



Production Journal



Diogo Constantino

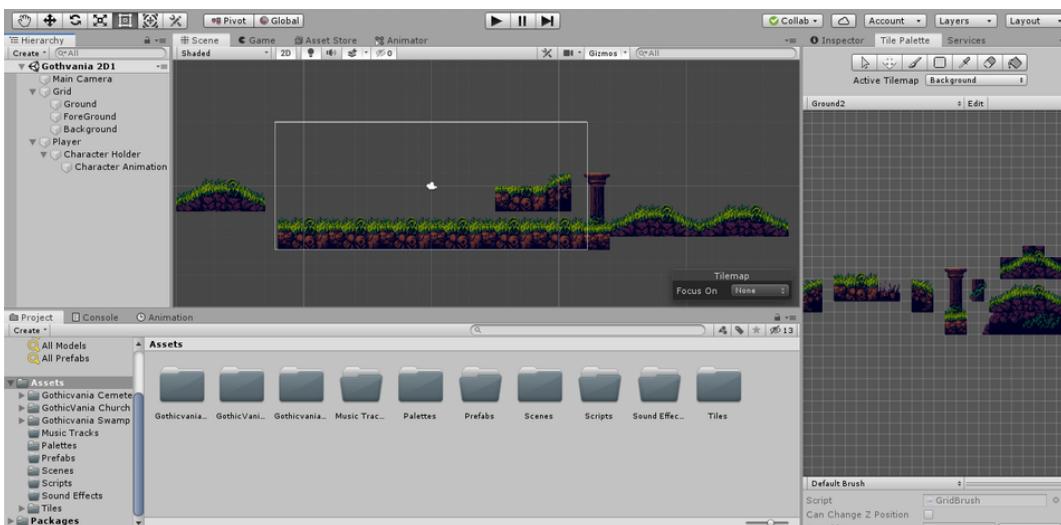
05/02/2020: I have downloaded and imported a 2D pixel art work that i found in the asset store to my game, in unity. I looked at the assets i have and it was pretty good in my opinion with high quality, especially for free, lots of animation, monsters, sprites, tilemaps and more.

After i looked at my assets and all i wanted to make a tile set with the pixel tilemaps sprites i have with the Gothicvania Cemetery Artwork. I right-clicked in the hierarchy window and clicked on 2D Objects > Tile Map, then i needed to go to window > 2D > Tile Palette. At first i tried to make The tile palette with the tileset that i have in the Environment folder, but there was a problem and i couldn't use it, so i thought that maybe the tileset-sliced might work and it does, i can drag the sprites and place it in my scene, i think the reason this type works is because it is sliced, making a square of a sprite separate instead of all together.

But then i noticed another problem, the sprites were very small compared to the grid box, it wouldn't even fill a square, but i quickly fixed the problem thanks to the research i did about pixel perfect, all i had to do was make sure that the Pixel Per Unit was 16 not 100, because that was what i saw the person made this assets was made for. I also had to check if Filter Mode was Point (no filter) and not any thing else, and finally i had to make sure that the object sprite Compression, in the inspector window was None and nothing else.

After that the tilemap i add to my scene look great now are are functional, i did the solution to all of my sprite from all the pixel assets i have, so it took some time but it wasn't hard, i could easily make changes by shift-clicking objects and changing their inspector to what needs to be, like Pixel Per Unit be 16 and more.

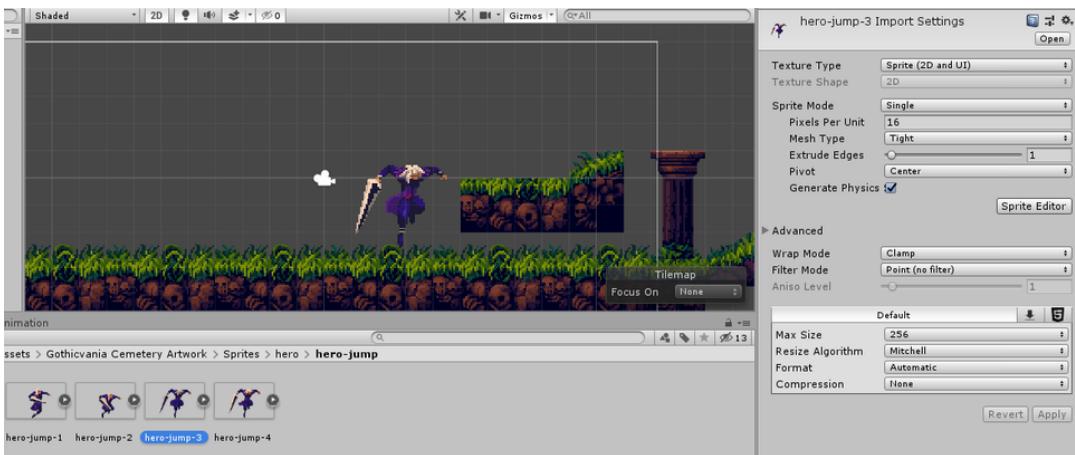
All this was possible thanks to the research i have done, especially the Pixel Perfect websites i read about and the video i watch about Tilemaps made by Brackeys.



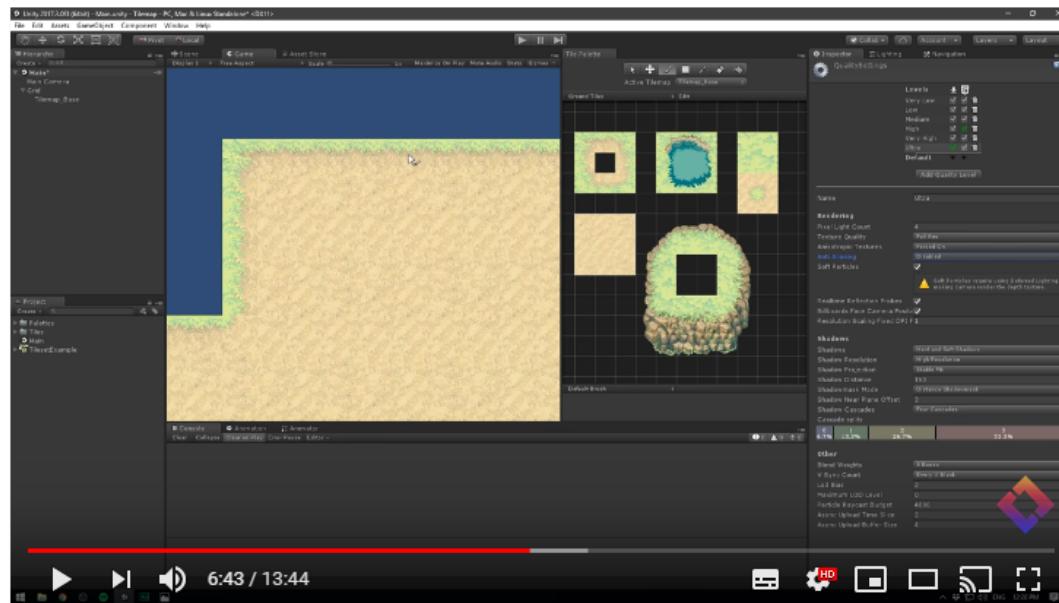
The assets i have in my game, and the tilesets i have in my scene, as a example.



The problem i encountered about a sprite being too small for one square of the grid.



How i fixed the problem i was having, simply change the Pixel Per Unit from 100 to 16.



TILEMAPS in Unity

645 697 visualizações • 31/01/2018

14 MIL 209

PARTILHAR

GUARDAR

SUBSCRITO



A video i saw about how to use Tillemaps in Unity.

25/02/2020: I finally started the actual production of my Project, so first i started to follow a certain video on YouTube called "2D Character Animation in Unity [pt 1 of 4]" so that i can implement a animation for my Player.

I then created a file called "Animations" and created a animation by clicking on the "Create" button on the Animation tab in unity that i have and call it "Running", which is where i'll place my running animation of my player, then i click on the play button of the Animation tab to preview it on my scene and it started moving, like a actual animation but it was moving too fast. So i decided to change the sample box from 60 to a lower number i liked but there was a problem, i encountered, there was no "Samples" box with numbers unlike the video i'm watching in other to do all this.

I tried looking around my Unity and the Animation tab to see where it was but nothing, i then moved the diamond shape dots in the Animation tab, i separated them further from each other, making the time between each frame longer and thus the animation is slower. I tested it a bit but the animation was never alright for me, some sections were one frame more apart then the others, it wasn't moving smoothly and other complications.

So i decided to go to google and search for a solution for my missing Samples box, and quite quickly i found the solution, i simply had to click on the small gear with an arrow pointing down then click on the show Sample Rate option and that fixed my problem instantly, turns out it was very easy.

After that i went place the dots in order and next to each other frame by frame, then changed the Sample rate from 60 to 14, pressed play and it was animating smoothly and working fine, but i didn't like how fast it was moving so i changed the numbers until i was satisfied. I found that 9 sample rate seem like a good speed in my opinion. I did this to all the other animations i created like the "Idle", "Falling" and "Jumping", after i placed my animations to the Animation tab.

Later in the Animator tab i organized my the animations i have and then created two new floats in Parameters called "horizontal" and "vertical". Then i made a C# script called "CharacterController", in this script, or now, it will carry the code for the animations, in order for my Player to animate at the start of my game. Then i added the new script i created to my player object and i clicked on the small circle next to the Animator box of the CharacterController script component and tried to find the Character Animation GameObject but i couldn't find any the only option i had was the Player GameObject, so i clicked on that and then press Play to see if the idle animation starts working in the scene and it does. I don't know why i couldn't drag or had the option to choose the Character Animation object, but my scene still works so i'll just leave it has it is.

After that, i even tested to see if the float in the Animator increase or decrease in value after i try to move the player with the arrow keys in play mode, and it works.

This was the web site and where i found the solution for the missing Samples box.

