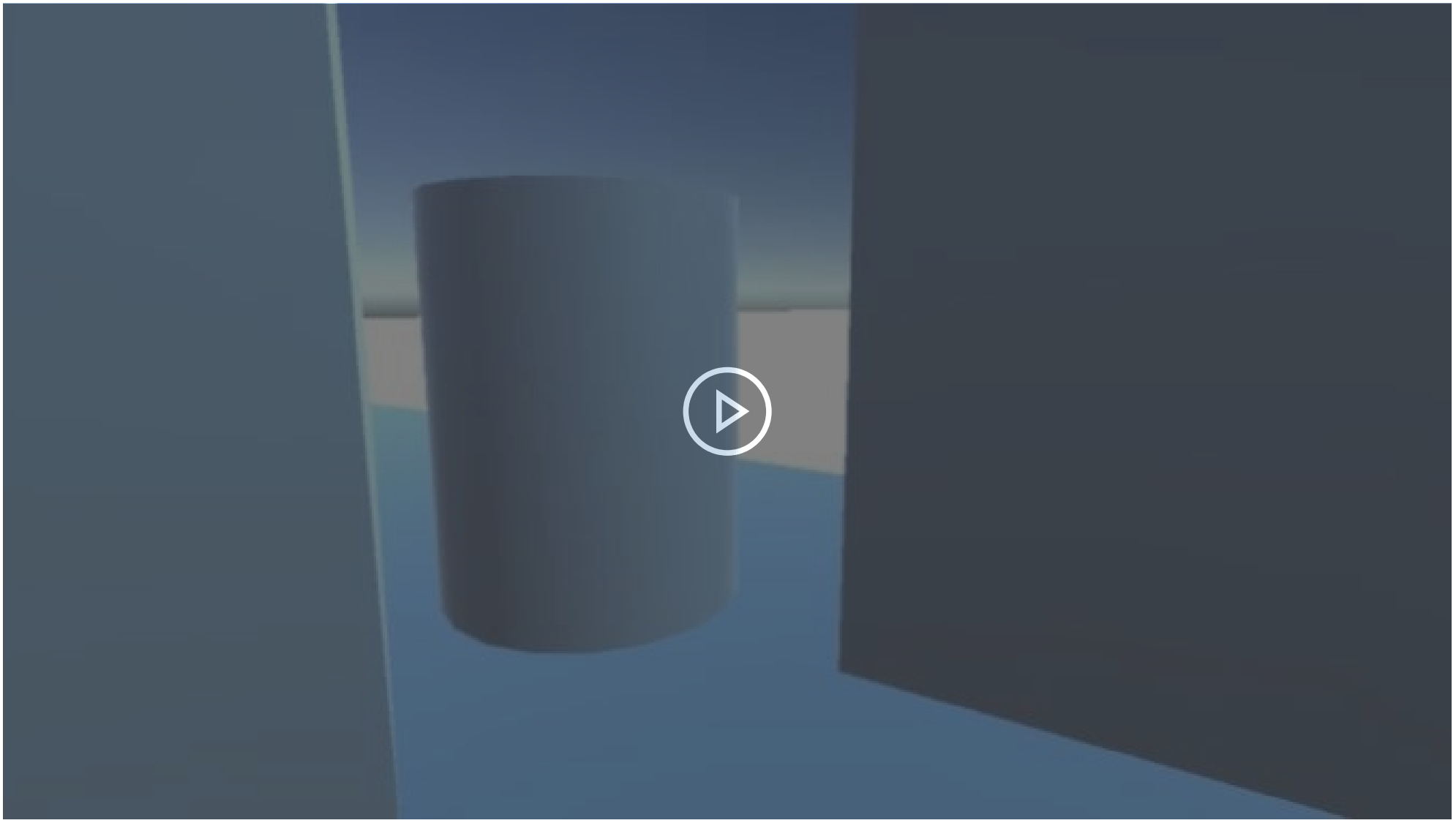
The blockout and terrain of the Ghost Town one of my peers made with some issues.



The blockout and terrain i copied and pasted to my Unity game.



The changes i made to the lighting to make my game look a bit better.



The test i did to see how the Player moves around the environment and how everything looks.

At 02/12/2020 i went to our groups discord and downloaded the file that one of the 3D modeler peer did, it's a blockout of the suppose Ghost Town map. But when i downloaded the file and dragged it to my Unity game, it appeared as a scene and when i opened it, the 3D objects where there but there was no terrain, which i found to be weird. I looked at the inspector of the terrain and it seems that there is no material or terrain data, i also don't know how to add some, there must have been some problem when the peer artist uploaded the map.

But even if the terrain wasn't working properly, i could still use the cubes, cylinders and all, and i could add them to my plane ground and test the player out and see how everything looks and feels. To do this i had to basically copy and paste everything into my scene. I also added a empty object called "Environment" and place all of the cubes, cylinders and all into it, in order for my hierarchy to be organize and clean.

There was one thing i noticed when i added the block to my scene was that the colour was different, i don't know why this happen but i could always change it later if i needed with materials or textures.

There was one thing i wanted to try and that was the rim lighting that i learned about last lesson with one of my tutors in 01/12/2020. I tried to copy the example and following the instructions. What i did was duplicate our Directional Light, rotate the duplicated one around the Y-Axis so it is facing the opposite direction of the original light. Change the colour to a soft blue, of the new light and then reduce the intensity to about half (0.5) and under Shadow Type, select No Shadows, in the inspector window.

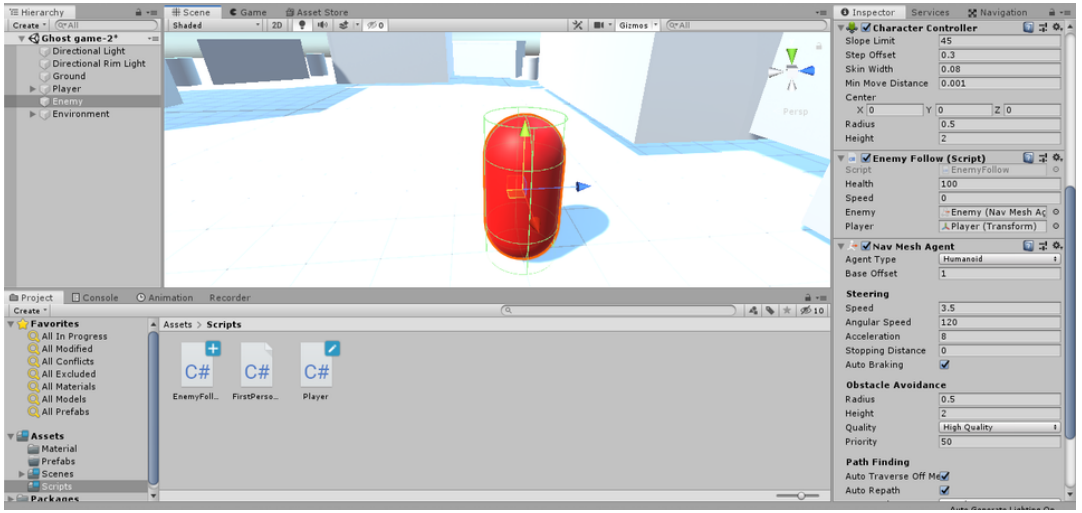
After i did all that, i looked at what i did and i think that it certainly looks better, but i don't know for sure since i'm not a artist and not very good in this area, but it's a improvement nonetheless.

There was also another thing i could have done to improve the lighting in my game and that was changing the Colour Space Rendering to Linear Rendering. This will allow for more accurate colour and lighting but i can't use this, mainly because WebGL and other browser export don't support Linear Rendering, that's why i will use Gamma instead of Linear Rendering.

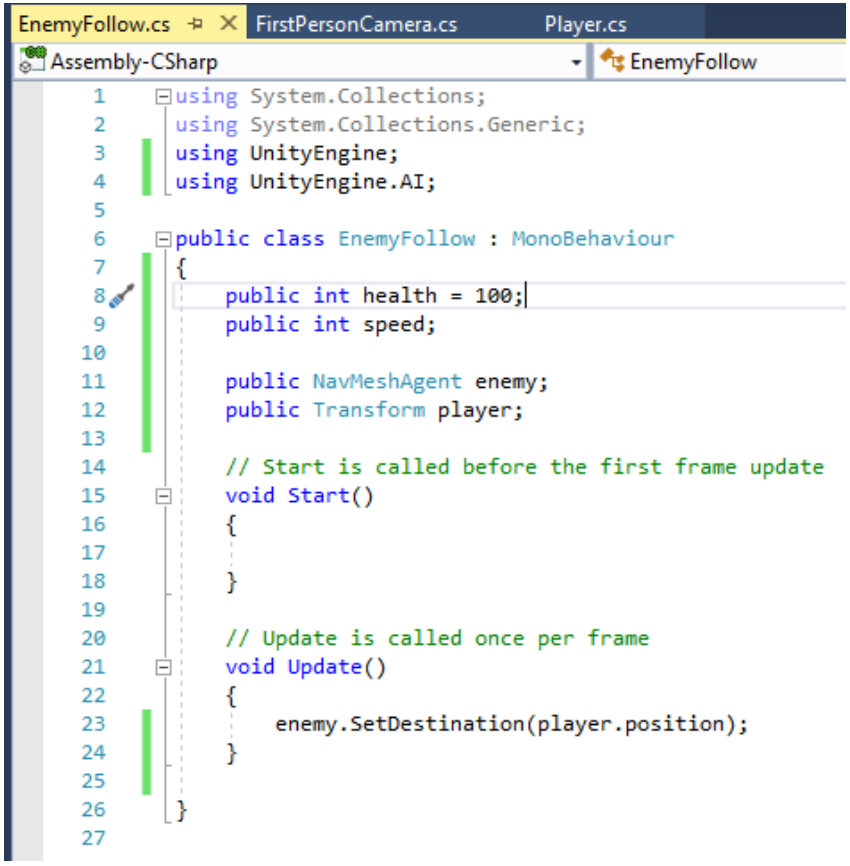
Then i went to play test my game to see how everything looks from the Players perspective while moving. I made the Player move around the area and see how the lighting affected the area, and after looking i can safely say that everything looks much better. The blockout itself

is a bit constraining for the player's movement in my opinion, and maybe objects are a bit too big but i can change all that in the editor.

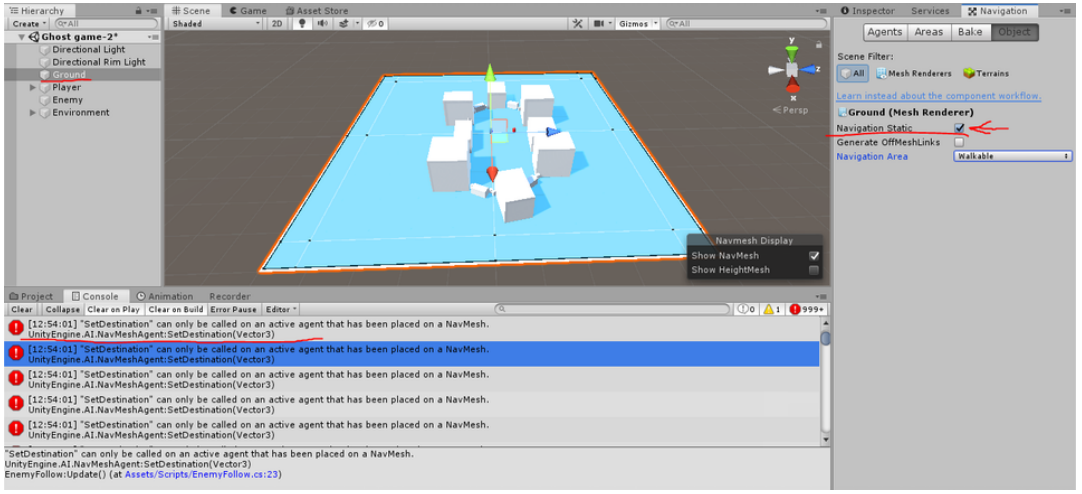
Overall everything is going smoothly, although i am now a bit worried about if the models, texture, environment and such made by my artist peers are gonna work in my Unity scene since my game has all of the code and such.



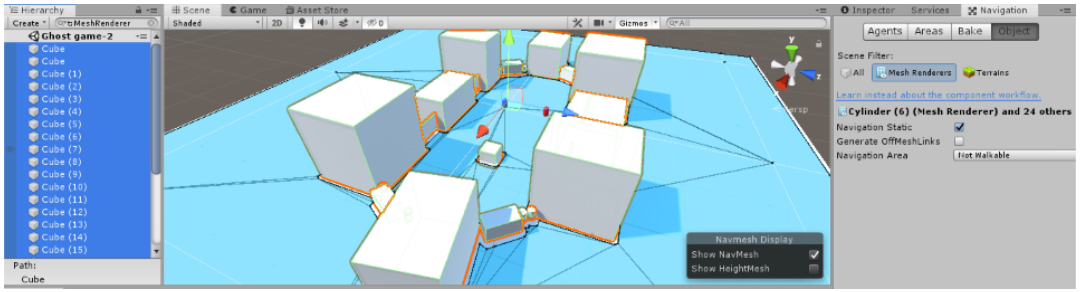
The Nav Mesh Agent i added to my Enemy.



The code i added to the new C# script called "EnemyFollow"



The problem of the enemy not moving and how i fixed that.



The additional changes i made to the Nav Mesh so the Enemy doesn't go through objects.

The test i did to see how the enemy moves and follows around the map.

On 06/12/2020 i started to watch a video called " How to make an enemy follow player - Unity NavMesh" on YouTube, this video will help me know how to make a enemy follow the Player using the Nav Mesh tool in Unity.

After watching the video i started following the instructions, what i did first was go to Unity's Window > AI > NavMesh, i also made sure that my ground was selected in order for it to be the only area the enemy can move on. Then i made some changes to the baked agent size and other stats, then baked so the stats are saved. I also made a new C# script called "EnemyFollow" so that the enemy can follow the Player.

Then after that, i added a Nav Mesh Agent to my enemy object and made some adjustments to it's stats. then i added the "EnemyFollow" script to it after adding code to it. I had to drag the enemy and Player objects to their respective slot in the Enemy Follow (Script) component.

Here is the code i added to the "EnemyFollow" script, i added the health integer because the Player will shoot the enemies and have their health go down, and once down to zero or below they die. The speed will make it so i can adjust the speed of the enemy and how fast they can move, but the Nav Mesh Agent already has a speed value so i don't know if i need the speed integer in the script, i'll find have to find out later.

The NavMeshAgent enemy is so that the NavMeshAgent component know who to move and the Transform player is to show the enemy who to follow.

Then i just added a small line of code in the void Update so that the enemy actually goes towards the Player.

Overall i didn't have to code a lot of things and it was very easy to do all of this.

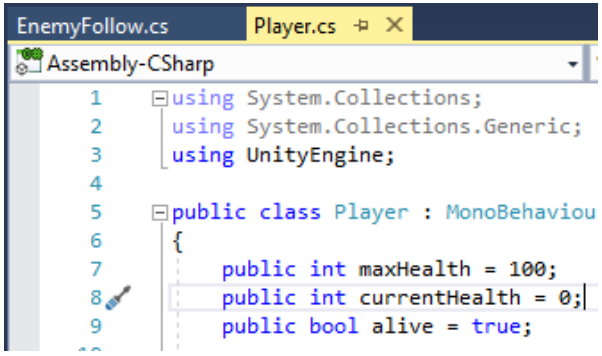
After doing everything i tried to test the enemy but it wouldn't move, after i exited play mode there was a error on the console. In order to fix this problem i was having i went to my navigation, went to Object and clicked the Navigation Stats and made the Navigation Area Walkable.

After i did this the ground now has a blue light on top of it, just like the tutorial i was watching, i didn't have this before because the Navigation Stats weren't on, i didn't know this was important because the video i watched didn't show this part of the navigation. Now that i have the Ground blue and walkable the enemy can now follow the Player.

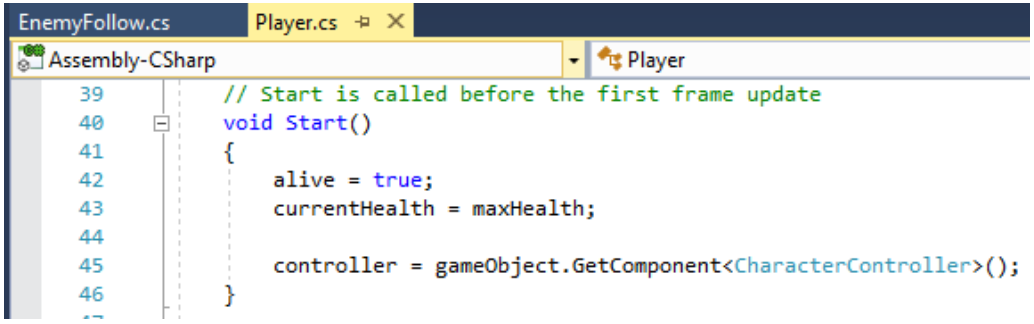
I wanted to see if make the enemy go around objects, i added a cube to the scene and selected all of the block and cylinder and made then non-walkable, then i baked my map and now my level map like this. Now the enemies go around the cube in order to follow the Player but i could make a enemy that also goes through non-walkable areas, making that specific enemy more like a ghost which is good variety.

Here is the final test i did to see how the enemy moves and follows the Player after all the changes i have made. Everything works well, the speed of the enemy is fine but i can always change that later. The enemy goes around objects, which is what i want. The enemy does go into the Player when it's close enough, maybe i can change it's collider or add some code so that it doesn't go inside the Player.