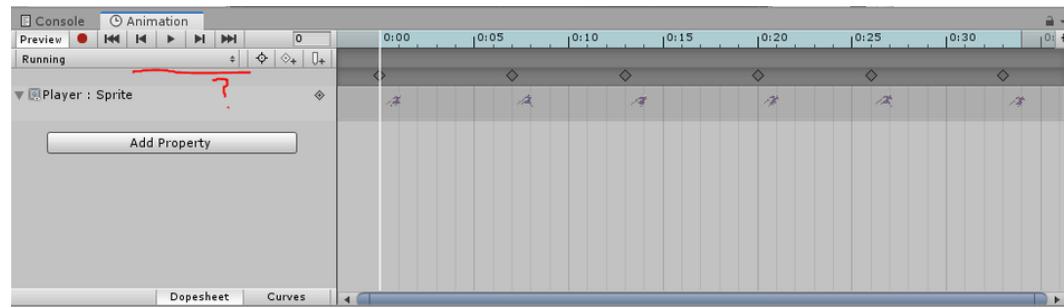
Link to the web site that helped me: (<https://answers.unity.com/questions/1642306/dont-have-sample-on-animation-windows.html>).



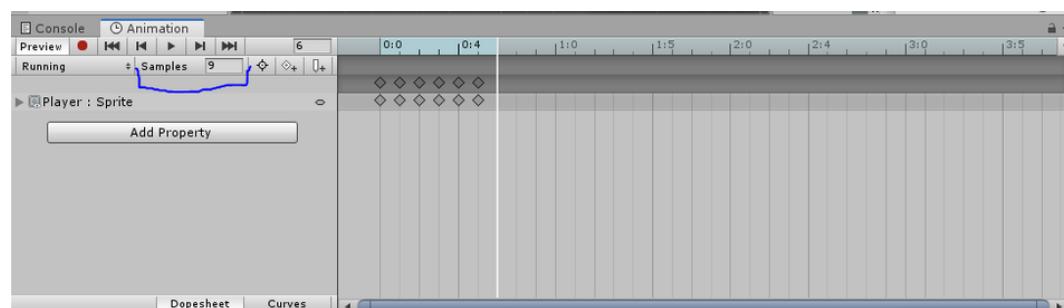
Don't have "Sample" on Animation Windows - Unity Answers

answers.unity.com

Unity is the ultimate game development platform. Use Unity to build high-quality 3D and 2D games, deploy them across mobile, desktop, VR/AR, consoles or the Web, and connect with loyal and enthusiastic players and customers.



The lack of the Samples rate box problem i had in my unity.



What happened after i fixed the lack of Samples Rate box, by going to the small gear box on the top right corner of this image.

Add comment · Share



Answer by unity_gTM5flAnuBv70A · Jun 20, 2019 at 08:02 PM
No topic about it, very annoying :/



1



Add comment · · Hide 1 · Share



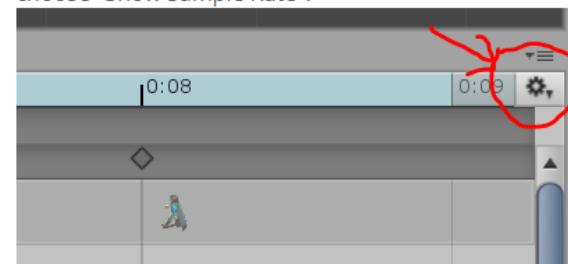
fistonfire03 · Jun 20, 2019 at 10:36 PM 0
If you manage to find a solution, can you tell me? Thanks



Answer by vsKelly · Aug 01, 2019 at 02:05 PM
On the far right of the Animation window, to the right of the clips, click the gear icon and choose "Show Sample Rate":



1



capture.png (9.9 kB)

Add comment · Share

Where i found the solution to my problem

26/02/2020: I continued where i left off, so i went to the Animator tab and started making transactions between animation states by simply dragging a arrow to where i want to transition.

I would then click on the transaction line, change the inspector a bit so the animation is better, like uncheck the exit time box and change the Transaction Duration to 0. Then i would click on the plus sign of the Conditions box in the inspector window, i did this to most of my states.

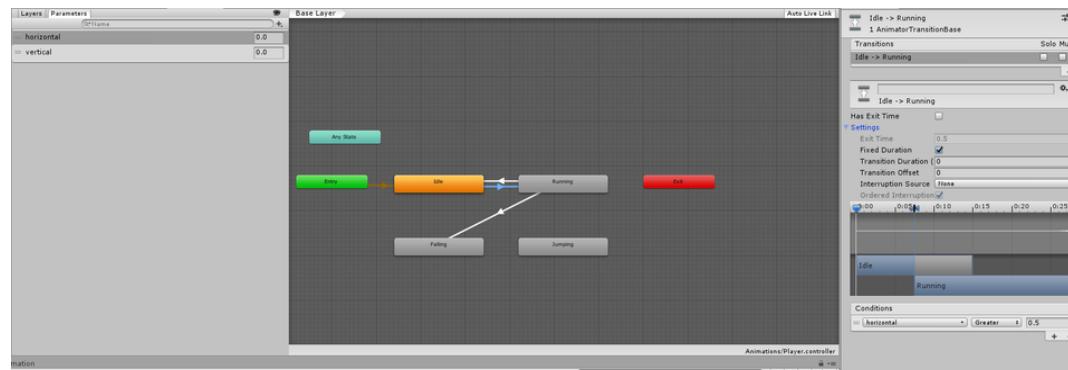
I then entered play mode and tested all this by using the arrow keys to move, and it all works well, the Player starts to run after i press the right arrow button and goes back to the idle animation when i don't hold the right arrow key, the Player starts "falling" when press the right and down arrow button but it would not go back to the running or Idle animation, i think the reason for that is because i haven't made a transaction for animations, especially to go back on other animations, yet but i will.

I also added more stuff of my CharacterController script, basically a "Mathf.Abs" in the animator.SetFloat "horizontal", the video i was watching as a tutorial said that this is suppose to help make the animation go back to the idle animation. But this script seems to have a problem or warning to it in the console saying: "Script 'CharacterController' has the same name as build-in Unity component. AddComponent and GetComponent will not work with this script.", which made me worry a bit about the script i had.

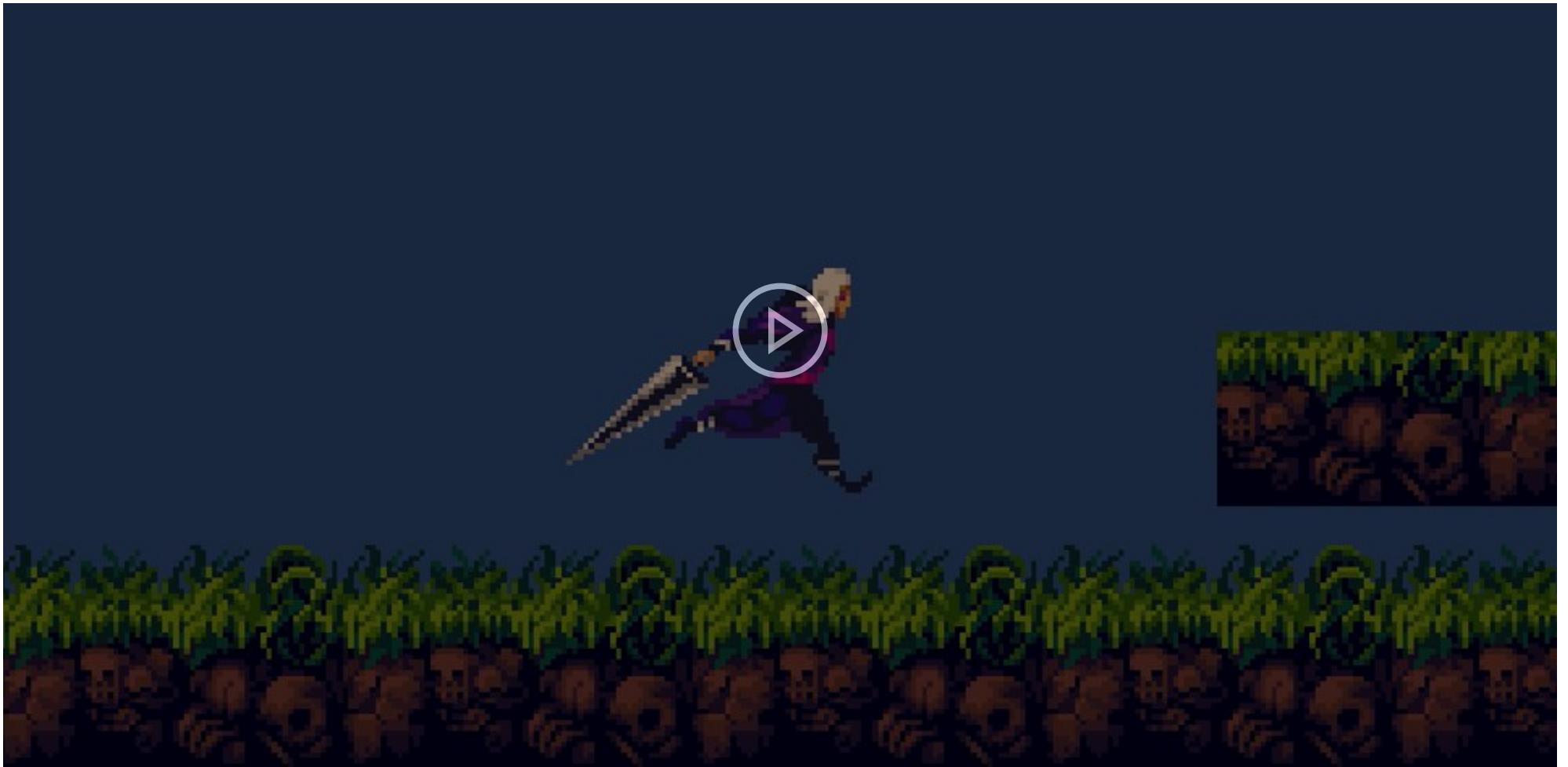
Because of that i have made another C# script called "PlayerController", i made this script for incase the other script has a problem in it or i can not use it. This way i can simply copy paste the code in my CharacterController script in order for the script to be the same and work as it is suppose to, without rewriting all the scripts.

But this will only be used incase of emergency.

After that i wanted to add more stuff in my scene so that i can test my Player and it's mechanics properly and see how the terrain looks when the Player, eventually moves. This will be like a test scene where it will all the different types of terrain, uphill, down hills, so i can test the Players movement really hard, and it will have all the enemies, to see how they work, wit the environment and the Player, all this will help me make possible modifications to my GameObject so it's better of every one when they play my game.



My states and the transaction i made to the states. And the changes i did to the transactions inspector window.



The test i did to see if i'm able to control the animations with the arrow keys and to see if the animations work.

The screenshot shows the Unity code editor with the tab bar at the top. The active tab is "CharacterController.cs". Below it, there are tabs for "Assembly-CSharp" and "PlayerController.cs". The code editor displays the following C# script:

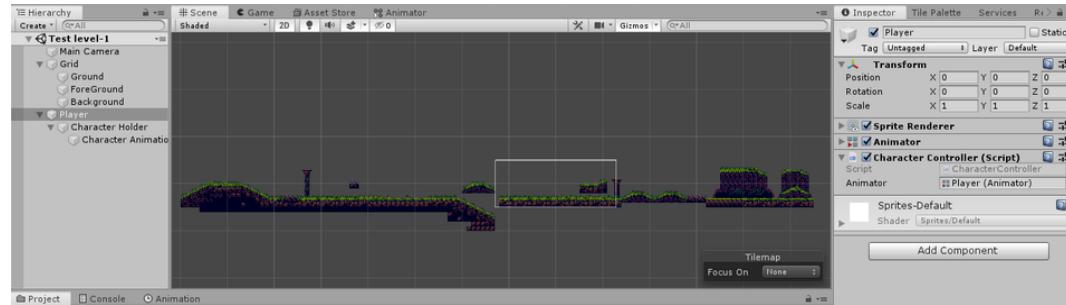
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class CharacterController : MonoBehaviour
6  {
7      public Animator animator;
8
9      // Start is called before the first frame update
10     void Start()
11     {
12     }
13
14
15     // Update is called once per frame
16     void Update()
17     {
18         animator.SetFloat("horizontal", Mathf.Abs(Input.GetAxis("Horizontal")));
19         animator.SetFloat("vertical", Input.GetAxis("Vertical"));
20     }
21 }
```

My CharacterController C# script that for now has the code to make the Player show animations.

The screenshot shows the Unity code editor with the tab bar at the top. The active tab is "PlayerController.cs". Below it, there are tabs for "Assembly-CSharp" and "CharacterController.cs". The code editor displays the following C# script:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerController : MonoBehaviour
6  {
7      /// <summary>
8      /// This will be used in case the CharacterController script doesn't work,
9      /// because I keep getting warning in the console of unity that says there is a thing with that name already and i can't do that.
10     /// </summary>
11     public Animator animator;
12
13
14     // Start is called before the first frame update
15     void Start()
16     {
17     }
18
19
20     // Update is called once per frame
21     void Update()
22     {
23         animator.SetFloat("horizontal", Mathf.Abs(Input.GetAxis("Horizontal")));
24         animator.SetFloat("vertical", Input.GetAxis("Vertical"));
25     }
26 }
```

My PlayerController, incase of emergency C# script, that will have the same code as the CharacterController script, and it will be used if the CharacterController doesn't work.



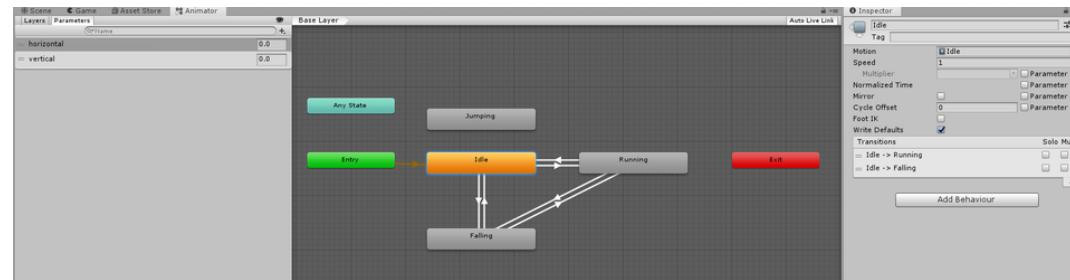
The changes and additional tiles i added to my scene, so i can test the Player and the Enemies to the limit.

28/02/2020: I continued where i left off and fully completed the video tutorial i was watching, by fully completing all of the transactions of the states, back and forward. I did this so that my Player can animate when i do something and then go back to another animation once the Player has a certain aspect in the game like the horizontal float becomes less then 0.3, so after the Player does the run animation it can always go back to the idle animation, this is the same with all the other state animations.

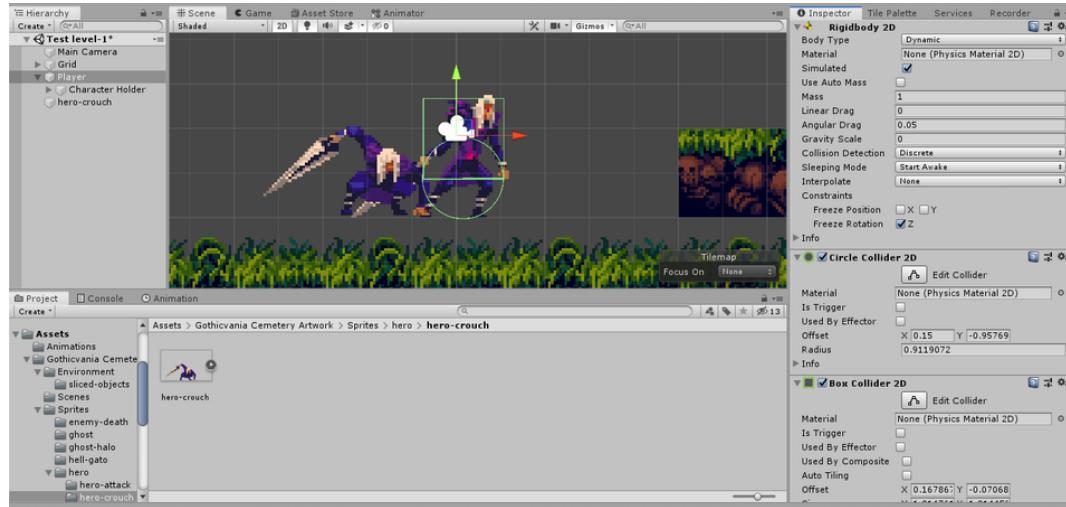
Later i started watching another video called "Super Mario - Running Tutorial in Unity", this video explains to you how to make your Player run with code and a Rigidbody 2D and a custom Edge collider 2D, the reason the person did this is because if you use a capsule collider there can be a error with tile maps where the Player will, unintentionally hop or do a small jump, which is not good. I personally will not be doing that because my levels don't all have smooth horizontal surfaces, it has some inclinations and downhills as well, so having a box or rectangle as a collider might not be a good idea. I will be using a box collider on the top half of my Player and a circle collider on the bottom half of my Player.

I will also not be using this code in this video, but of another video after i watch that video called "A Perfect Jump in Unity - A Complete Guide", this video has all of the code that i'm going to use, from the previous video and more all completed. This video even has a like where i can copy and paste the code saving me alot of time.

But i did do something that the person in the video "Super Mario - Running Tutorial in Unity" did and that was, adding components to the Player which are Rigidbody 2D and colliders, i added one circle and a box collider to my Player, i even placed a crouched variant of the Player next to a normal idle Player to see if the circle collider was at the right size, matching the size of the crouched Player. I also freezed the Z position of the Rigidbody 2D and made the Gravity Scale 0 so when i test the Player's movement, the environment won't impede me from testing the Player.



The Animator and all the states that all have their transactions done correctly.



The components i added to my Player and the crouched Player next to the idle Player to see if the size...
of the circle collider 2D fits the crouched animation, and it does in my opinion.

01/03/2020: Later i added the colliders and the Rigidbody 2D to my Player, i finally added a new script called "Player" to my gameobject Player and inside it is the code that i copied and pasted from the line of the video "A Perfect Jump in Unity - A Complete Guide". I basically just did that and then play tested my game and as i thought, there was a few problem, it would be too good to be that easy.

First there was a problem the ray casts and i need to adjust them because they weren't at the centre of the circle collider which is where i would have preferred it was there, but it was at the centre of the Player itself, but it was alright, i fixed the problem by positioning the ray casts properly. I could have made a empty gameobject and made the ray casts come out of there but it's alright everything worked in the end, it checks if the player is on the ground and all.

The Players movement was also not well, it was very slow and very slippery, so i looked at the inspector and did some changes and testing i basically increased Move Speed to 200 and also the Linear Drag to 65 so it would not become slippery when i tried to move the Player, with

those modifications the Player now moves more natural and smoothly, i'll make more adjustments if necessary in order to better the player experience.

Another problem i had was there was an error on the console that says: "Parameter 'onGround' does not exist.
UnityEngine.Animator.SetBool(String, Boolean)" Which i didn't fully understand what it meant, so i went to the video i was watching and rewatched it and even read the comments to find a solutions.

That's when i found a comment of a person having the same problem and the person that made the video offered a solution, i googled what a animation parameter was and realized that i had to make a new parameter in the animation tab in unity called "onGround".

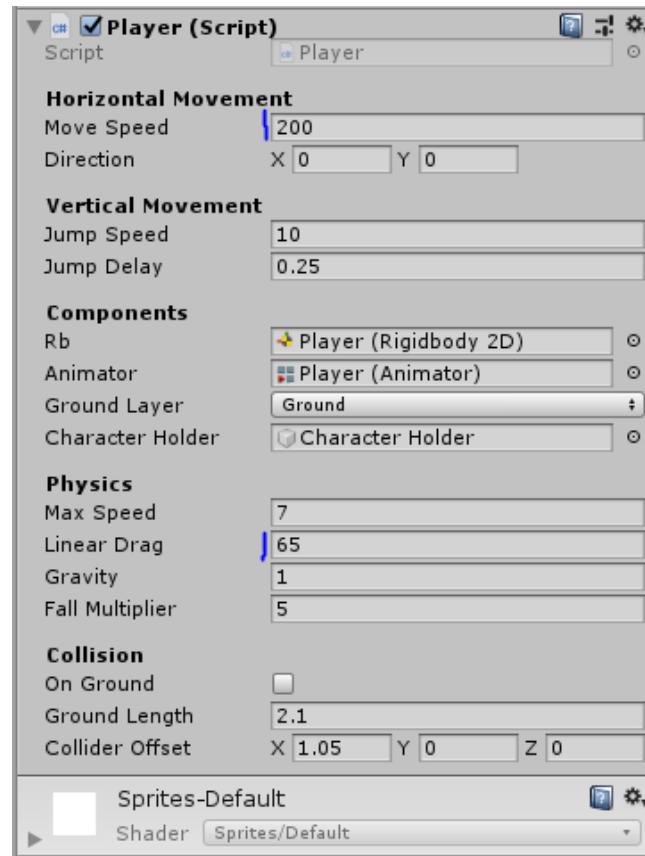
So i did that and now the warning of the onGround doesn't exist any more, which is great. But i still can't jump in game even after i added a onGround parameter and a Ground Layer to my Ground tile, I even tested my Game so that i can see how the Player moves and such.