Overall i made good progress, in my opinion.



The code i removed and added to the Player script.



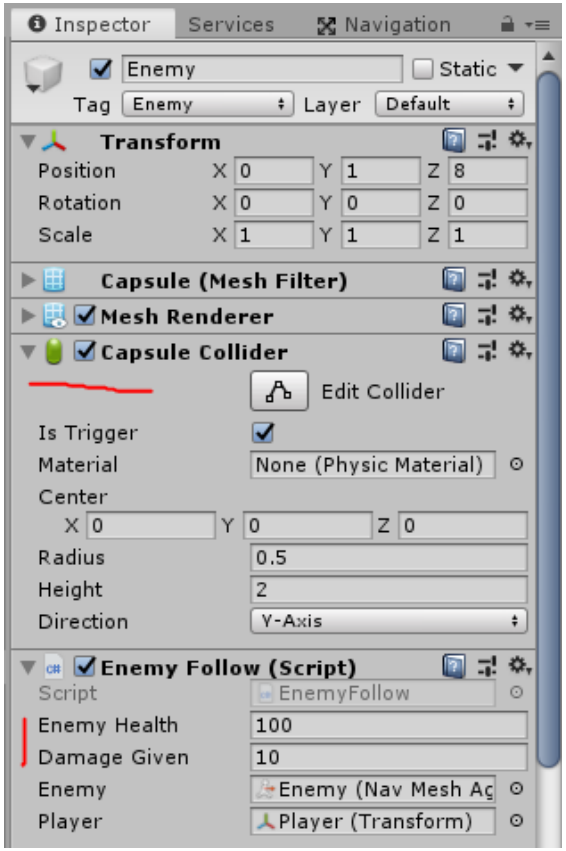
The code i added to the void Start in the Player script in the middle.

Player.cs

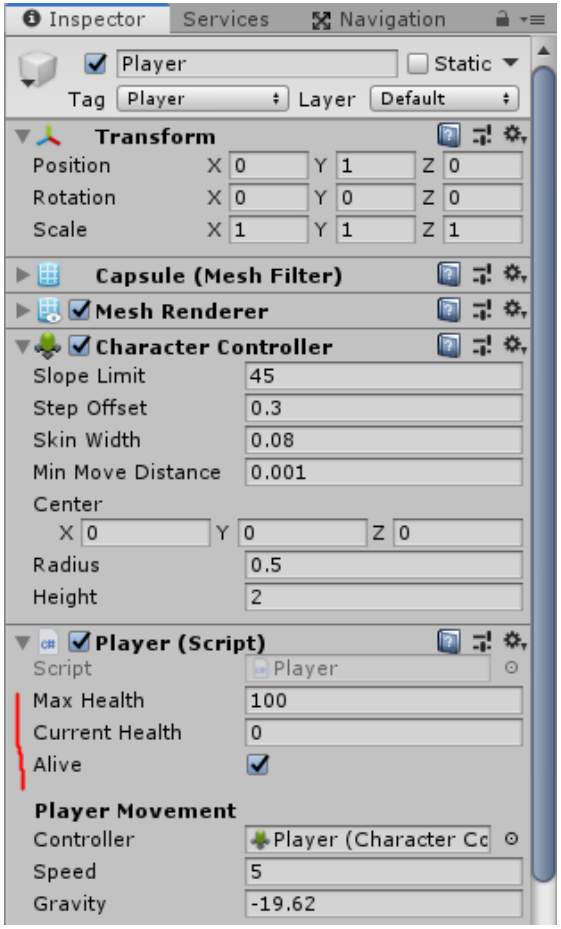
Miscellaneous Files

```
70
71      public void TakeDamage (int damageAmount)
72      {
73          if (!alive)
74          {
75              return;
76          }
77
78          if (currentHealth <= 0)
79          {
80              currentHealth = 0;
81              alive = false;
82              gameObject.SetActive(false);
83          }
84          currentHealth -= damageAmount;
85
86      }
87
```

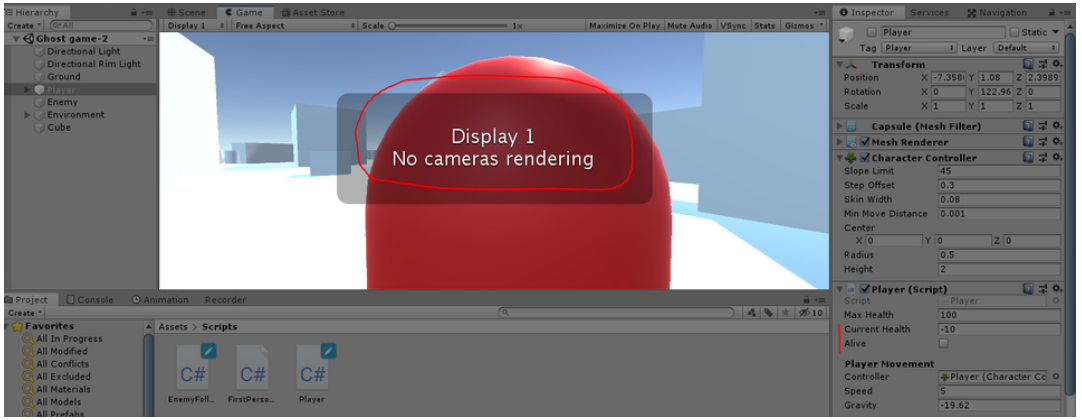
The TakeDamage method i created in order for the Player to take damage.



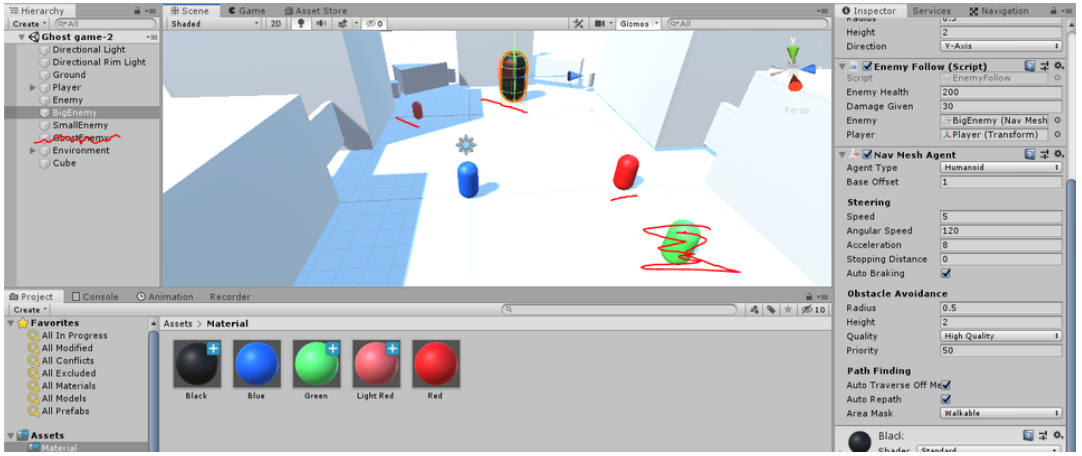
The changes i made to the Enemy gameObject, the components and the such.



The changes that happened in the Player (Script) component.



What would happen when the Player's health reaches below zero.



The new enemies i added to the game with some modifications and the green GhostEnemy i removed from Unity.

At 12/12/2020 i decided to start to make my enemy do damage to my player character and then later make multiple enemies with different sizes and stats.

I did a small research and found a short video showing you how to damage the Player's health, here is the video: (https://www.youtube.com/watch?v=g_3aHOaKwZY).

I started following the tutorial, adding code, making changes to my Unity game and all. I first made two small some variable in my Player script, to make the Player's health and the damage it's going to take when it comes into contact with an enemy.

Then i made a method that will make the Player lose health when it come into contact with an object that has the "Enemy" tag, which is my Enemy, then the Player will be destroyed when it's health reaches zero or below.

Unfortunately, when i tested this code out in game, nothing happened which is weird because i did everything right in the video i was watching. I thought that maybe because my Player doesn't have a Rigidbody, a proper capsule collider, or it's my Enemy that's the problem, i don't know. I tried everything, i made some changes to the code, nothing, i gave my Player a new collider, kept the old one then disabled it, gave it a Rigidbody, did that to the Enemy, i did everything and more but nothing worked.

So since nothing was working and i didn't want to waste more of my limited time i decided to research some more and i found a new video about the Player receiving damage, here is the video: (https://www.youtube.com/watch?v=gzXtfsezXWo).

I started watching then following the new video, i deleted the code i have done with the previous video. I went to my EnemyFollow script and i made it so the Player has health and the amount of damage the Enemy does to the Player. That and i created a VoidTriggerEnter that will actually make it so the Player receives damage when the Enemy touches the Player.

Then i went onto my Player C# script and added a few variable, a max and current health, as well a bool showing if the Player is alive or not.

Later i added the variable i made and added them to the void Start, the alive will always be true when you start or restart the game with the Player. The currentHealth will be the same as the MaxHealth in the start of the game but after that it will change when the Player receives damage.

After that i made a method called TakeDamage that will allow the Player to receive damage, lower it's health, be killed (disabled) depending on it's health.

Later after adding the code into the necessary scripts, i started making changes to my object in the Unity's editor. I first started doing some tests here and there to see what works, then i added a capsule collider and made it a Trigger so that the OnTriggerEnter works, then i made sure the values of the Enemy Follow (Script) are correct, i could also change them later if the enemy doesn't feel right according to me and my team, when it comes to gameplay.

Later i went to my Player and made sure it's stats were also correct, the Current Health is zero but it corrects itself when i start the game.

I did some testing to see if the collision works and it does, the health of the Player does go down by 10 when it enters contact with the Enemy.

While i was testing my game, i thought about changing the OnTriggerEnter into a OnCollisionEnter because i curious and wanted to see if the change was possible. I made the Enemy's Capsule not a Trigger so the collision with OnCollisionEnter works, then made some changes to the script and then tested everything out. For some reason when i tested this out my Player no longer was being damaged, the Player

was being touched by the Enemy, i 70% sure of it but for some reason nothing was working. I even did some adjustments in the code, to the Player and Enemy but nothing was working. After some tries and still not understanding what the problems was, i went back to what the code was before i tried experimenting with OnCollisionEnter.

When i went testing some more into my game and seeing the Enemy damage the Player, i noticed some things. The Player only really gets disabled when it's health is below zero, it should be that it gets disabled when it's zero or lower. Another thing is that when the Player is disabled the camera also gets disabled, the enemy also stops moving, probably because the Player no longer exists.

After all that i decided to make more enemies with some modifications between them. I made four new enemies, one big, slow but does a lot of damage to the player, another one that is small and fast. Then lastly i made made a another enemy that was like the original but does a little bit more damage but was suppose to have the ability to go through wall and objects. I also made additional materials to add to their corresponding enemies.

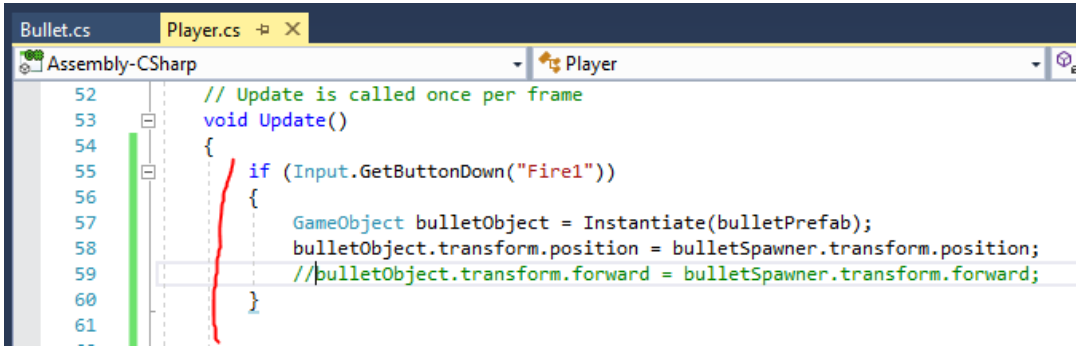
I made some changes to each enemy so that they are slightly different to the original red Enemy, they will have a different size, damage output, different speed and different points, when i add a point system in the future.

But as you can see on the image to the left that i have scribbled out in red the GhostEnemy in the scene and hierarchy, meaning i have deleted that enemy. The reason for this is because the passive ability i wanted my GhostEnemy to have, involving the Nav Mesh, wasn't working. I wanted this enemy to be able to go through wall and objects, making it easier for it to chase the Player and act more like a ghost. I wanted to make this enemy still follow the Player using the Nav Mesh feature in Unity but simply ignore some obstacles, all using

the Nav Mesh. I wanted every enemy to be using the navigation mesh and not add more code to one enemy, even thought that might have helped or not i don't know for sure, i didn't want to complicate things i wanted everything to be simple and i thought that sticking with the only the Nav Mesh and trying to solve problems only with that. In the end nothing was working after everything i tried, i deleted the GhostEnemy and now i only three different types of enemy.

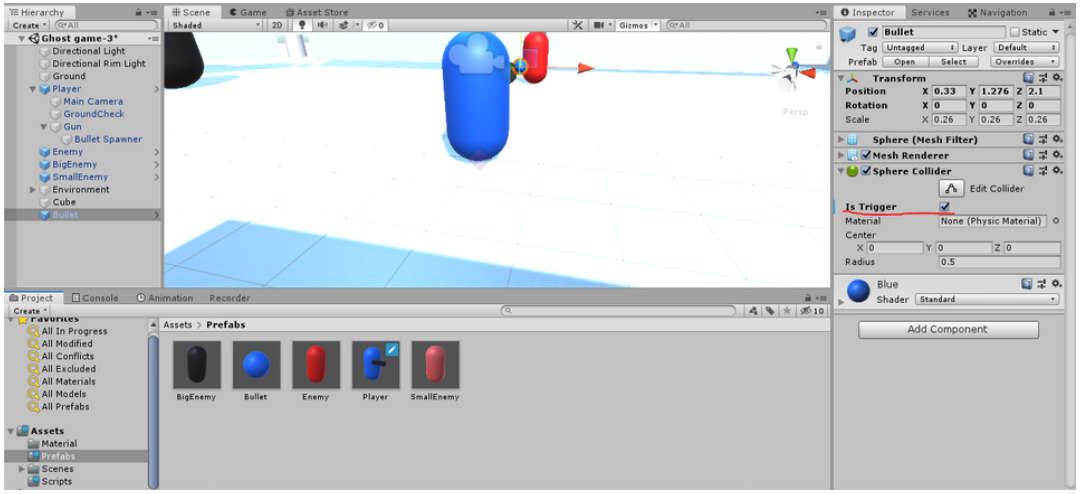
Later i, after doing all of that, went and tested out all of my enemies at once to see how they act when trying to follow the Player. All seems well, each enemy moves at their own speed and move around obstacles as intended, they also do their own damage to the Player, for example the BigEnemy does 30 damage to the Player and the Player does lose 30 health when in contact with the BigEnemy. The other enemies also work well and do their own damage.

Overall everything went well, the code works and my enemies follow and damage the Player. The thing i had trouble was the initial code i tried and then removed, and the the fourth new enemy i tried to create with a unique ability to pass through obstacles.



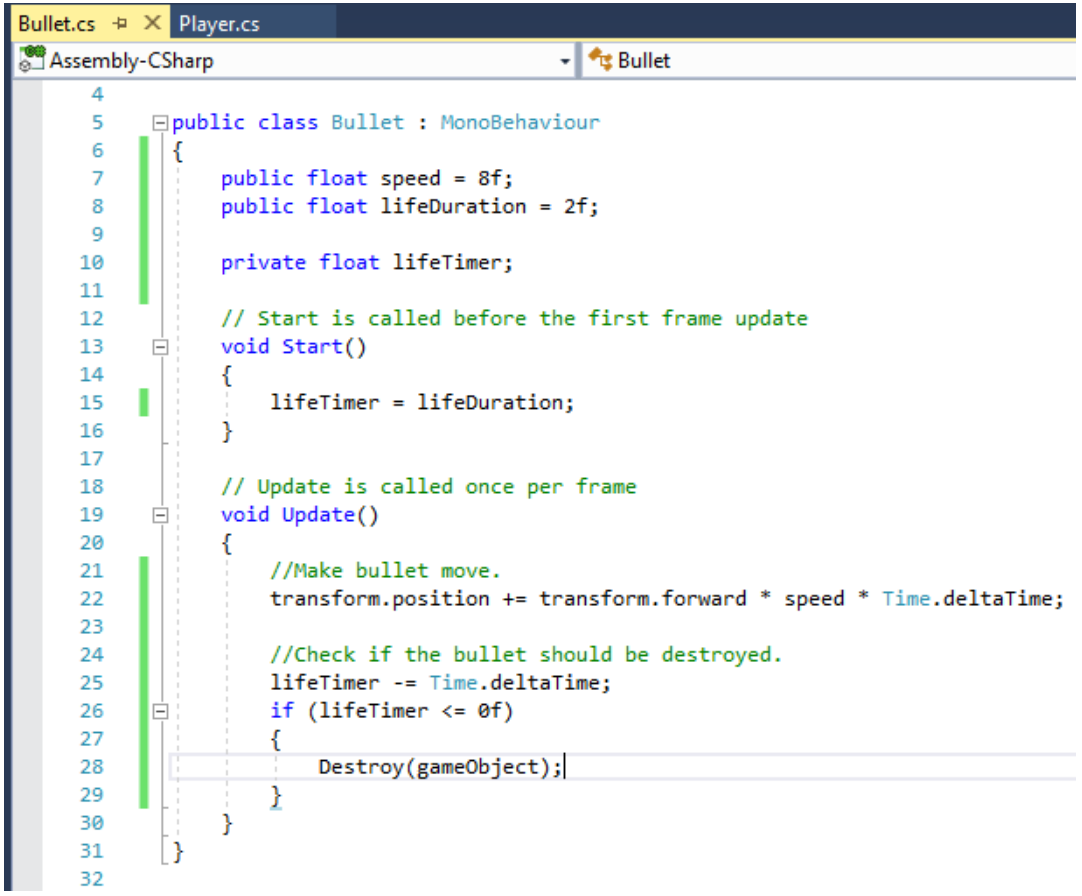
The new lines of code i added to the Player so that the Bullets would spawn.