I was messing around with my Unity game, in order to make my player able to jump, I even added a rectangle with a box collider to see if the Player can jump, and it couldn't, it just went through it to the floor. I then realized that it's because the Z axis was off and that a box collider can not collide with 2D colliders, so i changed the platform, aligning the Z axis and the collider to be box 2D collider, in order for the player to have a smooth straight horizontal line, so that Player can move smoothly without any bumps or things like that.

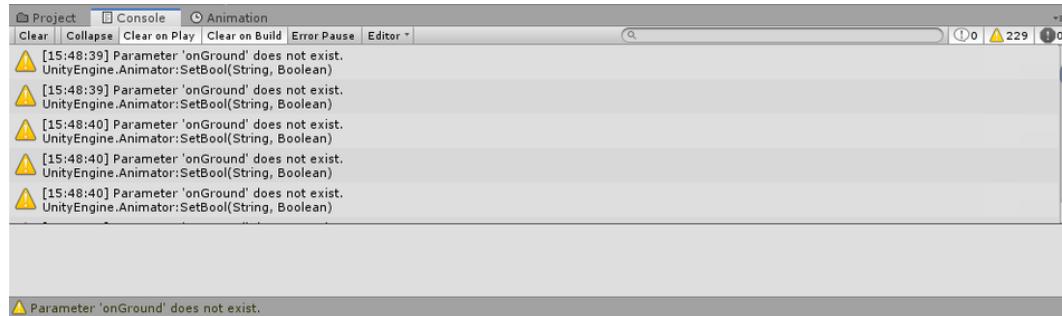
The problems my Player had was multiple things, it had a thing where it can't jump, another problem was that for some reason the raycasts acted like some sort of collider, when it should be sort of a trigger instead, when i pressed down or something equivalent the Player would do the falling animation but it should do that automatically when it is falling, because of the vertical axis being negative but it's not doing that, i have to press the down button for the falling animation to work. There is also much more but those are the main ones on the top of my head.

After some tinkering with the inspector window of the Player of the Rigidbody 2D and the Player script component, somehow the Player was now able to jump, the problem is i don't even know how i fixed the problem in the first place, it might have been the angle drag or the jump speed but i'm not sure but what matters is that my player can now jump, even if it's a weak small jump.

After that i made some modifications to the players inspector window, mainly the jump Speed, Move Speed, Linear Drag and Fall Multiplier. This was the test i did to see how my Player moves and jumps with the values i like that gave me a "good" feeling for the Player and its movement.



The Player script i added to my Player gameobject and the changes i made to the script in the inspector, so that the Player can move properly.



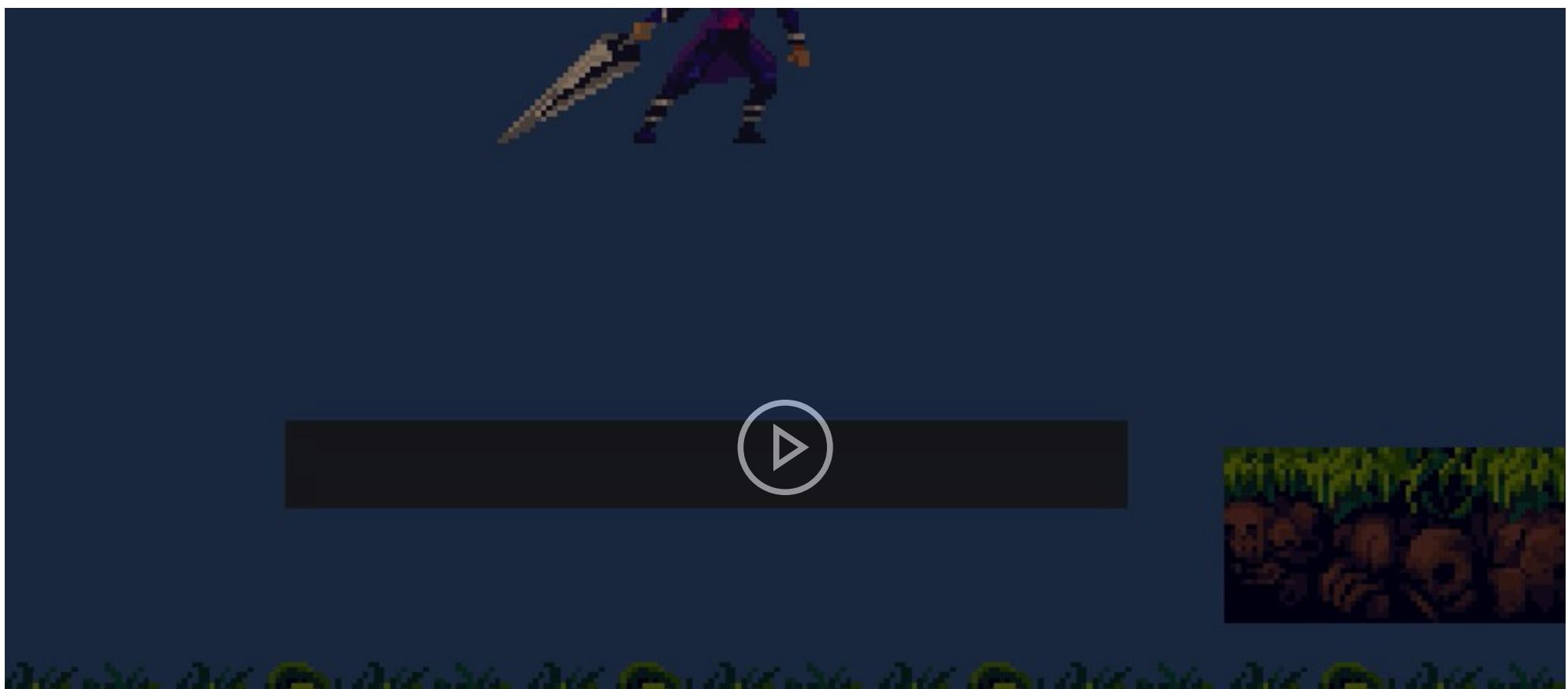
The error warning i kept receiving, when i was in play mode, when i copy and pasted the code to my Player script to my Player object.

Saffron Há 4 meses
'Parameter onGround does not exist'
RESPOSTER

▲ Ocultar resposta

Press Start Há 3 meses
Create an animation parameter called onGround
RESPOSTER

The comments i found on the "A Perfect Jump in Unity - A Complete Guide" video, that helped me remove the warning of the onGround not existing in the console.





The test i did in my unity game, to see how the player moves and jump, and to see the problems my Player currently has.

03/03/2020: I continued where i left off, i began to mess more with the inspector of the player with it's Rigidbody 2D and the Player script component. I even began messing with the "Player" C# Script that makes the gameobject Player move, flip and jump.

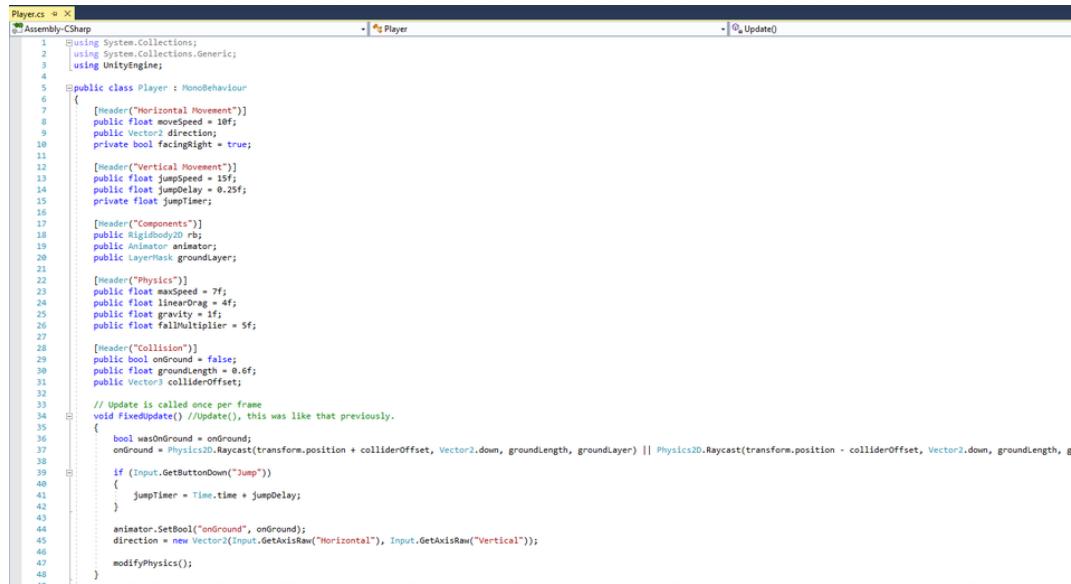
I made a simple change in the script that has made my game better in terms of movement and the falling animation, and all i did was change the void Update() to void FixedUpdate(), and the other FixedUpdate() to Update() as seen in these images to the right.

I also moved some code around and made some changes to see how that can effect the Players movement, i mainly deleting some unnecessary code like the squishing bounce effect, that the video i'm using as a tutorial has, the linear drag code and placed some code in the new FixedUpdate() and more.