The test i did to see if the bullets did spawn when i pressed the fire button.

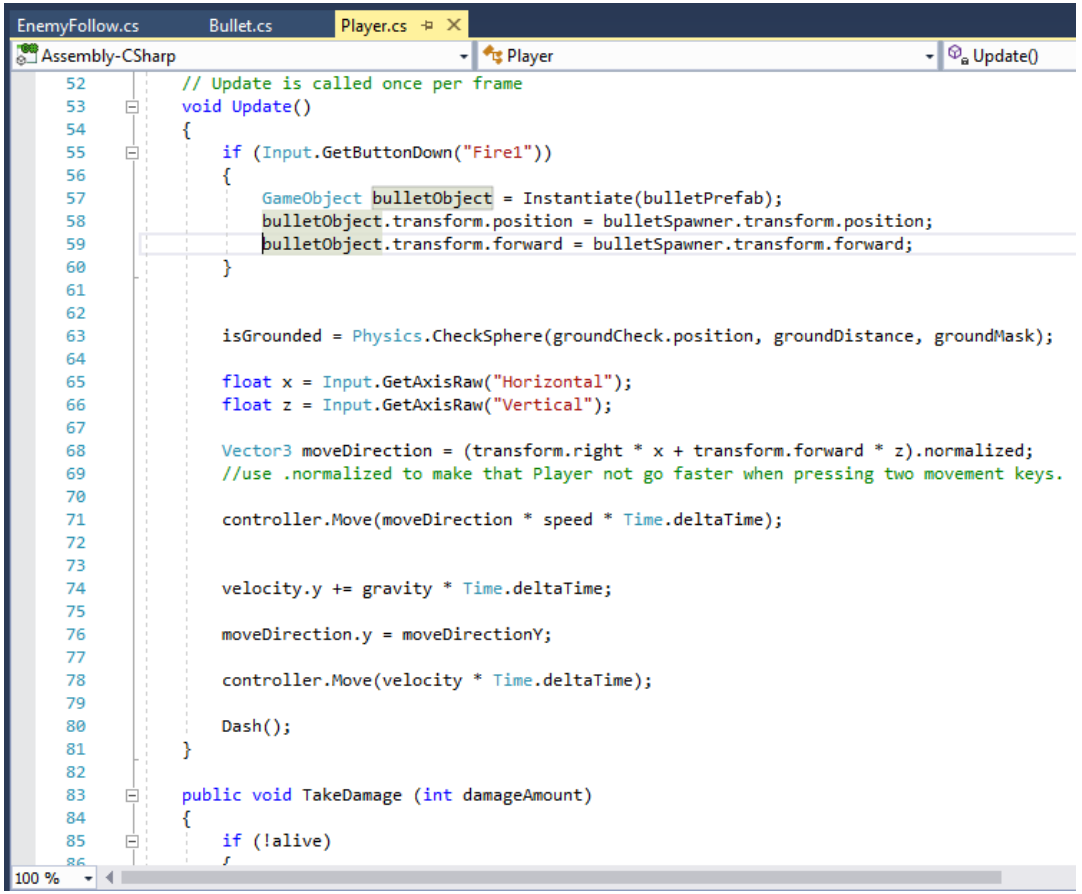


The change i made to the Bullet prefab so that the bullet goes through objects, instead of stopping t...

Player's movement.



The code i added to the C# script called "Bullet".



The changes i made to the Player C# script so that the bullets go to the direction the camera is facing and not just one point.

On the 05/01/2021 i have finished adding all i needed for my Player to be able to shoot a bullet, make the bullet move forward, make it damage the enemies and destroy it. What i basically did at the start was make a gun placeholder so i know where the gun is coming from and then i created a blue sphere that will act as my bullet. After that i made the Player, Bullet and all of my enemies into Prefabs, this will be useful if i want to make multiple copies of them via code in the future.

Then i made a little test to see how the gun looks in first person, it look good, i could move the gun a little bit closer to the center of the camera. But there was a little problem i faced, when every you look up or down, the gun wouldn't follow, it only followed left and right. I looked more closely into my hierarchy window of my Player to see what i did wrong and after some changes i found a solution. I had moved the gun to be a child of the Main Camera, with that the gun now fully works and moves as intended, this makes it look much better.

After i solved that small issue i started watching some videos on YouTube about how to make the Player shoot and damage enemies. After watching the ones i liked and were useful i started making code into my game for the shooting mechanics to work. First i went to my to my Player script and created tow new variables that would be "bulletSpawner" and "bulletPrefab", then in the inspector window of the Player i dragged the Bullet prefab to it's designated slot and also the Bullet Spawner, which i created and placed slightly in front of the gun, to it's slot in the inspector.

Later i added more lines of code that will make it possible for the Player to spawn bullets when the Fire1 button is pressed. I also commented out the last line of code as i believed that it wasn't necessary since the tutorial i watched didn't have a bullet spawner and was shooting from the center of it's camera.

I then did a test to see if the bullets were spawning properly and thankfully they were but i did notice something, and that was that the bullets could collide with the Player preventing it's movement. This may not seem important because when the bullets move it won't stand still and affect the Player. But i still wanted to solve this issue because i believe it would improve the bullets that way.

In order to fix the issue of the bullets colliding with the Player i decided to make the bullets collider a Trigger. I then did a test and now the bullets go right pass the Player, so i believe that solution was a success.

After that I started adding code to my C# "Bullet" script which i created beforehand. I used one of the tutorials i watched to add the code. In this script is the code that will determine how long the the bullets live for, how it moves forward and how fast. Also how the bullet should be destroyed.

After that i did some tests and while the bullets were moving forward, they were all pointed at one specific point, no matter where the player was.

I looked back at the my Player script and decided to have on of the lines of code not a comment anymore, then i tested my game. Now everything works perfectly now, the bullets go into the direction i want.

When i was testing my shooting mechanics and seeing how it was, how the speed was and such. But then i noticed something that bother me, when you fire the bullets and move around the bullets seem to move like it has drag in it. What i mean is that when when the Player

moves forward, the bullets i shoot spawn more back then the original bullet spawner is, basically inside the gun. If the Player moves backwards then the bullets spawn more forwards, it looks very weird as you can see in the video to the left, i want to solve this small issue and i even did some research on similar problems and how to fix it. Here is a site i found that may be useful to me: (https://gamedev.stackexchange.com/questions/162854/projectile-not-correctly-inheriting-parent-velocity-unity3d)

Later i wanted to make it so my bullets could damage and destroy my enemies. After some research i found a way to do that, i then updated my "EnemyFollow" script accordingly. I had to create a Bullet tag so that my enemies know what is damaging them, i also had to adjust my previous OnTriggerEnter that was affecting the Player health. I also learned that you technically can have more then one OnTriggerEnter that are the same, even if i the (Collider collision) is different like (Collider other) for the other one it still counts as the same, so all of the code that involves Triggers are in the same OnTriggerEnter.

Later i made some more adjustments here and there, i added a Rigidbody to my blue bullets, changed a few values of speed of the bullet, adjusted the damage the enemies take when the bullet touches them and much more, all to make the 3D ghost FPS game feel better to

play.

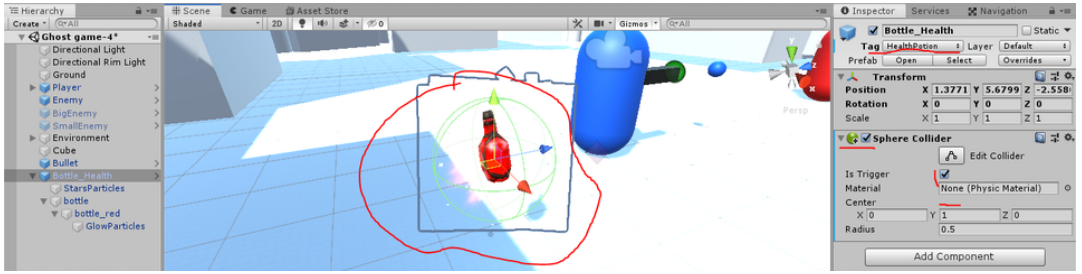
I then did a final test to see how everything feels and plays, i'm happy to say that everything feels good, the movement, the shooting and everything is good, although there could be some improvements. Overall i did take a good break throughout the Christmas holidays, i just wish i did a few more work on my project. I still have quite a few things to make before i'm happy with the progress i'm making, i will move onto to make a explosion barrel, a health item for the Player and then a system to spawn enemies around the Player.

Here are also the YouTube videos i watched that helped me make a shooting mechanics and damage my enemies in my game:

-(https://www.youtube.com/watch?v=9kU1vRKkaF0);

-(https://www.youtube.com/watch?v=MKjEOVvU6ug);

-(https://www.youtube.com/watch?v=Pm3sczeq3ks).



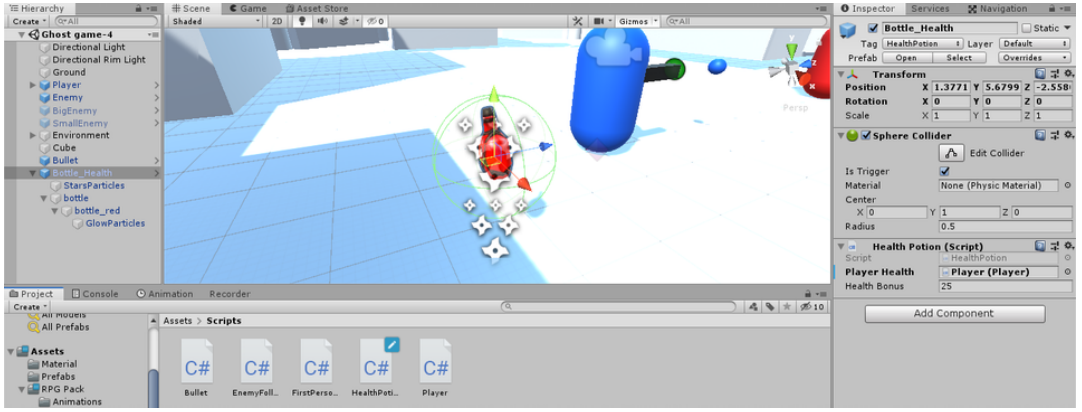
The changes i made to the health potion item, the collider, the Trigger and more.

HealthPotion.cs | Player.cs

Assembly-CSharp | HealthPotion

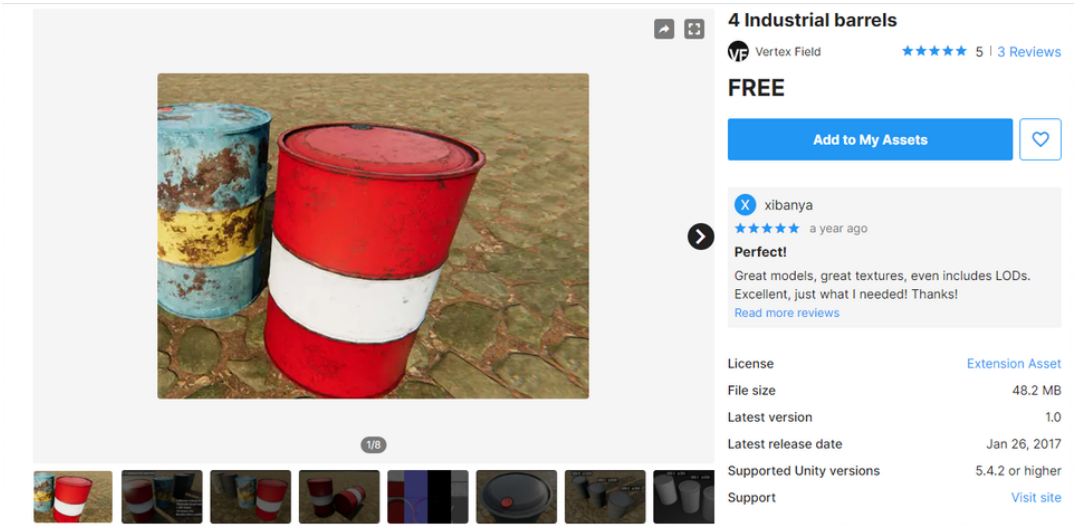
```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class HealthPotion : MonoBehaviour
6   {
7       public Player playerHealth;
8
9       public int healthBonus = 25;
10
11      private void OnTriggerEnter(Collider other)
12      {
13          if (playerHealth.currentHealth < playerHealth.maxHealth)
14          {
15              playerHealth.currentHealth = playerHealth.currentHealth + healthBonus;
16              Destroy(gameObject);
17          }
18      }
19
20  }
21
```

The new C# script i created called "HealthPotion" that will make the Player heal themselves.

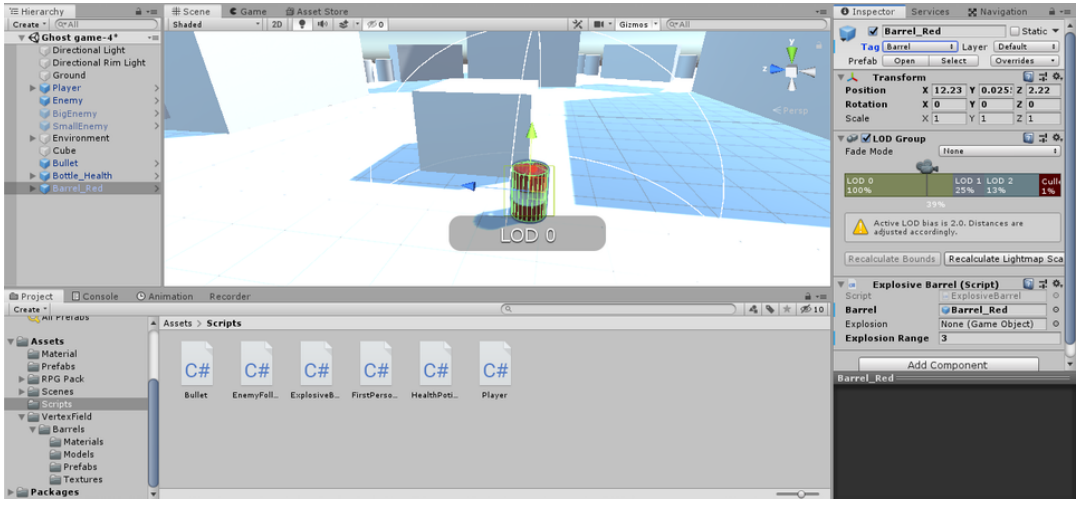


The new script i made then dragged to the health potion item and dragged the Player to it's slot in th... inspector of the health potion.

The test i did to see if the Player would take the potion and heal themselves after taking damage.

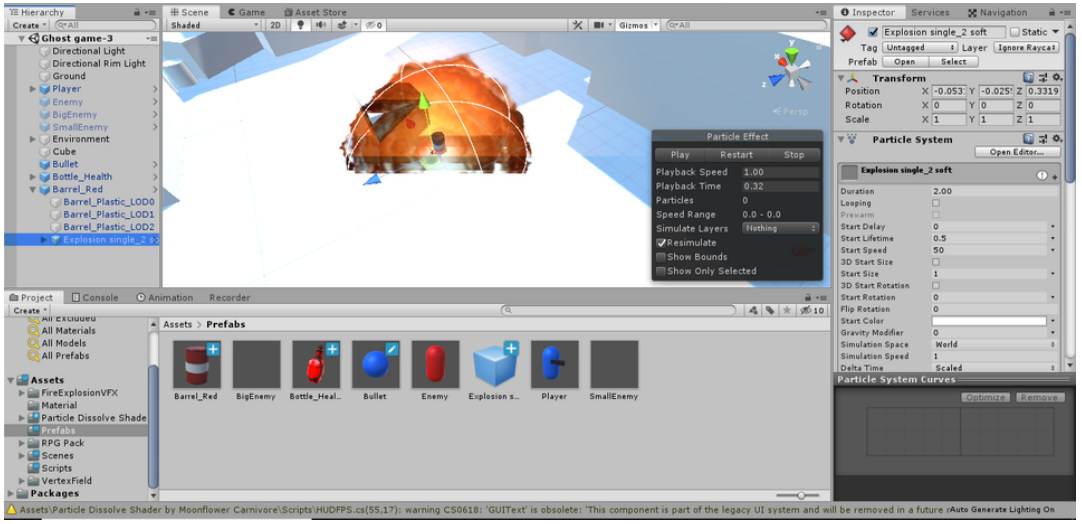


Where i got my red barrel object.



The new scripts i created and added to the new red barrel which i placed in the scene.

The additional changes i made to the "ExplosiveBarrel" script.



The additional assets i downloaded when i was having trouble with explosions not working, the changes i made to the barrel, the explosion itself and among other things.

The final test after i have fixed everything to see if the explosion appears after you shoot the barrel and if it damages the enemies.

By 14/01/20201 i have finally completed the two objects i wanted to add to my game. I started to look for items that can look like health items that would heal the Player, in a obvious way so people can know what to do in case they need health. The reason i want health items in game is so that players can play my game for much longer and and heal themselves if they make a mistake. So skilled player can have a high score if they tried and used the potions wisely.

After i found a item i like in the asset store, i downloaded it and install it to my unity game. Then i chose the red potion Prefab and dragged it to the scene, and to my surprise it has everything i want. It has animation, a glow effect and particle effect, i just wanted a 3D model of a potion so i could do the animation but this as done the work for me and more. This will definitely save me a ton of time.

After seeing how the health potion looks like, it was time to make changes to is so that it can affect the player and vice-versa. I made a new tag called HealthPotion, i don't know if i needed that for a tutorial i was going to watch but i thought to my self that might as well be prepared and do it now. I also gave the potion a sphere collider and made it a Trigger so it can interact with the Player.

I then created a new C# script called "HealthPotion" that would make it so the health item could heal the Player, I watched a video called "Health Pickups in Unity" on YouTube, even if the video is 2D i could simply change the things that are 2D to 3D and it's easy to do.

The script will have a value of how much health the Player gets once it touches the red bottle. Then the bottle will destroy itself, making it disappear. But to know how the health item know who to heal it needs a Player gameObject which i can drag and place it into it's slot on the inspector in it's component.

After what i did i made sure all the component script had the right value and gameObject in it before i tested it in my game.

I tested the health potion by having a enemy that would damage me and then i would go to the health item to see if anything worked. First i went to the item without getting damaged to see if the item would still disappear if i touched it and thankfully it didn't. Then i got damaged by the enemy, and touched the health potion and i looked at my Players health and it did heal me by 25 points of value. Everything seems to be working fine, but i did discover a small issue, when the Player takes a little bit of damage, like 90 out of 100 and takes the health item, it will heal the Player and then it's health will be 115/100. Since the Player doesn't have a health limit the additional health can be higher then 100. I won't be fixing this small issue because i don't believe it's that important to fix and it's unlikely people will notice it.

Later i went to Unity's asset store to look for some 3D gameObject barrels that i can download and have in my game. But now after i did everything i now thing that what i did was unnecessary since one of my teammates is going to make a personal cartoonish barrel for our ghost FPS game. I don't really know why i didn't just add a cylinder and just added a red material to it to my scene. Maybe because i thought i needed it or maybe since it was so easy to add and use the health potion objects i downloaded i could do the same with this barrel i installed. I should really think if i need certain assets or not before installing it. But what's done is done, i'll be more careful and not install assets from site when i'll have the assets from my teammates.

Here is where i added the red barrel to my scene, made some adjustments to it, like it's position, i gave it a tag called "Barrel" even if it's not necessary, i still created one because i wanted to and it doesn't cost a lot to do. I also created a new C# script called "ExplosiveBarrel" wherein it is the code to make a explosion particle effect happen and be able to kill enemies in a radius.

I started watching a tutorial i thought would be great for my game, the video is called "MAKE EXPLOSIONS in Unity (Tutorial)". This tutorial will help me make is so that a explosion effect will happen when a bullet hits a red barrel, and then a if a enemy is near the explosion range of the sphere then they will be destroyed. That and some other stuff that the new C# script will do to the barrel and everything around it.

Here is the bottom part of the ExplosiveBarrel script since i could fit everything into one image. This part of the script will make the explosion happen when a object with the Bullet tag hits the barrel, and the the radius of the explosion range can be seen thanks to the OnDrawGizmo, this is helpful to show the range of effect of the explosion.

Then later in the tutorial it show something about the enemy and how the explosion can affect it, something to do with ragdolls and force. But my game won't have those types of things, i just want it to kill my enemies. I originally wanted my barrels to damage them but then i thought that can make things complicated so i simplify the code to just immediately destroy any enemies that touch the explosion.

In order for the explosion to work and everything i would need to download and install actual explosion particle effects, otherwise the barrel and enemies near it would simply disappear without any cool effects. SO i went to the Unity asset store and downloaded one free explosion particle i liked, then i installed it into my unity ghost game. Then i added it to may scene to see how it looks like and if i like it. It

looks good and so i drag the explosion prefab into the correct ExplosiveBarrel slot in the script component ,since it existed when i downloaded the asset, i then tested the barrel and for some reason nothing was working. When i shoot the barrel the barrel would simply disappear with no explosion effect, but if i drag the explosion prefab to the scene itself i could see it, modify it and such with no problems, i'm only having troubles spawning it in the barrel.

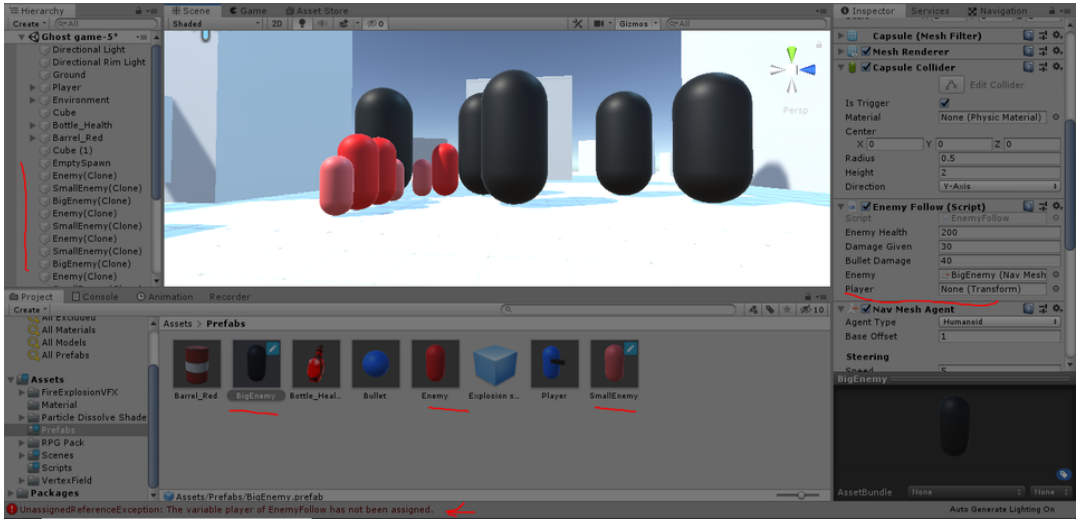
I think that maybe it the explosion itself that's the problem, so i download a different explosion asset and install it in the game, works fine in the scene alone but then i drag it to the slot of the ExplosiveBarrel script component, then i would test the barrel and the same thing happened, nothing. I think that maybe the connection with the bullet and barrel isn't happening, so i add a Debug.Log("explode"); to see if the trigger is being activated and it is after doing a simple test, so i now know it's not the trigger. I commented out the audioSource.Play(); because it kept giving me a error saying it was trying to find sound but i did have any, because i didn't add any to the game yet for the explosion sound, i will in the future maybe if i have time. I started googling solutions to why my explosion effect isn't working and i found a website saying that i could try to instantiate it, the website is this:(https://community.gamedev.tv/t/how-to-add-explosion-effect-after-the-player-gets-destroyed/85581/2).

I then add a line of code about Instantiating then i tested the barrel and still nothing was happening. I decided to look at the tutorial i watched for the explosion again, and i found out that maybe i would need to make the explosion into a child of the barrel in the Prefab. After i did that and some changes to the explosion particle, it's size, what particle comes out and such i tested it out and now after everything it finally works. I don't even exactly know how but it works and that is what is important. The changes i also made to the explosion makes it look like it's the perfect range of the sphere gizmo.

Her is what the explosion looks like, the prefabs i have, the changes i made to everything and the assets i downloaded when i was having trouble with the explosion.

After everything i finally did a final test to see if the red barrel made a explosion after i shoot it with my Bullet Prefab. I glad that after the test it has been revealed that the explosion does happen after you shoot it, the explosion it self is quite good in my opinion, a tiny bit large but overall good for the project, the animation is also good. For some reason the barrel needed multiple bullets to destroy it even though it just need one, maybe it's my bullets or maybe it's the collider of the barrel i'll have to check it out later and maybe make some small changes. Another thing i notice is that the enemy some times doesn't die after the explosion, i don't know if the enemy wasn't in the explosion radius, or maybe it was behind cover or something i'll have to look more into this issue if i want it to consistently kill enemies.

Everything i did was interesting to say the least. The health potion was quite easy and straight forward in my opinion and save me a lot of time, but the explosion inside of a red barrel was what i didn't like a lot, mainly because it took me a lot of time to fix just so i can have a simple particle effect of a explosion. But i am now grateful for the experience as i now know how to make a explosion work, the code, the inspector, value and other stuff. All of what i experienced will help me be a better programmer.



The test i did, the changes i needed to make to my enemies because of a problem, the spawning problem and more.

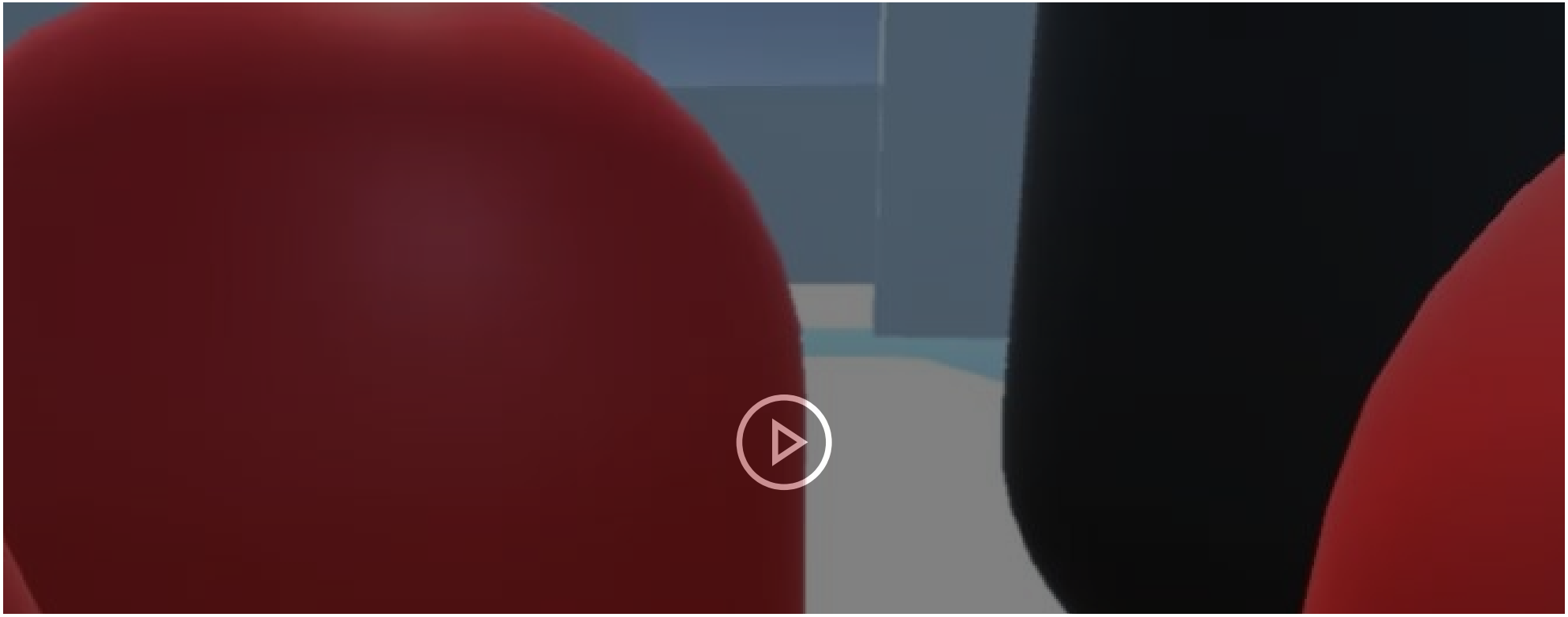
The test that i did to see if the enemies spawned correctly, but there were a few issues that appeared immediately.

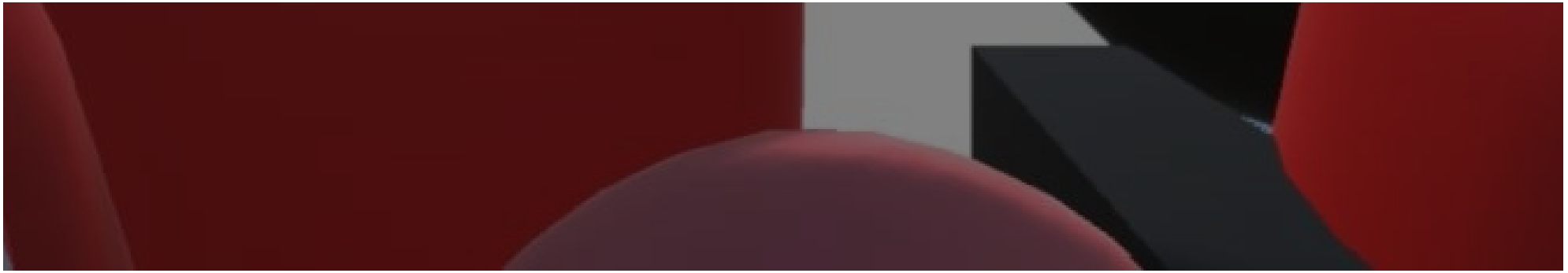
EnemyFollow.cs | SpawnEnemies.cs

Assembly-CSharp | EnemyFollow

```
6   public class EnemyFollow : MonoBehaviour
7   {
8       public int enemyHealth = 100;
9       public int damageGiven = 10;
10      public int bulletDamage = 40;
11
12      public NavMeshAgent enemy;
13      public Transform player;
14
15      // Start is called before the first frame update
16      void Start()
17      {
18          player = GameObject.Find("Player").transform;
19      }
20
```

The one additional line of code that fixed the problem with the enemies not properl... following the Player.





The final test on the spawning of the enemies, everything seems to work well.

On 19/01/2021 i finished making my enemies, of all types, spawn properly, following the Player, damaging the Player, being killed by bullets and dying with the explosive barrel.

But in order to achieve all of that i needed to look for some tutorials and videos on how to make this possible to the best of my abilities, after watching some videos and sites i found one video i liked and used it in order to make my enemies spawn in a certain area. I first started to make a new script called "SpawnEnemies" and i also made a green cube to help me figure out the area of spawning for the enemies. The video i used to help me is called "HOW TO RANDOMLY SPAWN ENEMY POSITIONS WITH C# UNITY TUTORIAL".

Then after i watched the video, i practically copied the video's code into my new C# script, with some modifications since my game is more complex and has more stuff in it. The changes i made will hopefully make sure that all of my enemy types can spawn one at a time not just one enemy always.

Hopefully i types the values correctly so that my enemies spawn in the correct area and that each enemy doesn't spawn below or above the ground.

After i made the changes in the script i went to my editor in Unity and went to test the code out, but there was a problem right away. The enemies did spawn but for some reason were not following the Player for some reason, when i look more closely into my enemies in the inspector and that's when i realized that for some reason the Player slot in the Enemy Follow component is empty when is should have the Player Prefab in it. So i fixed that onto all of my enemy Prefabs and then i tested the game, and now the enemies follow the Player or at least up until a point. When the enemies spawn they will go to a position the Player was at the start but then just stay at that position or try to get close to it, even if the player is some where else the Player can't do nothing to change the enemies position.

Another thing that is a problem is that not just one enemy is spawning, it is three at the same time of all types, i believe i kinda know what the problem is and that i might need to make some changes to the new script i created.

Here is the test i did and the problems i was facing, what it looks like and how. The good thing is that the spawn timer is working correctly, spawning enemies every 3 second, i just don't like that the enemies are all spawning all at the same time and not properly following the player.

I did some research and i found a web site of someone having problems with their enemy AI following the Player, and the problem the person was explaining seemed exactly like the problem i was having. A random person commented on a solution and what the person needed to do, the person asking for solution said that it worked, that means i have to try this out, because if it worked for them it must work for me. I tried the simple solution that is a simple one line of code to my EnemyFollow script, i tested this out and now my enemies follow my Player properly. All thanks to this website: "https://answers.unity.com/questions/1727685/i-need-help-with-enemy-following-my-player.html".

Later i researched more videos and sites on how to make my enemies spawn properly, spawn a random enemy type on every spawn and have the enemies spawn practically every where on the map instead on a square area.

I found a short video called "*Random Enemy Spawner - Unity 2D Tutorial (NEW)*" on YouTube, that had some differences compared to the other video i watched. This video uses arrays so that i can have multiple points that spawn the enemies and a array of enemy prefabs so that the spawning makes it so one random enemy appears on each spawn, instead of all of my three enemy types.

I had to make some changes to my SpawnEnemies script in order for things to work properly, as you can see in the image to the left of the top part of the script the changes i made.