I tweaked everything so that controlling the Player feel good but i, in my opinion, still have a problem i would like to fix, and that problem is, the player can get irregular jumps, what this means is when the Player is standing still and you jump, you jump normally, but when you run and then jump you do a even higher jump, which doesn't make and sense, there is no consistency, it not normal in video games to have

a higher jump when you simply run and a shorter jump if your standing still, it would make more sense to jump further not higher by simply running and jumping, so i am trying to fix this problem by doing lots testes and changes to fix this but still not fixed.

I will continue to try and fix this problem but if i can't or i have to make to many drastic changes in the code to fix this then i'll just leave it as it is, since it's not a massive game breaking problem. I'll just focus on something else like adding the crouch and the attacking mechanic with animations.



The screenshot shows a Unity code editor with the Player.cs script open. The script is a MonoBehavior class with various attributes and methods. It includes sections for Horizontal Movement, Vertical Movement, Components, Physics, and Collision. The Update() method contains logic for jumping and physics modification. The code is annotated with comments explaining its purpose.

```
Player.cs
Assembly-Sharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
    [Header("Horizontal Movement")]
    public float moveSpeed = 10f;
    public Vector2 direction;
    private bool facingRight = true;

    [Header("Vertical Movement")]
    public float jumpSpeed = 15f;
    public float jumpDelay = 0.25f;
    private float jumpTimer;

    [Header("Components")]
    public Rigidbody2D rb;
    public Animator animator;
    public LayerMask groundLayer;

    [Header("Physics")]
    public float maxSpeed = 7f;
    public float linearDrag = 4f;
    public float gravity = 1f;
    public float fallMultiplier = 5f;

    [Header("Collision")]
    public bool onGround = false;
    public float groundLength = 0.6f;
    public Vector3 colliderOffset;

    // Update is called once per frame
    void FixedUpdate() //Update(), this was like that previously.
    {
        bool wasOnGround = onGround;
        onGround = Physics2D.Raycast(transform.position + colliderOffset, Vector2.down, groundLength, groundLayer) || Physics2D.Raycast(transform.position - colliderOffset, Vector2.down, groundLength, groundLayer);

        if (Input.GetButtonDown("Jump"))
        {
            jumpTimer = Time.time + jumpDelay;
        }

        animator.SetBool("onGround", onGround);
        direction = new Vector2(Input.GetAxisRaw("Horizontal"), Input.GetAxisRaw("Vertical"));

        modifyPhysics();
    }
}
```

The code that i'm using and have copied in a link from a video, and the changes i did to it.



The test i did in my game with my Player to see how the new changes work, and the problems i have with my Player and it's movement.

13/03/2020:I started implementing the research i did to my project. First wrote two lines of code from the video "Unity3D Player Jumps Too High - inconsistent Jump Height", which is:

```
" rb.velocity = Vector3.zero; " and " rb.angularVelocity = 0f; "
```

And so i typed this in to my Player script and placed it on the FixedUpdate, tested it and it didn't work, so i moved the code to Update, didn't work. I placed it everywhere in the code and that still doesn't fix my problem.

I even did the same thing but changed the code to instead being "rb.velocity = Vector3.zero;" to be "rb.velocity = Vector2.zero;" and my problem has not been fixed yet.

When i did my test the results were practically the same, the Player couldn't jump for some reason, the Player was slow and it fell very slowly, just with some minor differences.

I then, placed every thing back to the way it was and started messing around with the other part of the code, especially the raycast mechanic, to see if the Player is grounded or not, so he can jump or not. I thought that if i simply moved the raycast code to the Update void instead of the FixedUpdate void, it would fix the problem i still have being the raycast acting similar like a collider. At first i thought the problem was fixed but with further testing, it turns out the problem is still there, i don't know if i should simply make the ray cast small enough that people won't notice or should i keep trying to fix the problem, i don't know yet.

After doing the tests, i have realized another problem with my game, my Player appears to do a falling animation when running sometimes, this usually happens after i jump and land on a tile map with a tile collider. This is quite annoying for me since it breaks the player's immersion a little bit and it doesn't make any sense to randomly do that. I believe the problem comes from the ground having a tilemap with a tile collider, having very small gaps of space where the Player seems to "fall", i believe i can fix this problem by adjusting the Animation tab of the falling animation to need a higher value or "speed" to activate, meaning the Player needs more height to activate the falling animation, so he isn't easily falling randomly when running normally.

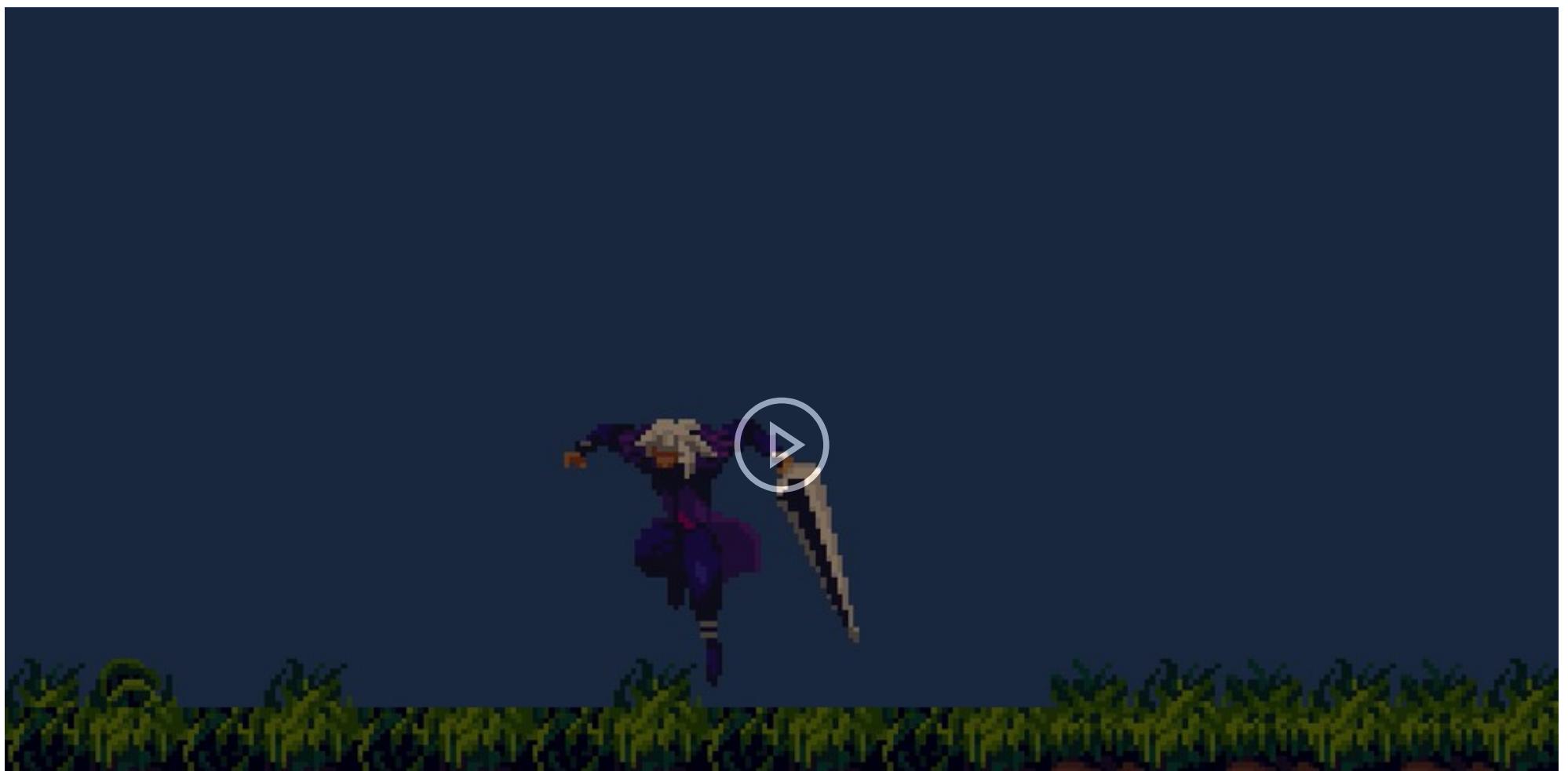




Testing to see if the two lines of code i added makes the player jump better.



Testing to see if the two lines of code i added makes the player jump better, but with some differences in the code. The results are a bit different.





The test about the little error or problems the Players gets when moving after it jumps.

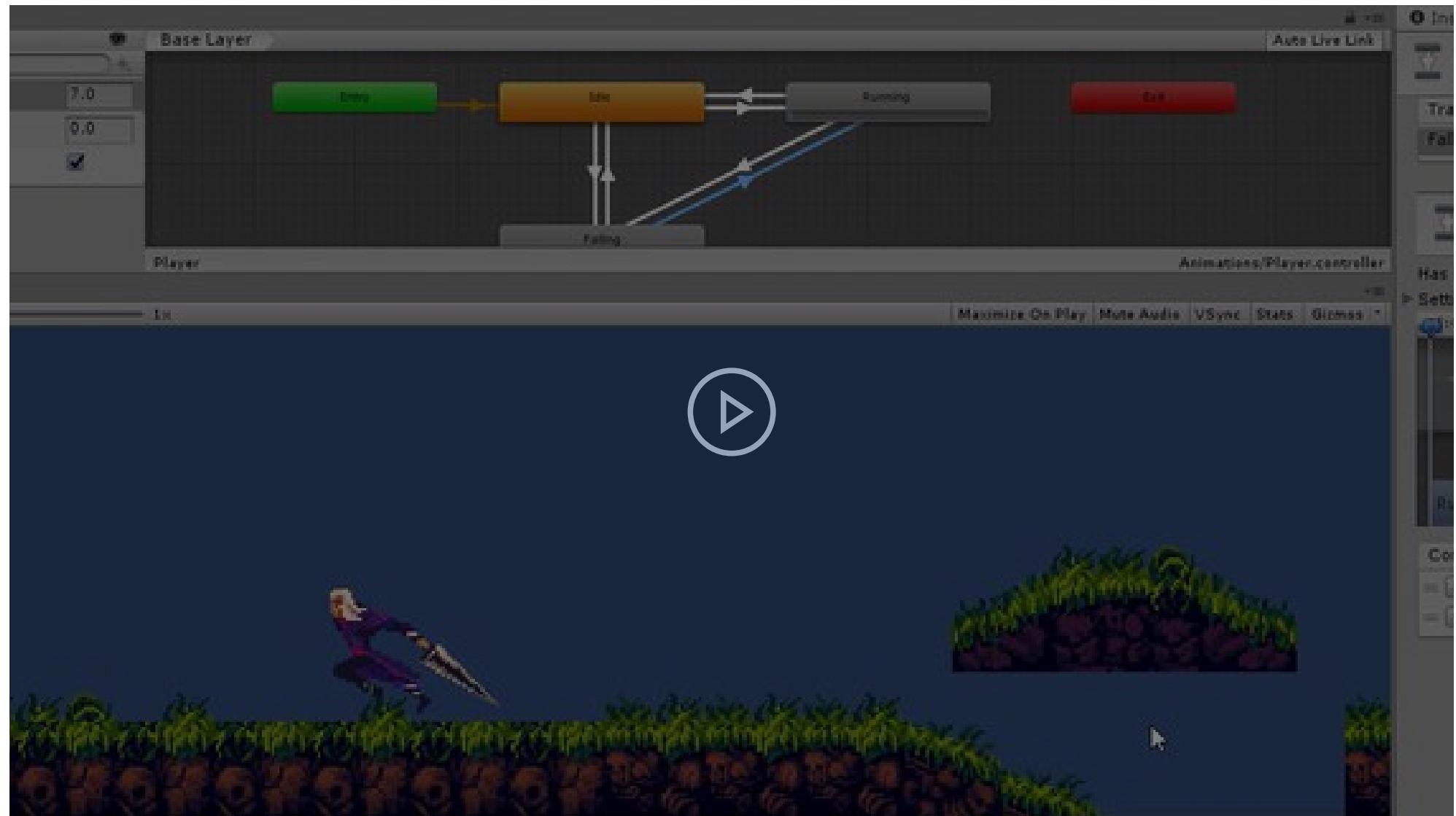
16/03/2020: What i basically did was try to fix the problem my Player had with randomly doing the Falling animation, the reason for this, i believe, is because of the vertical movement having a sudden dip below negative values because of the tilemap collider 2D having very small gaps, maybe, making the Player move weirdly.

I did a small video show the effects it has to my Player, how my Player looks like when that happens, the change in values that causes the Player to do that and the small window of that that makes the Player not be able to jump because of that.

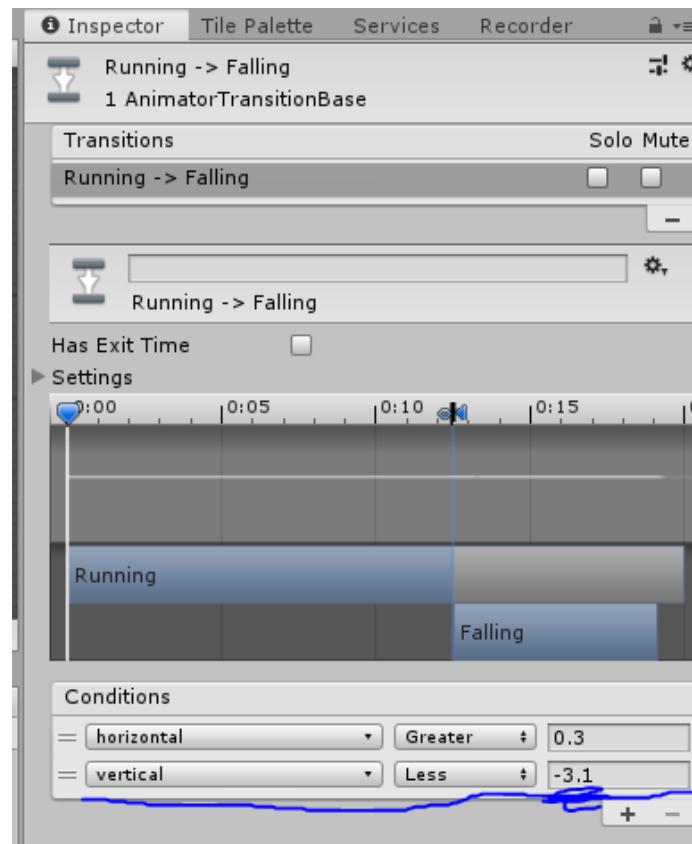
I mostly fixed this problem by basically increasing the values in the Animator tab inside the transitions, making it harder for the Player to randomly do a Falling animations because it now requires more time for the Player to be actually falling then just a small change in height, like a minuscule drop that should effect the Player that much.

The to find the right value i had to do a lot of trial and error to see what fits right with the Player, i found that the value -3.1 doesn't make the Player do the Falling animation when running and it doesn't impact how the Player looks in the air, i even increased the vertical value on

the transitions between the Idle and Falling animations just to be sure the problem doesn't appear. But i will change the value to be higher if the problem still arises.



Video on the problem i have with the Player randomly falling because of the y axis or vertical movement.



The change i did to the Condition inside the transition from
Running to Falling, but also to others as well.

19/03/2020: After doing some more testes and small changes, to the two lines of code, about the jump mechanic and trying to fix it, i began looking into the other research i did about inconsistent jump, see if those that fix my problem i have with the Player.

I went to this link i had in the Ongoing research: (<https://forum.unity.com/threads/solved-inconsistent-jumping-height.515369/>)

I began reading the forum to see if any of this can be applied to my game, then i looked into by Player script, i tried to do some thing after reading the forum, and that was add " void Start() { rb = GetComponent<Rigidbody2D>(); }, but that didn't do anything. I then started doing other things, change a few numbers, take some code out and nothing works, i even went to another Unity forums to see if they have any solutions that might work of my game: (<https://forum.unity.com/threads/jumping-with-addforce-is-inconsistent.298813/>), but there was none.

But then i stated looking into the Linear Drag of my game inside the script, previously i have found out that the cause of my Player jumping higher when running is because of how the Player no longer has drag on it, making the jump stronger then if the Player was standing still, since i i'm pretty sure i know the cause of the problem i began looking more in depth on the linear drag and all the codes that have it in my Player script.

I started messing around it in the script, add a " linearDrag = 0f; " to different parts of the code and see what happens, what i found out was that if i add " linearDrag = 0f; " say at the inside the bottom of the void FixedUpdate() or mostly any block of code, this would make the Player very "slippery" because now there is now drag to make the player slow down to a halt, but the interesting thing i found was that the jumping are now consistent, when you run and jump or if you are idle and you jump the jump height and the same.

The test i made that makes the jump consistent and the same height, proving that this is the right direction in solving this problem i have.

This test made me realized that i have to do something with the linear drag in order to fix this problem, so i began messing around with that simple line of code then, when i added that to the void modifyPhisics() inside the if (onGround), i replace "rb.drag = linearDrag;" with "rb.drag = 0f;", and this made a interesting result, that made the Player not have drag when running but gain it when jumping, because of that i focused more on this part of the code and started changing it, that's when i, everything was like it was, i just made one simple change and that was changing the rb.drag = 0f; to rb.drag = linearDrag; in the else statement in the if (onGround). This is when the jumping

mechanic finally was consistent, no matter if that Player was running or standing still, because of this change, now the Player had linear drag all the time, meaning that the Player's jump and running speed was slow but i could increase that in the inspector window with the options i have, and i did.

After that i did more tests to see how the Player moves now with the new change and thankfully it's not bad and it now has better movement in my opinion.

Here is the test i did to see how the Player moves now, and the improvements in it's mobility, it may look slow but i could always increase the jump and running speed.

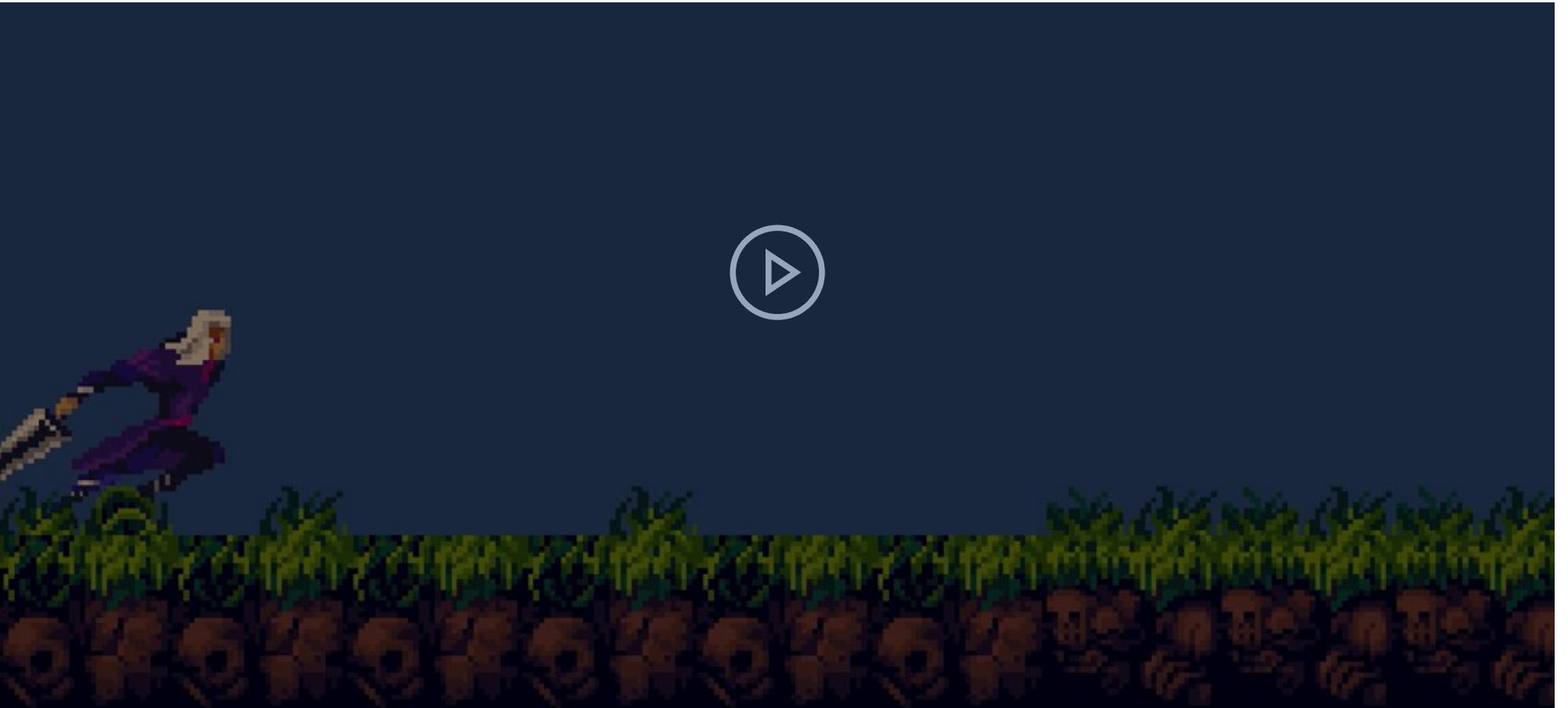
After all that i wanted to finally add a camera to my game, more specifically a cinemachine camera, i went to window > package manager and install the cinemachine package, after that i went to the new tab cinemachine and clicked on Create 2D Camera, and with that a new game object was in my hierarchy window called "CM vcam1" also know as a cinemachine virtual camera.