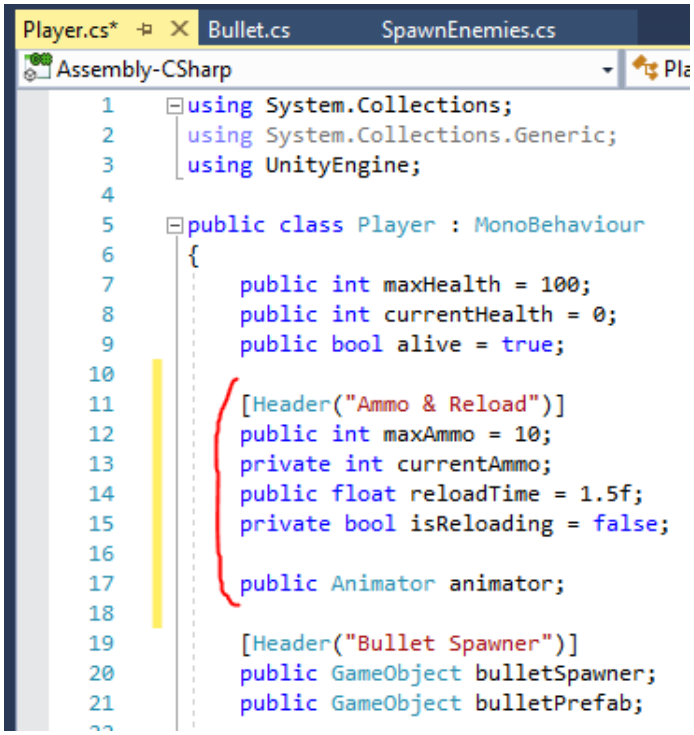
Here is the bottom part of the script i made changes to in order for the enemies to spawn properly, but that wasn't the only thing i changed or added. When i was making the changes i thought about how i could make the game more difficult the longer the Player stays alive, making the game more challenging and fun to play. So i googled some ideas to do this and i found a site to help me out and i used it to make the enemies spawn sooner the longer you play the game, up until a minimum spawn limit. Here is the site i used: "https://forum.unity.com/threads/making-enemies-spawn-faster-based-on-how-long-the-player-has-been-alive.465559/".

I did some testing and it was a bit difficult for me to understand what did what as i just copied and pasted the code onto my script and made some adjustments, but eventually i figured out how everything works.

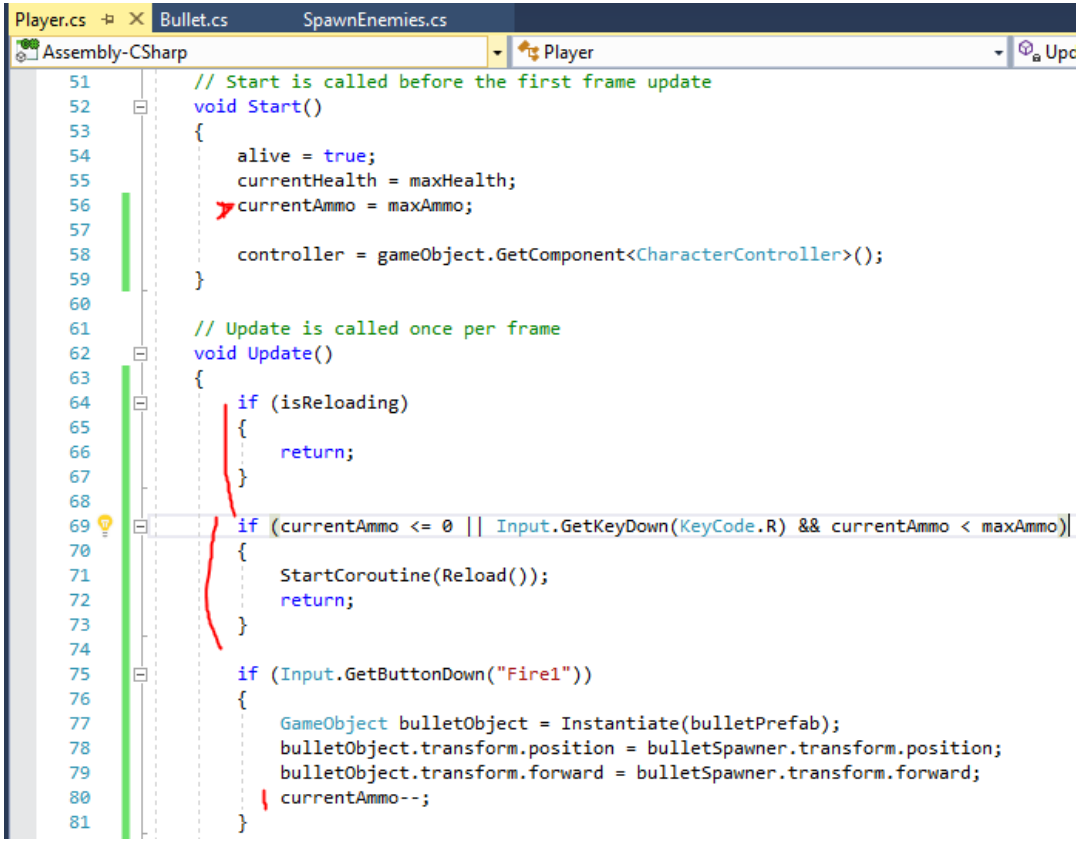
Here is the changes i made to my scene, the additional spawn points i added, the changes i made to the Spawn Enemies component in the inspector and such. What i basically did was make multiple copies of an empty spawn point where the enemies will spawn and then add them add to the Spawn Points slot in the inspector of the Spawn Enemies script component. I also adjusted the SpawnTime (the code that changes the time enemies will spawn) so it's better.

Here is the final test that i did to see how everything is working and if there are any problems. After i did the test i found out that everything works well, the enemies spawn correctly and at the right height for each enemy. They spawn randomly at the spawn points, then follow the Player correctly, even the explosion works correctly, you can see in this video at one point where a bunch of enemies die by the explosive barrel. The health potion still works correctly and everything else seems fine and working as intended. The only thing that seems inconsistent is the bullets hitting the enemies, sometimes i need the correct amount of bullets to kill an enemy but sometimes i need more and i don't exactly know why. This inconsistency usually happens when i shoot the bullets to fast or when the bullets are not close enough to the enemies even if it looks like it.

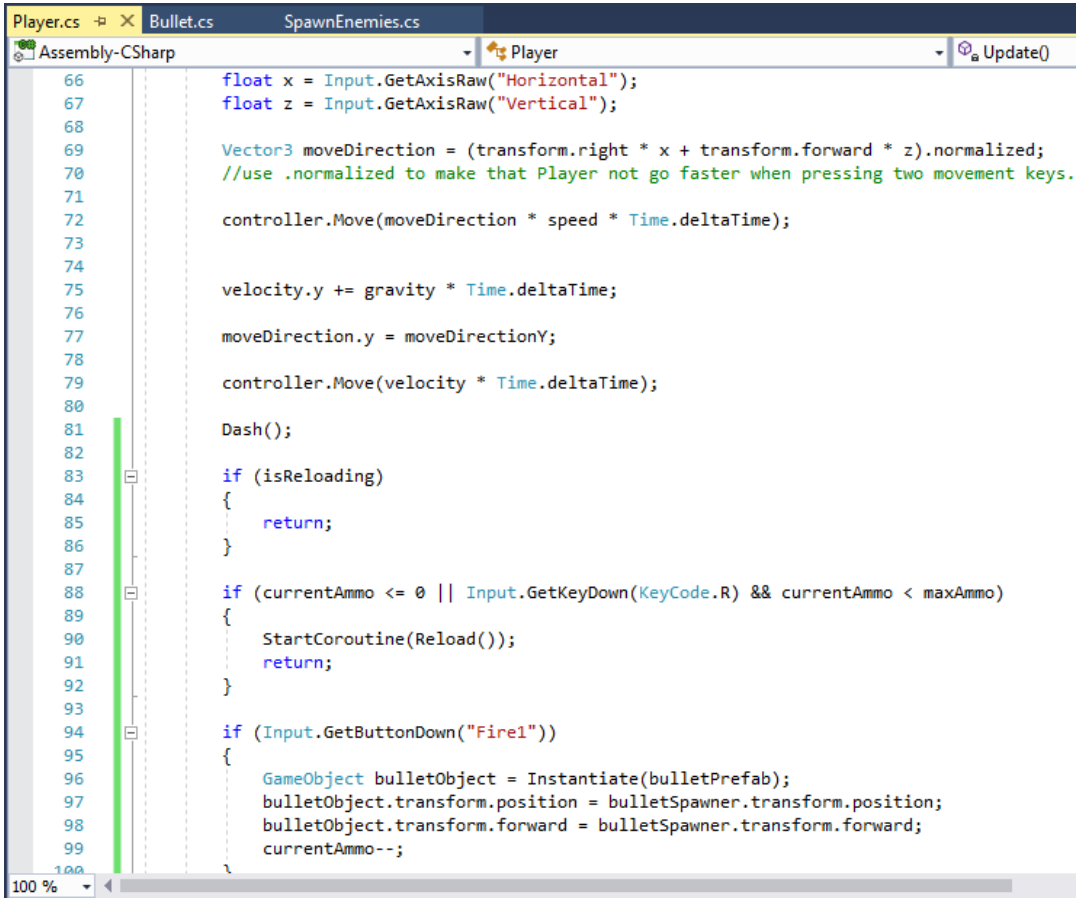
Overall i believe that this was a interesting experience, i learned a lot about spawning that will definitely be useful to me as a programmer, i struggled through a lot of things mainly but everything was solved in the end. Now i don't have a lot of things to do in terms of mechanics, just the ammo mechanics, then it's just add a very simplistic UI and menus and leaderboard score system since i don't have a lot of time left, hopefully i have everything i want completed before the deadline, i believe it's going to be very close.



The new variables i added to the Player script.



The code i copied with some changes from the tutorial i was watching.



The slight change that made the reload issue be fixed and now the Player can move and reload at the same time.

24/01/2021 was when i started to make it so my gun basically could have ammo and be able to reload in game. I first looked at videos to help me with this, and the first video, that was in my Ongoing reflection which saved me some time research, i saw and watched was a great one to begin with, the video is called "Ammo & Reloading - Unity Tutorial" and it was super helpful, simple and easy to understand.

With that i began applying what the video has taught me, the first thing i did was i basically copied the variables that the YouTube video added, to determine the reload speed, max ammo, if the Player is reloading or not and what is the ammo that the Player has at the moment.

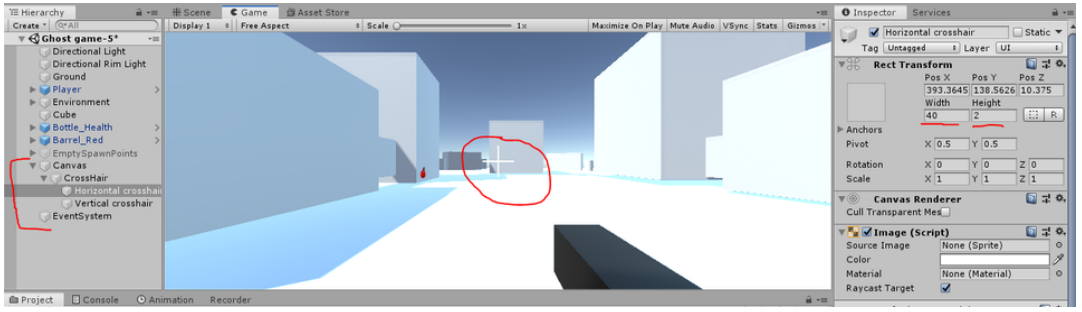
Later i copied more code that would make the reload happen when the Player does a certain action or criteria. I also modified the code a bit, for example i added the "|| Input.GetKeyDown(KeyCode.R) && currentAmmo < maxAmmo" so that the Player could reload the gun any time they want as long as the ammo is less then the max ammo.

Then lastly, i added the Reload method at the bottom of my Player script that will make sure if my Player is reloading or not, do the animation, set the current ammo to max ammo then make the reloading false. For the animator i don't have a animation yet for my gun, because i'm waiting to ask one of my teammates to give me all of their assets, like the gun, town with textures and all. After i get that i can make a simple animation that the video even shows you how.

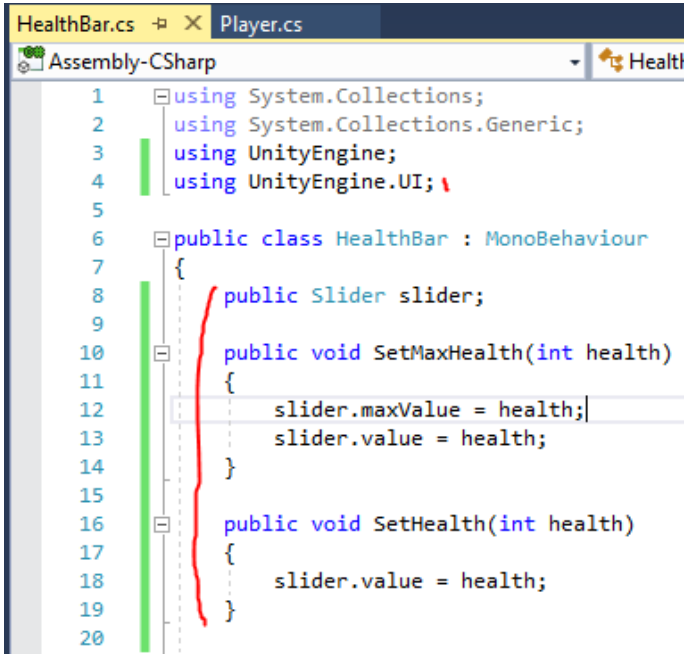
After all that i went to test the reloading in play, it was good except for one problem that every time the player reloads, whether they run out of bullets or press the reload key the Player will stop moving and i don't know exactly why this happens. It has to do something in the void Update i am sure of it.

I looked more closely into my Player script to figure out the problem of the player not moving when reloading, i believed it had to do something with the return; below the if statements. So i tried a few thing to solve it, changing the return, removing it, putting the other reload if statement in another, nothing worked, but then i thought about maybe it's the order of the code that's wrong, maybe i have to put the reload code below the movement code. So i tried that tested my game and now my game functions normally, the Player can reload and move at the same time.

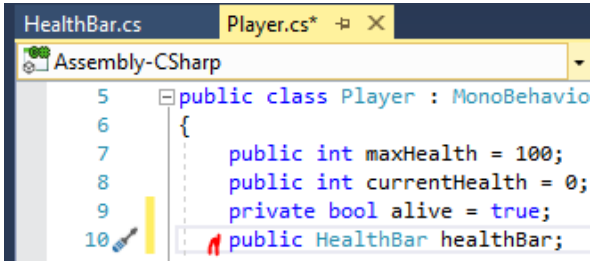
I'm glad that everything worked out, that this task was overall simple and easy to do, i'm glad i did it, it'll make my game more interesting and challenging to play. But there was one thing i couldn't do and that was the animation of the gun reloading, i need the gun model that one of my team mates has made and even added me to a organization in unity so i can go into their own game and maybe transfer the map, models and textures. But when i opened their game there was nothing there, maybe this happened because the version of their game is different and i could open it otherwise without changing the version and that messed up their game and made everything empty, i'm not sure, the only thing i know what to do is i'll have to ask my teammate to just give me the files or models directly to me through Teams or something alike in order to get everything done.



The new crosshair i created with the canvas and UI, so it's easier to shoot and aim.



The new C# script i created called "HealthBar" and the code in it.



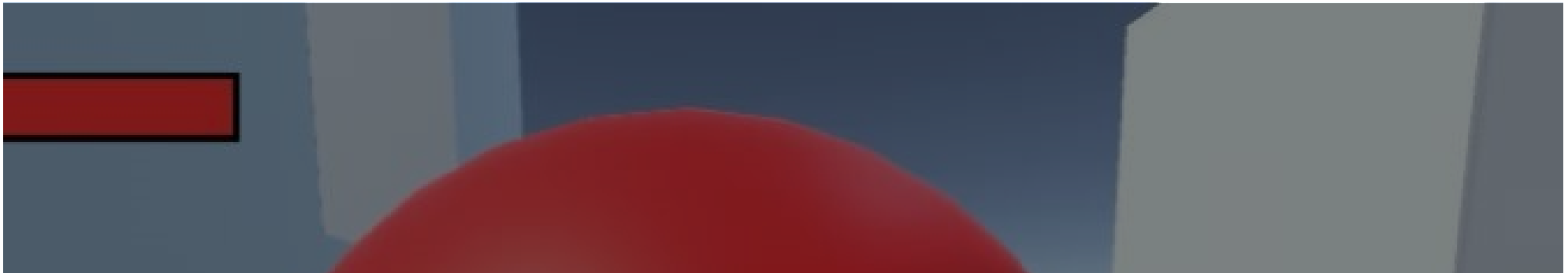
The new variable i created in my Player script.