Then i started to make it follow the Player by dragging the Player to the Follow slot in the inspector of the CM vcam1, then i messed around with the options i have in order to make the camera look fine for the people playing my game, the image to the has the options i have and the changes i did to them.

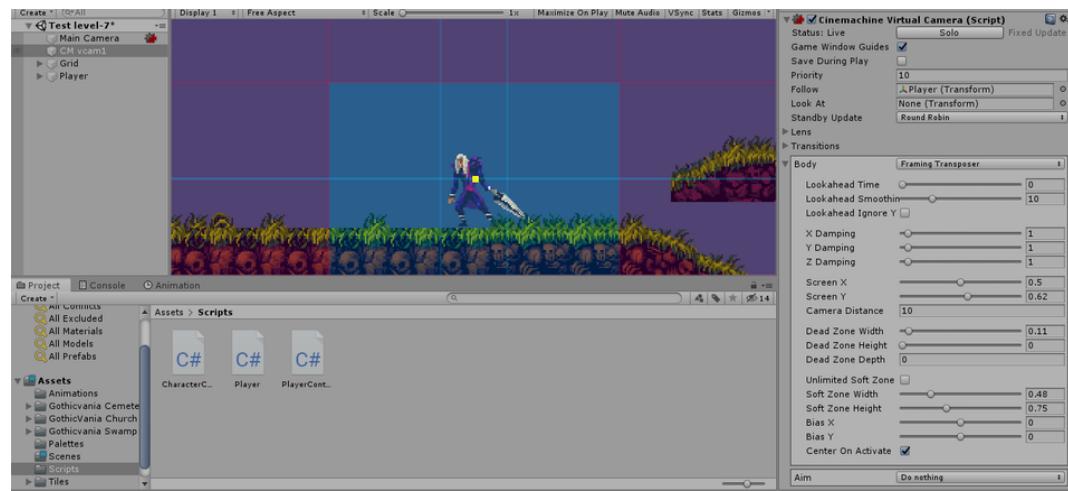
This is the test i did to see how the Camera feels and if i need and more changes or not, having a camera now is very useful and helpful in understanding how my game should work in order of people playing my game to enjoy it.

```
Player.cs  X
Assembly-CSharp  Player  modifyPhysics()
85     void modifyPhysics()
86     {
87         bool changingDirections = (direction.x > 0 && rb.velocity.x < 0) || (direction.x < 0 && rb.velocity.x > 0);
88
89         if (onGround)
90         {
91             if (Mathf.Abs(direction.x) < 0.4f || changingDirections)
92             {
93                 rb.drag = linearDrag;
94             }
95             else
96             {
97                 rb.drag = linearDrag; //Before -> rb.drag = 0f;
98                 rb.gravityScale = 0;
99             }
100        }
101        else
102        {
103            rb.gravityScale = gravity;
104            rb.drag = linearDrag * 0.15f;
105            if (rb.velocity.y < 0)
106            {
107                rb.gravityScale = gravity * fallMultiplier;
108            }
109            else if (rb.velocity.y > 0 && !Input.GetButton("Jump"))
110            {
111                rb.gravityScale = gravity * (fallMultiplier / 2);
112            }
113        }
114    }
```

The small change in the code i did to fix the inconsistent jump in my game.



How the Player's jump now moves consistently even after running making everything more natural. The Player does jump farther when running but that is relatively normal in video games and even in real life.



The cinemachine i added to my game, it following the Player and the options i have selected with a few changes and different values to my liking.

20/03/2020: After all i have been thru, i wanted to make more animations of my Player so i went to the Animation tab, i had in the bottom. With the animation states in the Animator tab now i start organizing them and then making transitions, first i start doing that to the Jumping and Jumping2 states first.

Basically on the Jumping animation i made the condition that it had to have less vertical movement at 12, i did some testes and this number fits well in making the animation work. Then on Jumping2 i made the condition that it needs to be between a vertical movement of -2 and 2 because, this jump best represents a animation that show that the Player is going from moving up to moving down, a switch from positive values to negative.

I then went and tested to see if everything works and unfortunately it didn't, here is a video showing what it was like.

For some reason, i haven't figure it out yet, the Jumping2 animation was being the default of the Player making the Player look like it was jumping even though it was on the ground, but everything else was working, the falling animation after the Player goes up and then starts falling.

I decided to look at the Animator tab again and started changing some numbers so that the Player doesn't look like it's jumping even when it's standing still, mainly increasing the values, i wasn't successful but then i thought about adding a new condition to the transition from Jumping to Idle, that is the onGround bool being true and vice-versa from Idle to Jumping, after that the Player wasn't in jumping state when it was on the ground, but i had another problem involving the Player not plying the Falling animation when it's going down. Later, after some thinking, i realized that there didn't need to be for Jumping2 to Jumping, it had to be Jumping2 to Falling because the Jumping2 didn't need to go back to a different jump when it's going down to the ground. So i made a simple transition from Jumpin2 to Falling with the condition of vertical movement being less then -2, and with that i tested my game and it works fine now.

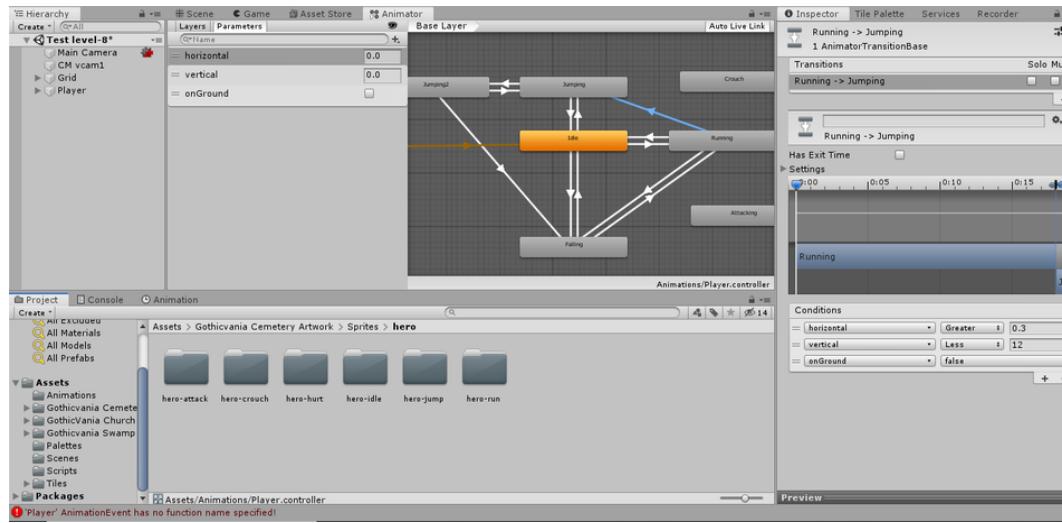
I also had some error in the console saying " 'Player' AnimationEvent has no function name specified! ", i still don't know what this mean or how to fix it but i do know that it doesn't stop me from playing my game and as far as i know, it doesn't break or change my game in the slightest, so i'll just leave it there and fix it if i have time.

Now i just have to add transitions from Jumping animation to Running so that the Player can run and jump and look like it, not just look like your still running but you simply went up.

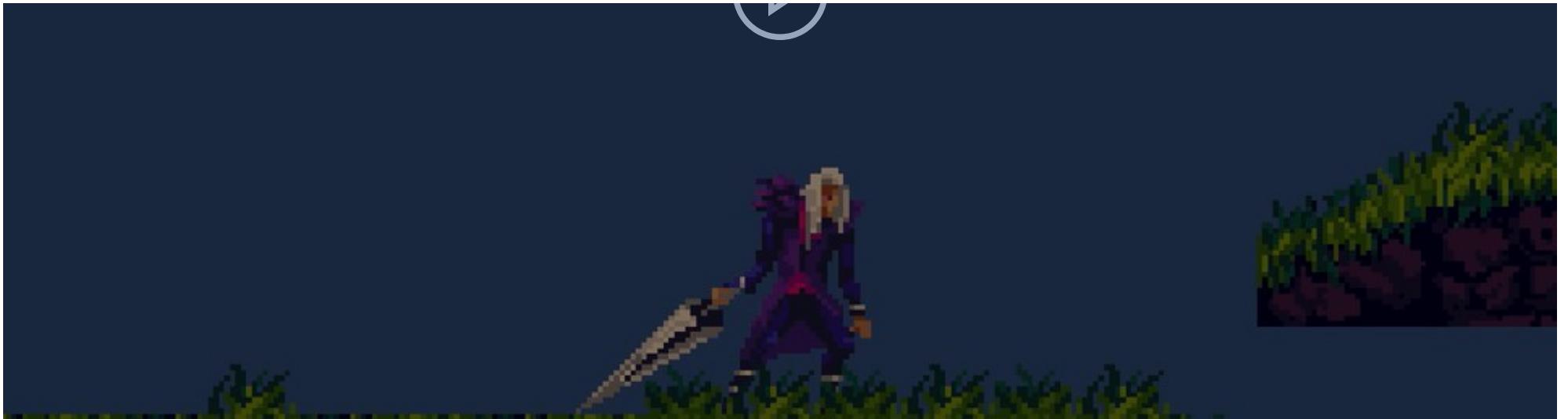
I added two transitions from Running to Jumping an vice-versa, i added conditions saying that to change states the Player needs to vertical movement less than 12 and onGround to be false and to also have a horizontal movement greater than 0.3. At first there was a problem with running, the Player was running and jumping at the same time, like shifting from one and another really fast for some reason, i solved this problem by simply taking out the transition from the Jumping to Running, which solved the problem, which makes sense since the upward jumping wasn't going back to Running.