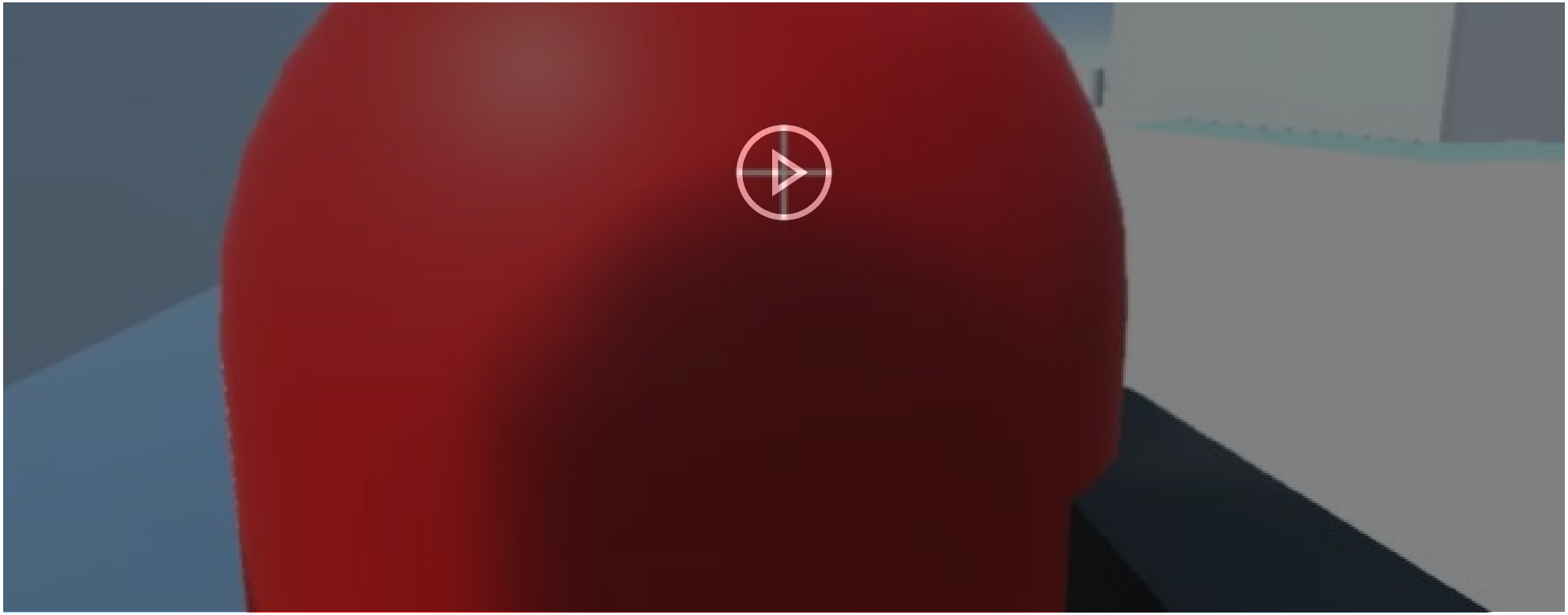
```
Player.cs*
CSharp                                  Player
    void Start()
    {
        alive = true;
        currentHealth = maxHealth;
        healthBar.SetMaxHealth(maxHealth);
        currentAmmo = maxAmmo;
        controller = gameObject.GetComponent<CharacterController>();
    }
```

The additional line of code i added in the Void Start of the Player script.

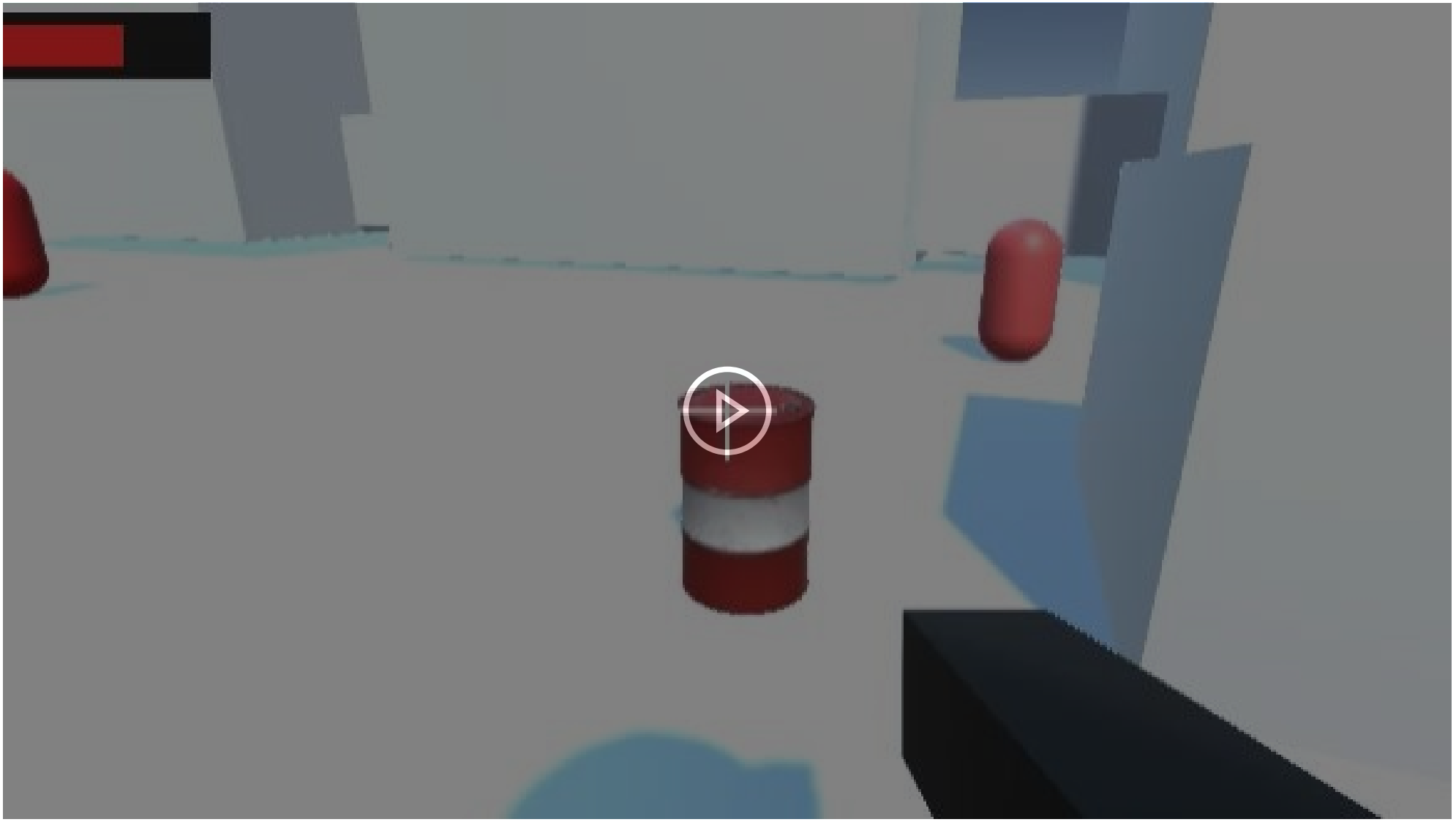
```
althBar.cs    Player.cs*
Assembly-CSharp                         Player
105     public void TakeDamage (int damageAmount)
106     {
107         if (!alive)
108         {
109             return;
110         }
111
112         if (currentHealth <= 0)
113         {
114             currentHealth = 0;
115             alive = false;
116             gameObject.SetActive(false);
117         }
118         currentHealth -= damageAmount;
119         healthBar.SetMaxHealth(currentHealth);
120     }
```

The new line of code i added in the TakeDamage method.





The test i did after adding the code to both my Player and HealthBar scripts.



The final test i did to see how the Health Bar is and how it reacts to everything. Luckily everything seems to work as intended.

26/01/2021 i finished two thing i wanted to add to my game, one was very simple to do and it was adding a simple crosshair, so it's easier to aim. The other thing i wanted to add to was a health bar to indicate how much health the Player has, without it it would be hard to play the game properly, this task was the one that was hard for me and took a lot of time to solve the problems.

What i did first was follow a video i liked called: "Unity Tutorial - Simple Crosshair" a simple tutorial that explains to you how to make a crosshair using the canvas and blank images. As you can see in this image to the left, it's hard to see but i have a crosshair, you can see the things i made and the changes i made to it in the inspector window, i like the crosshair i have, i won't change it since this one does the job.

Later after i did that, i did some research on how to get a health bar into my game, to make the life of the Player much easier. I found a video called: "How to make a HEALTH BAR in Unity!" a recent video from 2020 and with high views, so it must be good. I started watching it then i started following it's instructions, i did the things i needed to do in the editor with the images, slider, canvas and such, i even added additional things, like a health text, score text because i'm going to have a score for my future learderboard and more for my game personally.

Then i made a new C# script called "HealthBar", i added this to my Health Bar gameObject that represents the actual image of a health bar i made myself. I copied the code that appeared on the video for this script and added it to my new C#script.

After that i went to my Player script so that the Player script and HealthBar script to connect to each other and be able to change values in a script through another script. On the top of the script i added a public variable that is the HealthBar script.

Later on the void Start i added a simple line of code that is: "healthBar.SetMaxHealth(maxHealth);" hopefully this will make it so at the start of the game the health bar will be automatically set to the max health.

At the bottom of the Player script i made some modifications to my TakeDamage script, so that when the Player takes damage the health bar will be accurate to the Player's actual health.

Later i did a test to see if after everything i have done the health bar works or not. As you can see from this video my Health Bar doesn't work all that well, for some reason the Player taking damage isn't connecting to the health bar so nothing is happening until the Player reaches a health of zero then the health bar will be completely empty and then after another contact with an enemy the Player will be destroyed. I don't know why this problem appeared, i practically did everything as showed in the video i watched just with some minor differences to better suit my game. I started looking into the Player script code to see what i did wrong or what the exact problem was.

I looked around, made some adjustments and such, did A LOT of tests and nothing was making my health bar work. I was looking at my Player's Prefab inspector and i realized that the slot for the health Bar was empty, but the Player prefab in the hierarchy had it on. I looked more closely and i tried to drag and drop the Health Bar UI into the Player Prefab health Bar slot but it wouldn't let me, nothing appeared even when i pressed the small circle near to the right of the slot.

I thought that maybe i should make my Health Bar UI into a Prefab and try again to see if that worked. I did that, i tried to drag and drop the Health Bar UI to it's designated slot in the Player's Prefab inspector but it still wouldn't let me. It would only let me drag and drop the UI on the Player that was in the hierarchy and not in the Prefabs folder, but when i tired to override it on top of the inspector it wouldn't properly override or save, it would still say that the changes i made in the Player script and it wouldn't disappear, this shows me that there is definitely a problem with the player being a Prefab or Health Bar being a Prefab or even both.

Later after doing many changes to my scripts and editor with no success for solving the problem i was having, i even made my Health Bar and Player normal and no longer a Prefab anymore but nothing was still working correctly. I decided to look up a solution from the internet. I found so interesting things but none that clearly was right for me or that would immediately solve my problem. I found some sites that looked useful: "https://forum.unity.com/threads/slider-finding-gameobjects-for-instantiated-prefabs-cant-drag-drop-in-editor.469800/ " and "https://forum.unity.com/threads/how-to-intentionally-break-a-prefab-connection.382290/". But this one (https://stackoverflow.com/questions/44365701/why-isnt-unity-allowing-me-to-apply-certain-changes-to-a-prefab) at first wasn't something i thought would help me until i tried the little advice that one person gave when a person was asking for a solution to their problem that was similar to mine. The person said to add a sort of "findobjectoftype, or gameobject.find" code to better help you, i did that and added "healthbar = FindObjectOfType<HealthBar>();" and after i did that and went to test my game, my Player's health bar now reacts when a enemy attacks the Player, the actual health bar goes down according to how much damage the player takes, it gets smaller and smaller until it reaches zero then the player dies after taking another hit from the enemy.

I learned a very important lesson that i will apply to my future projects and work, how to make things work in Unity. "Prefabs can only reference other prefabs since there might not be an instance" i learned this like the person who was asking for help in the site i research that helped me. I should make gameObject into Prefabs only when it's absolutely necessary for it to be a Prefab, like trying to instantiate multiple enemies or something alike.

The image to the left are the changes i made to the editor, my objects and such so that everything works, as you can see i even have a slightly different health bar because i had to redo all of my Player and Health Bar prefabs, thankfully it wasn't too hard to do.

Here is the test i did after i fixed the problem i was having with the health bar not reacting to the Player getting damaged. Now the health bar moves when you touch an enemy and how much health Bar is shorten is connected to which enemy does the most damage. So now the health bar works properly but there are still a few minor issues, for example, the health bar doesn't update when you touch the health

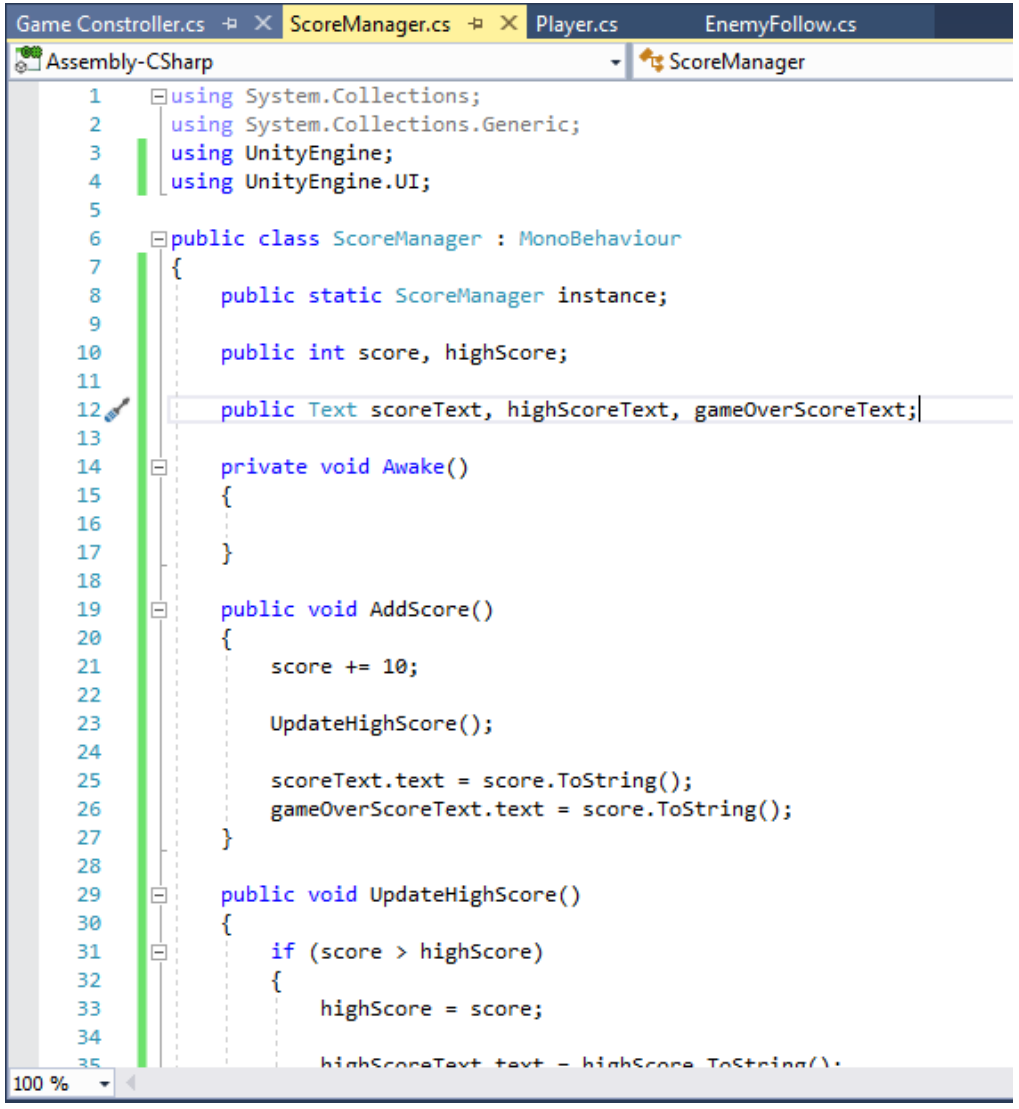
potion, it only increases your health, visually, after an enemy touches you which is weird. Another problem is that if the Player has 10 health and takes 30 damage from a big enemy the player will have -20 health but will not die, only after getting another hit from any enemy will the player finally die.

All of the problems i am having must involve the code in my Player script not updating the mechanics correctly. I believe i have to do something that involves the void Update.

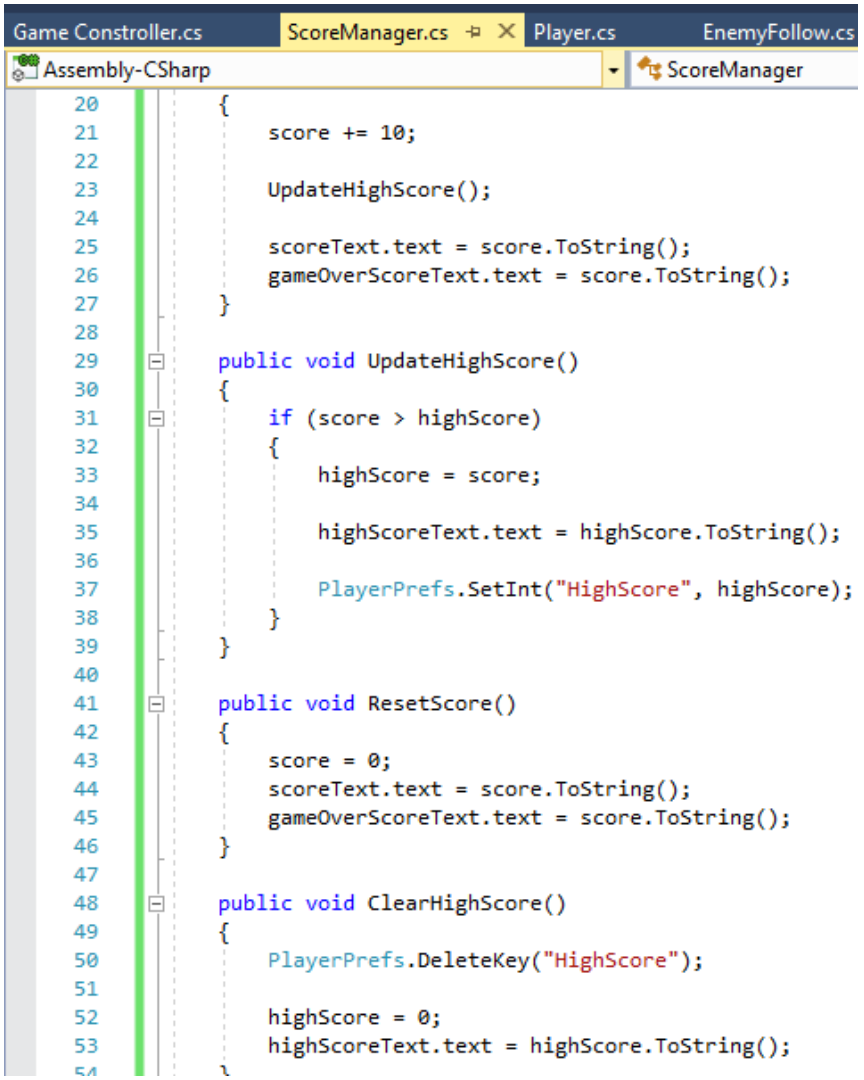
So i look at the code that involves the Player's health and health bar. I saw it and decided to copy it and paste it to the top of my void Update so that the game is constantly updating if the Player's health is below 0 or not.

After i made the changes to my Player C# script, i went to testing and now everything works as intended, the Health Bar it accurately depicting the health of the Player no matter what you are doing. The main reason i wanted to have this health bar in my game is so it's could be helpful for my audience, the player who enjoy fast FPS action games, it's good to give then a challenge but it is also necessary to help them out, like giving them instruction on the mechanics, what does what, a visual indication of their health and much more, the little things that people don't think to much about. I also decided to remove the Dash text because i don't intend to have a dash bar, as i don't have time to implement that to my game, i also won't have a menu in my game, as that may take too much of my time, i'll instead focus on polishing my game and adding a leaderboard system with points that you get by killing enemies.

I learned quite a lot with this task i just did, it will make me a much better programmer i'm sure of it. I learned what not to do, i learned something interesting about Prefabs, continues updates in the script and more that will sure help me in the future.

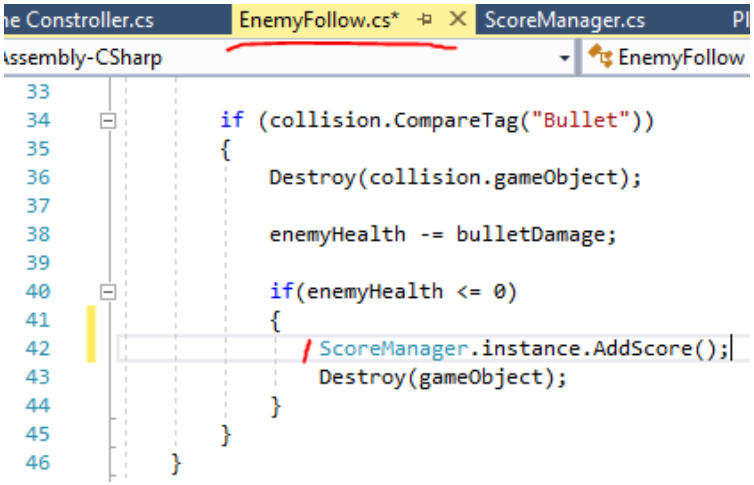


The new C# script i created called "ScoreManager" with the code i copied on the top of the script.




```
20      {
21          score += 10;
22
23          UpdateHighScore();
24
25          scoreText.text = score.ToString();
26          gameOverScoreText.text = score.ToString();
27      }
28
29      public void UpdateHighScore()
30      {
31          if (score > highScore)
32          {
33              highScore = score;
34
35              highScoreText.text = highScore.ToString();
36
37              PlayerPrefs.SetInt("HighScore", highScore);
38          }
39      }
40
41      public void ResetScore()
42      {
43          score = 0;
44          scoreText.text = score.ToString();
45          gameOverScoreText.text = score.ToString();
46      }
47
48      public void ClearHighScore()
49      {
50          PlayerPrefs.DeleteKey("HighScore");
51
52          highScore = 0;
53          highScoreText.text = highScore.ToString();
```

The code i added in my own script at the bottom.



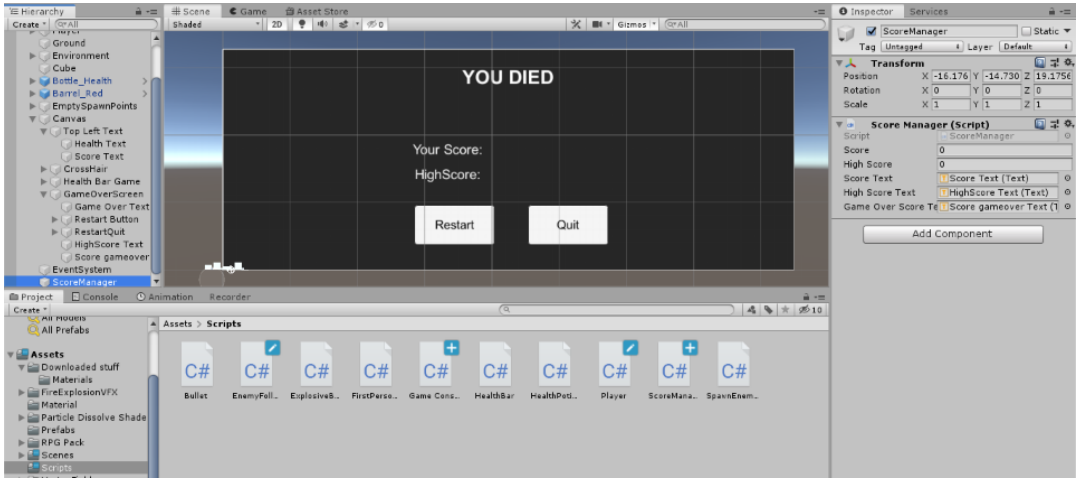
The small change i made in my "EnemyFollow" script.

Game Controller.cs | ScoreManager.cs | EnemyFollow.cs | Player.c

Assembly-CSharp | GameController

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using UnityEngine.SceneManagement;
5
6   public class GameController : MonoBehaviour
7   {
8       public GameObject gameOverScreen;
9
10      // Start is called before the first frame update
11      void Start()
12      {
13
14      }
15
16      // Update is called once per frame
17      void Update()
18      {
19
20      }
21
22      public void Restart()
23      {
24          SceneManager.LoadScene("Ghost game-10");
25          gameOverScreen.SetActive(false);
26
27          ScoreManager.instance.ResetScore();
28      }
29
30      public void QuitGame()
31      {
32          Application.Quit();
33      }
34  }
```

The additional new script called "GameController".



Here is where i added a sort of game over screen so you can see your score and high score when you die.

```
Game Controller.cs | Player.cs* | ScoreManager.cs
Assembly-CSharp
70  void Update()
71  {
72      if (currentHealth <= 0)
73      {
74          //currentHealth = 0;
75          alive = false;
76          gameObject.SetActive(false);
77          Destroy(gameObject);
78
79          gameOverScreen.SetActive(true);
80      }
81      healthbar.SetHealth(currentHealth);
82
```

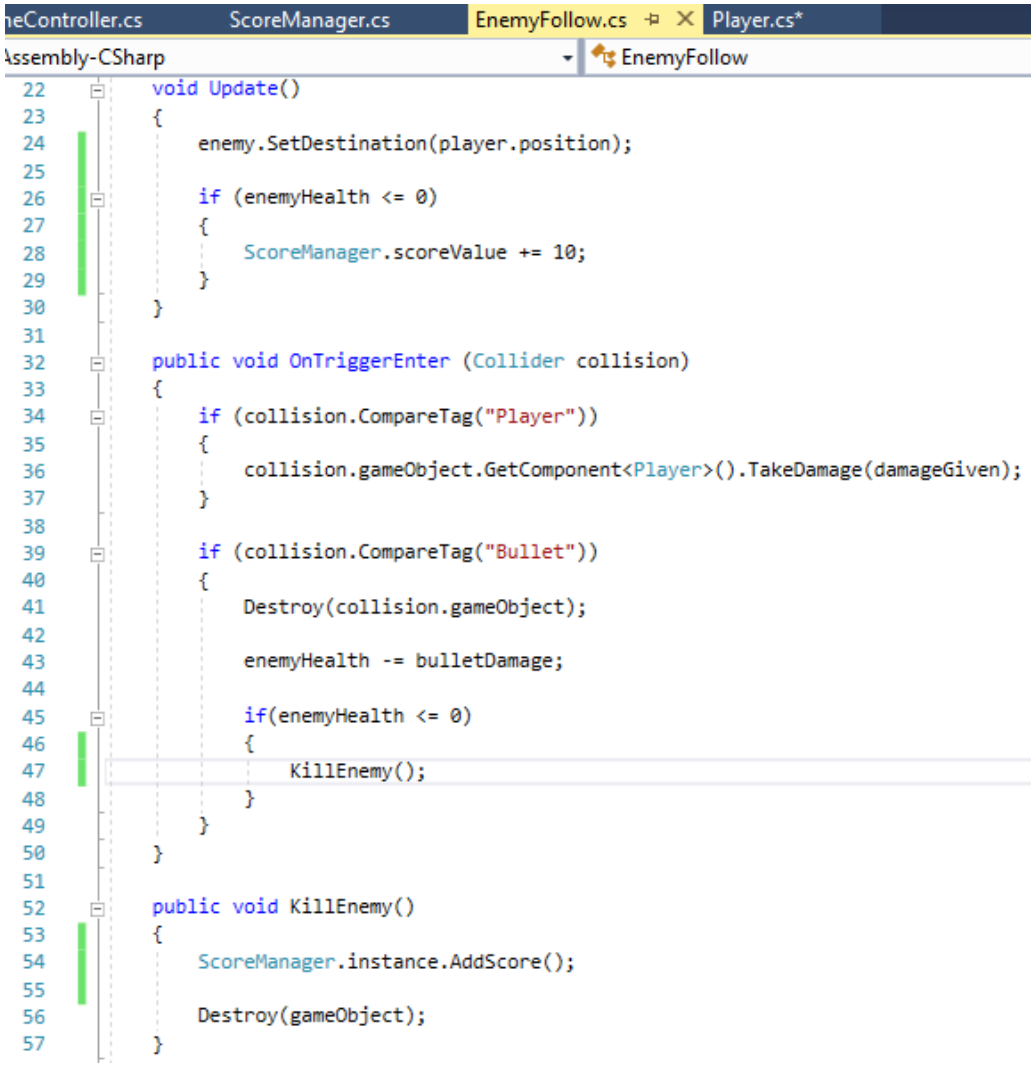
The additional line of code i added to my Player script.

GameController.cs | ScoreManager.cs | Player.cs* | EnemyF

Assembly-CSharp | GameContr

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using UnityEngine.SceneManagement;
5
6   public class GameController : MonoBehaviour
7   {
8       public GameObject gameOverScreen;
9
10      public void Restart()
11      {
12          SceneManager.LoadScene("Ghost game-10");
13          gameOverScreen.SetActive(false);
14          Cursor.visible = false;
15
16          ScoreManager.instance.ResetScore();
17      }
18
19      public void QuitGame()
20      {
21          Application.Quit();
22      }
```

The changes i made to my GameController C# script.



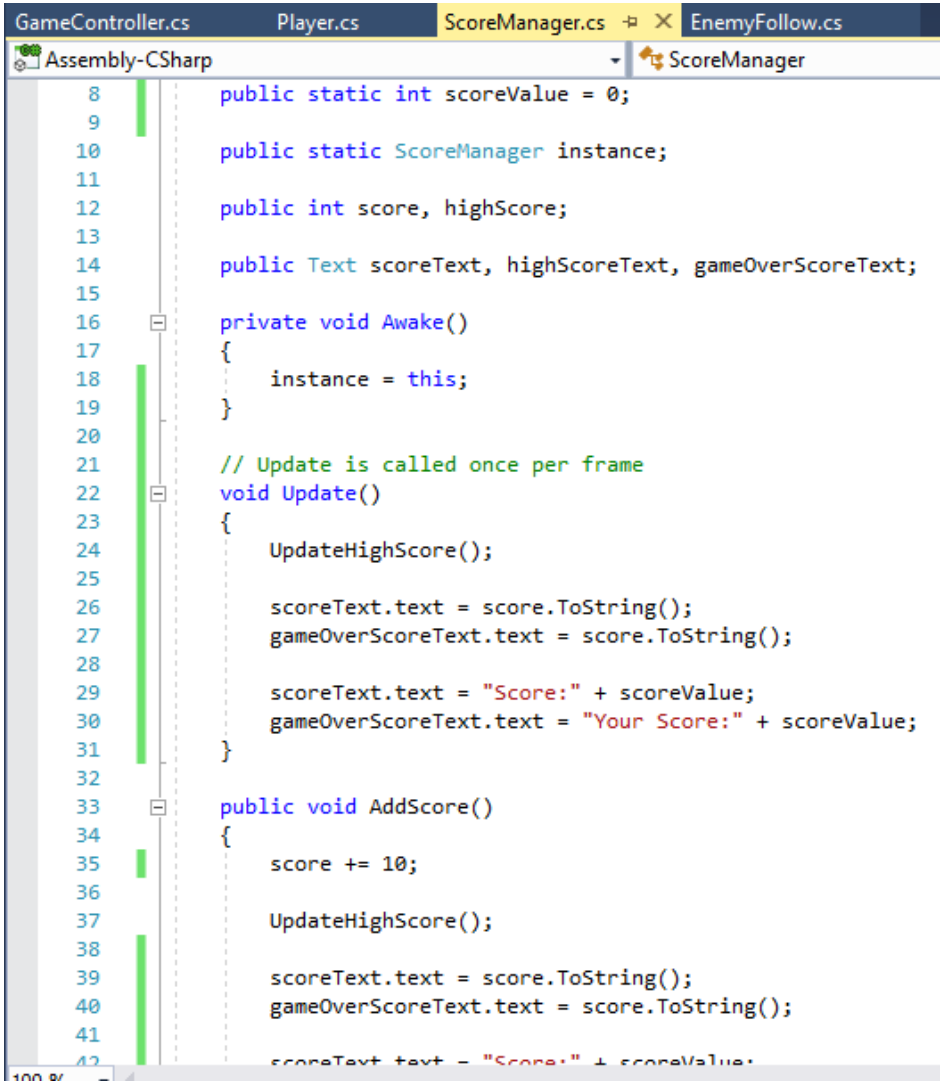
The changes i made to my "EnemyFollow".

GameController.cs | Player.cs | ScoreManager.cs | EnemyFollow.cs

Assembly-CSharp | Player

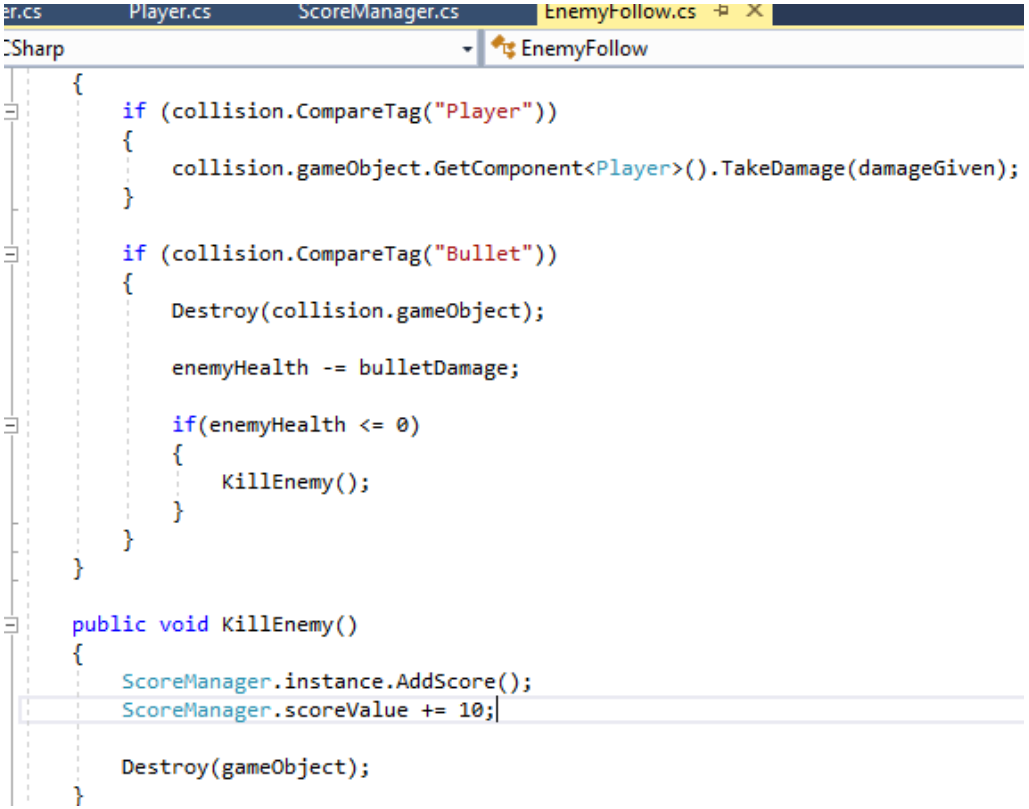
```
40      public float dashLength = 0.15f;
41      public float dashSpeed = 100f;
42      public float dashResetTime = 1f;
43
44      public CharacterController characterController;
45
46      private Player player;
47
48      public GameObject gameOverScreen;
49
50      private Vector3 dashMove;
51      private float dashing = 0f;
52      private float dashingTime = 0f;
53      private bool canDash = true;
54      private bool dashingNow = false;
55      private bool dashReset = true;
56
57      // Start is called before the first frame update
58      void Start()
59      {
60          alive = true;
61
62          currentHealth = maxHealth;
63          healthbar.SetMaxHealth(maxHealth);
64
65          healthbar = FindObjectOfType<HealthBar>();
66
67          currentAmmo = maxAmmo;
68          controller = gameObject.GetComponent<CharacterController>();
69
70          player = gameObject.GetComponent<Player>();
71          gameObject.GetComponent<Player>().enabled = true;
72      }
73
74      // Update is called once per frame
```

The additional lines of code i added to my Player C# script.

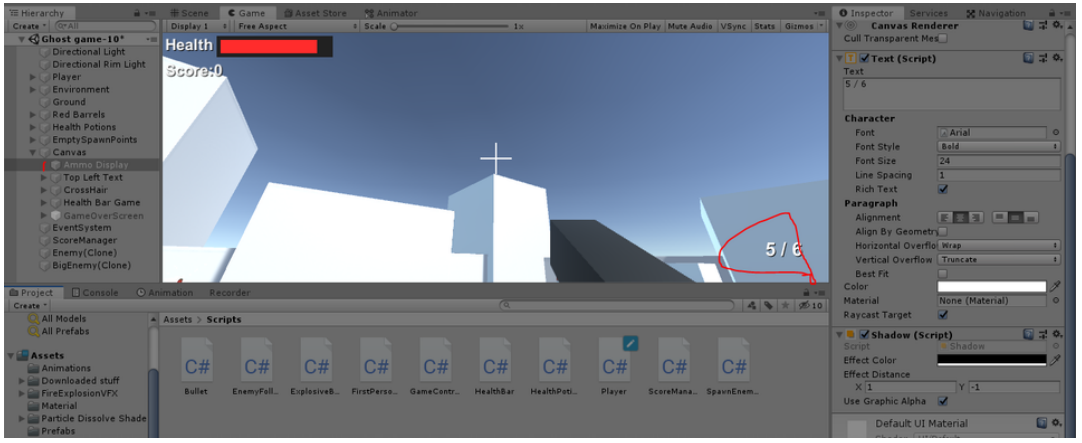


The changes i made to the top part of my ScoreManager script.

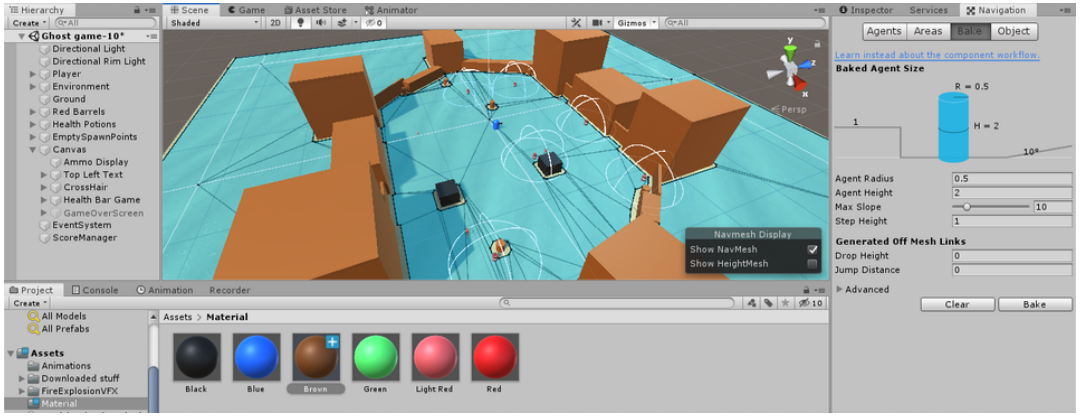
The changes i made to the bottom part of my ScoreManager script.



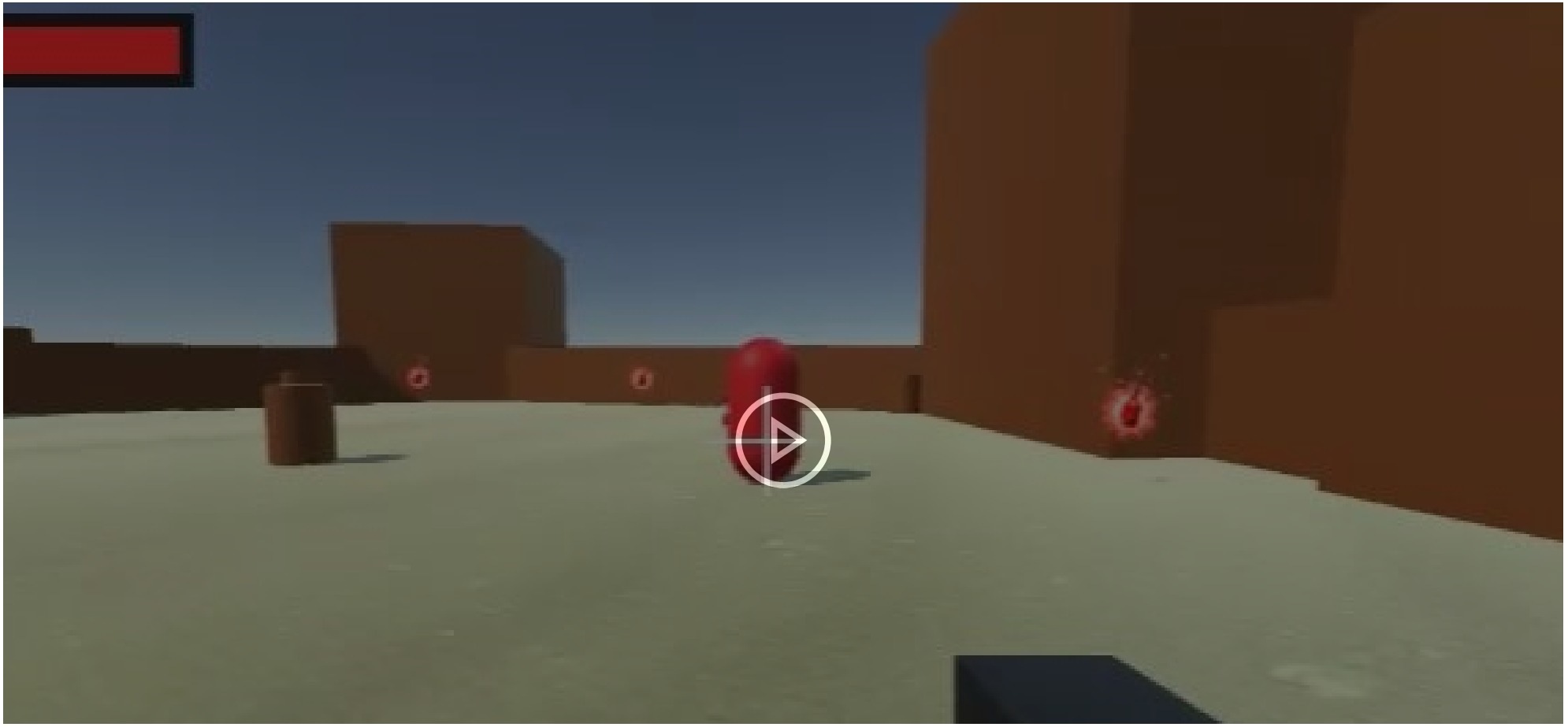
The minor changes i made to the EnemyFollow script.



The changes i made to my UI so now you can see how much ammo you have before reloadining.



The change i had to make for the navigation in order for enemies to properly follow the Player.



The final test after i have added every feature possible with the time i have left.

At 29/01/2021 i decided, after much thought, that with how much time i have left until i submit my work after that, i won't be able to update my SharePoint. So with that in mind, i did some research on making a leaderboard and with the videos and sites i saw it seemed a bit complicated to do and would take too much time and effort to do with the time i have left. So i decided to make a simple score system with a high score when you die, then i'll redo my map, make it better and not some sort of test map, then i'll add a simple gun reloading animation and a ammo display, displaying how much ammo you have left.

With all that out of the way i started research way to make my enemies give the Player points toward the score in the upper left corner UI next to the health bar. I found a lot of videos and websites on the internet to help me with this, but i mainly am using a video called: "Storing High Scores in Unity 2019" as i believe this is the best one for me, not to simple but also not too complicated.

After watching the video i then copied all of the code it had onto the new C# script i created called "ScoreManager", this will make sure everything involving scores and points works and is updated properly. The player will kill an enemy and the the AddScore method will be activated giving points to the player and with that the score UI will be updated hopefully.

Then depending if the normal score is higher than the high score, then the high score will be updated to be the same as the score. Then when i make a way for the Player to restart the game the score near the health bar and the score in the game over screen will be set to zero and hopefully the highscore will stay the same. I also added PlayerPrefs in this code, that i copied form the video, so that if you exit the game and play again you'll still have your highscore, you won't lose your progress. Hopefully this works i'm not too sure about this as this is

a lot of code that i don't understand and it's a lot of stuff i have never done before so i have no experience in this, i personally believe that there is going to be some kind of problem on the way, that hopefully i can solve.

After doing everything i needed in the new script i made, i moved onto my EnemyFollow script and added one simple line of code connecting this script to the ScoreManager script. So if an enemy's health reaches 0 then the Player will get points. This is where the code deviates from the video i'm watching because in the video the person doesn't have enemies, so i'm using my own logic to apply the code in the video to my own game and hoping everything works as intended.

After that i made another new script called "GameController" that will be used to control the game Over Screen, so that it can activated when the Player dies, so that the Player can restart the game and quit it with it's buttons, a simple script overall.

Later i went to my Unity editor and i made my Game Over screen. Basically i created a image that would block the players vision after they die, then their will be a title, your score and highscore and then there will be tow buttons, one to restart the game, so you can play again and get a higher score, and the other one is to quit the game.

In the Player script i made some adjustments in the void Update so that the Game Over Screen be activated after the Player's health reaches zero.

After i did all that i decided to test my game to see if everything works. After i did the test i noticed some problem that i knew were gonna happen, for some reason or another the score wasn't updating, not in the scene UI or the inspector. Also the highscore wasn't updating, even if i manually updated the score in the inspector.

The main problem i see is that for some reason the death of an enemy isn't translating that to getting points, nor is the score connected to the highscore so it won't update, i'll have to do more reearch or find a way for everything to be connected in order for things to work.

Another problem was that when the Player died the game over screen would appear but the Player wouldn't be able to interact with it, not even with the arrow keys, so i'll need to make a way for the Player to see and more the mouse cursor when the Player dies, so it can press the buttons.

I added one line of code that is: "Cursor.visable = false:" so that the cursor is not visible again after you press the restart button in the game over screen, i also made the script a bit more clean.

In the EnemyFollow script i made a small adjustment after the research i found and watched a different video called: "How to add a score counter into your Unity 2D game| Easy Unity 2D Tutorial", hopefully this will make the score go up by 10 points when i kill an enemy.

Then i wanted to make it possible for the Player script to be disabled when the player died so that you can't accidentally shoot an enemy and get more points when the game should be over. But first i needed to make it so that at the start the Player script was active, i don't want to accidentally restart a game with no Player script, otherwise how would i move and shoot with no script.

Then i made it possible for the mouse cursor's visibility and Player script to be active when the player's health reaches zero. I used the "Cursor.visible" code that is same as the "FirstPersonCamera" script just now is true instead of false.

Later i went to my ScoreManager script after seeing multiple different videos and site on how to get a type of score system in your game, I decided to add this on top of the other code i was using, it may be confusing and unnecessary but i want to do this because i don't want to delete the old code, have the new code not work and then i'll have to figure out what lines of code work and don't work. Everything might get complicated and people might not understand what i'm doing or what this code is, but that's a risk i'm willing to take.

Here is the bottom of the script and the additional code i added to it, it may look like i just have a lot of repeated code in it, but i hope that this will make the score appear in the UI and inspector.

In the EnemyFollow script i simply added one additional line of code that is: "ScoreManager.scoreValue += 10;" hopefully this will make it so the player will gain 10 points upon the death of an enemy.

As i was making these additions to code in the scripts, the changes and adjustment i made to them, i kept doing tests, testing if the score updated, if the highscore changed if the score was high enough and the game over screen. To see if what i was changing working or making things worse, luckily the changes i was making made the game slightly better and i was seeing progress, the score was being updated, UI, game over screen, inspector and all.

But there was still one small problem, and that is the mouse still can't be moved when the Player dies, you can now see it, but for some reason you can't move it. To solve this small problem i did some research on mouse cursor code in Unity and i found this: "https://docs.unity3d.com/ScriptReference/Cursor-lockState.html" this helped me understand how to make it possible to move my cursor again after the player dies. So i added one simple line of code to unlock the mouse upon the death of the player. I played my game to see if i can see and move the mouse when entering the game over screen, and i was able to move the cursor and click on the buttons, i could quit the game if i build it, which i will, and i can now restart the game and play again with the score now zero again.

Here is my hierarchy, the changes i made to it and more. At the start of the game the game over screen will be disabled, along with everything in it, the text, buttons and all.

After all the changes i made i did a test to see if everything i did was helpful or if i succeeded in getting what i wanted. By the end of the test i realized that i fixed most of the problems that my score system had, but there is still one small problem, the highscore doesn't properly update when you restart a new game. When you get, for example, 70 points and then die the highscore will be 70 but if you press restart and get 40 points and die, the highscore will say 40 when it should be 70. I was thinking about how to fix this problem but then i realized i don't have a lot of time to fix problems, i have to finish this game up, polish it, then i have to document all of what i did in SharePoint which might take some time.

Later after doing the tests, i decided that i wanted to update my map and everything in it. Basically what i did was move the blocks, cylinders and rectangles and adjusted their size so that the players arena is much bigger and better to fight. Then i made duplicates of the health potion, EmptySpawnPoints and explosive barrels and scattered them around the arena. Now the map looks much better and feels better to play at.

Later i decided that i wanted to finally add a reload animation to my gun, i didn't do it last time because i was waiting for my gun model from my teammates to appear, but then i got the files for it but i couldn't open it. Me and a team mate of our group told me to download WinRAR, so i did that, tried it again and the models still don't work or don't appear. SO i decided since the models don't work, might as well try with the simple black rectangle bloc i have in the scene. What i did was rewatch a video i used for the reloading mechanic, when to the part of the video that talked about animation and did what happened in the video. I made a simple animation with the Animation, just moving the gun's X rotation then saving the animation. After that i go to the Animator and make some transitions and a bool parameter called "Reloading" so that if "Reloading" is true the then animation will activate if not then the animation will stop or no animation will play, this was quite simple to do overall.

Then i went to my Player script and un-commented the code i wasn't using but now i am, and added some new code, mainly the new yield return new WaitForSeconds and 0.25f seconds.

Later i wanted to add a sort of ammo display so it would be easier for players to know how much ammo they have. I found a video called: "How To Make A HUD in Unity (4. Ammo Display)", i'll basically copy the code that this video has but of course with some adjustments since my game is different. What i first did was add two variables that would help me make the ammo display.

Later i added some simple lines of code that would help me make the display possible. I added a line of code that would make it accurate to how much bullets the Player has, and have the " / 6" next to it to tell you what your maximum bullet amount is.

As you can see This is what the ammo display looks like in game, to do this i made a simple text UI and placed it in the bottom right hand corner, i also made all of the text UI a bit better by adding a shadow behind the text so it's looks better and it's easier to see.

Then i went to my navigation tab and a clicked bake so that the enemy AI navigation is updated to the new map and enemies can move around the objects. Another thing i did was add a very basic brown colour to all of the block which are suppose to be buildings, so now with a simple colour it's much better to look at. It could be even better with some actual models but i couldn't get them to work so i have to do this instead to make my game at least bearable and presentable.

After all of the changes and features i added to my game i did a final test to see how everything is going and what things i fixed, what problem persists and how everything looks now.

I believe that everything in my game works well, the shooting, movement, dash, health, health potion, explosion, enemies and everything works. But there are a few issues that aren't major but they do bug me a little. Here are the things that i couldn't fix properly: some times you can't dash even if you didn't dash for quite a while and when you dash through a enemy sometimes you won't collide with that enemy

and take damage, the same thing applies to the health potion. The highscore doesn't update properly, if you get 100 points, die, try again and then get 50 points the highscore will say 50 when it should say 100. The ammo display doesn't display that you have 0 bullets for some reason, if you have one bullet left and shoot, the display will still say one, just like the video to the left.

These things i couldn't fix them in time, but luckily these issue are not a big deal for most of the time, if there was some problem that would crash or make the game unplayable, that would be worse.

Besides these problems, there were a few features i couldn't implement in my game because they were too complicated to do, i didn't have time, the resources weren't given to me in time or i didn't prioritize them first thus i didn't complete them first. The things i wanted in my game but couldn't were, a intro animation, a leaderboard system, a terrain, building, enemies and gun models, a proper menu, music and sound effects.

I couldn't add the intro animation because the our fourth team mate didn't give me the animation, so i couldn't do anything about it. I didn't add a leaderboard to my game because it seemed too complicated for me and i didn't have a lot of time to mess around with that. The terrain and models, were given to me but i couldn't get them to work, even after i downloaded WinRAR. The Menus and music i didn't add because they weren't a high priority and i didn't have much time to make them.

I did what i could do with the skills, resources and time i had at the time, i believe i did a decent job. There were things i did right but there were also things i did wrong and could improve upon. But everything i just experienced will help me in the future by giving me skills, experience and more, all to make me a better programmer.