I then tested to see if everything works as i wanted it to and i'm happy it does. I then started doing more tests in order to get the Player to look better when moving, after doing some adjustments in the Animator tab, mainly increasing or decreasing a few values.

Even though i like how my Player moves and looks, there are still some minor problems, like the Player doing a jumping or falling animation when uphill and downhill respectfully, that aren't game braking just a little annoying in my opinion, so i won't solve those problems right away but near the end if i have done everything i'll solve this small problem because this isn't really high in my priority list, but the animation i have now are fine as it is.



What my Animator looks like after i made all the changes and made everything organize.



The animations that i have now, with some minor issues.

23/03/2020: First i started following a video tutorial about 2D melee combat called "HOW TO MAKE 2D MELEE COMBAT - EASY UNITY TUTORIAL" so i can know how to make my Player attack and do damage to the enemies.

Then i created a enemy so that my Player can attack it, then i created two C# script called "Enemy" for my enemies and "PlayerAttack" which is the attack mechanic that my Player will have. After watching the video i decide to basically copy the attack mechanic code and put it in on my PlayerAttack Script, everything, but there might be some changes since my game will be different then what the video tutorial has, and i will be using the Physics2D.OverlapBoxAll basically the boxraycast because i think it's better for my game personally and i can change the hit box of attack more accurately.

Here is the code that i have copied from the video i'm watching, the code you can't see has:

Physics2D.OverlapBoxAll(attackPos.position, new Vector2(attackRangeX, attackRangeY), 0, whatIsEnemies); which is the area of effect that will damage any enemies in the area.

The other part of the code i copied.

Originally i had a OverLapCircleAll but i decide to change it because i think it will be better for my game. I also changed what i had in the Input, previously i had, Input.GetKey(KeyCode.Space) but then after doing some research on GetButton and GetKey i decided to use Input.GetButtonDown as it's better for my game and it's more reliable.

After that i copied the code for the Enemy script of the video to my own Enemy C# script, but with some changes since not all of my enemies will be the same, i think that i will use this for a basic stat that the enemies will have or it will be a specific script for a enemy i have that will simply be idle and do nothing i'm still not sure, if i choose the basic stat i'll need to do some changes to it so that will be useful for all enemies which i think i can do, it won't be easy but i can do it.

Later i went to my Unity game and added the Scripts to their designated GameObjects, and did some modifications in the inspector window, mainly change the Enemy's speed to 0 and change the Players stats to what i like, for example, the Players attack interval is 0.3, the Players Damage is 40 and what the Player can attack the what the boxraycast will be at, it's the AttackPos.

I also made a Enemy Idle so that my Player can test if the attack works or not, to see if the enemy can be destroyed or not.

Currently the animation idle for the enemy doesn't work, yet, i will have to change some things in order for a person to see a enemy in game.

I also have a small error on the console that says something about the " 'Player' AnimationEvent has no function name specified! ", i even went and did research and i still don't know how to solve this little inconvenience, but that's all right if i don't solve this problem because it's not causing any problems to me or my game, so if i can't fix this i'll leave it just as it is.

And even though i am having a problem with the Enemy being invisible, the Player can still do damage to it and destroy it, so that is good to know.

The screenshot shows a Unity code editor window with the tab bar at the top containing "Enemy.cs", "PlayerAttack.cs", "CharacterController.cs", and "Player.cs". The "PlayerAttack.cs" tab is active. Below the tabs, the assembly dropdown shows "Assembly-CSharp". The code editor displays the "PlayerAttack" class definition:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerAttack : MonoBehaviour
6  {
7      private float timeBtwAttack;
8      public float startTimeBtwAttack;
9
10     public Transform attackPos;
11     public LayerMask whatIsEnemies;
12     public float attackRangeX;
13     public float attackRangeY;
14     public int damage;
15
16     // Update is called once per frame
17     void update()
18     {
19         if(timeBtwAttack <= 0)
20         {
21             // Then you can attack
22             if (Input.GetButtonDown("Fire1"))
23             {
24                 Collider2D[] enemiesToDamage = Physics2D.OverlapBoxAll(attackPos.position, new Vector2(attackRangeX, attackL
25                 for (int i = 0; i < enemiesToDamage.Length; i++)
26                 {
27                     enemiesToDamage[i].GetComponent<Enemy>().TakeDamage(damage);
28                 }
29
30                 timeBtwAttack = startTimeBtwAttack;
31             }
32         }
33     }
34 }
```

This is the top part of the code I copied for a video I was watching.

The screenshot shows the Unity Editor's code editor window with the tab bar at the top containing "Assembly-CSharp", "PlayerAttack.cs", "CharacterController.cs", and "Player.cs". The "PlayerAttack.cs" tab is active. The code in the editor is as follows:

```
11 public LayerMask whatIsEnemies;
12 public float attackRangeX;
13 public float attackRangeY;
14 public int damage;
15
16 // Update is called once per frame
17 void Update()
18 {
19     if(timeBtwAttack <= 0)
20     {
21         // Then you can attack
22         if (Input.GetButtonDown("Fire1"))
23         {
24             Collider2D[] enemiesToDamage = Physics2D.OverlapBoxAll(attackPos.position, new Vector2(attackRangeX, attackRangeY), 1);
25             for (int i = 0; i < enemiesToDamage.Length; i++)
26             {
27                 enemiesToDamage[i].GetComponent<Enemy>().TakeDamage(damage);
28             }
29
30             timeBtwAttack = startTimeBtwAttack;
31         }
32     }
33     else
34     {
35         timeBtwAttack -= Time.deltaTime;
36     }
37 }
38
39 void OnDrawGizmosSelected()
40 {
41     Gizmos.color = Color.red;
42     Gizmos.DrawCube(attackPos.position, new Vector3(attackRangeX, attackRangeY, 1));
43 }
```

The other bottom part of the code i copied from a video.

25/03/2020: I resumed my what i was working on and tried to solve the problems i was having, but first i went to Edit > Project Settings > Physics 2D and unchecked the box that Enemy and Enemy connect to each other, what this will do is, it will make use that gameObjects of that type won't collider with each other. This is great for my game because when enemies are patrolling a certain area they won't bump into each other and be stopped on their tracks.

Later i wanted to finally fix my Enemy idle not showing in game, so i started looking in my Unity game for problems it may have or for something i may have forgotten, and while doing so i noticed that the Enemy idle Z position was in a negative value, so i changed that to 1, played my game and now the enemy show an idle animation, just like i wanted it to. With this experience i'll have to be more careful it gameobject i create and to not overlook and at even the most insignificant thing, because one small error and it can break everything.

I also made sure that the Enemy idle's layer was set to Enemy so that the Player can actually damage the enemy, then i made a few changes to my code so that it's better and more organized. Then i play tested my game, i can see the Enemy fine and when i click the "fire1" button which i made it to be the letter K, it will deal damage to the enemy, i know this because i can see the enemy's health drop in the Enemy idle inspector and because of the Debug.Log i made to make sure it works, after pressing the "fire1" button, the attack button which is "K", after three hits the enemy disappear.

After that i wanted to make my Player be able to do a attack animation, i have already made a Attacking animation before so that state was already in the tab, so all i had to do was add transactions to the Attacking animation and from it. I made parameter which is a Trigger called "attack" and i would make so that this would connect in my PlayerAttack script to the animation, so the Attacking animation represents the attack parameter.

Then i made transitions from and to Attacking state, the transitions to Attacking had the condition of the attack trigger parameter, but from Attacking it had no condition because i believed it didn't need one, the transitions were only to idle, Running and Falling just to see if it works. I then went and entered play mode to test it and it didn't work, because it would be in a state of constantly shifting between any other state and the Attacking state.

So what i did to fix this was to delete the transitions that came from the Attacking state and then tested again to see if that fixed anything, unfortunately, it didn't, it just made it look like the Player was constantly in a Attacking position, the Player could still attack normally but be in a weird mid attack state, as showed in the video to the right.

During the test if you pressed the attack button multiple times quickly and stop, the Player will continue to attack then stop, this will last longer or sooner depending how many times you pressed the button. I thought about changing the time interval to see if that would fix that problem but there has only been a small difference in change so i decided to change the Animator a bit.

What i tried to do was make one simple transitions from the Any State cyan box to the Attacking state box with the attack condition, when i tested it, it still didn't work, at the beginning the Player would run, jump as normal but when you attack you were doing the attacking animation infinitely, i thought i could fix this problem by going to my Animations folder and uncheck loop Time box, at the problem still persisted, just now the Player won't loop forever just attack one and stay at the last frame of attack while moving and jumping.

I was a bit tired of trying to fix the attack animation problem i was having so i tried to fix another problem i was having, i looked again at the research i did about this and realized that the error " 'Player' AnimationEvent has no function name specified! " on the console appears because of the Add event button that looks like a small pencil with a plus sign on in, was in on of the Jumping frames, and since i didn't do anything to the event, there was a small error, but it was easy to fix, i just had to delete those pencils on the frame in the Animation tab, with that there was no more errors or warnings in my unity console.

27/03/2020: After watching a video called "MELEE COMBAT in Unity" i have found the solution in the video, so what i did was go to my Animator tab in Unity make a transition from Any State to the Attacking state with the attack condition, then make another transition to the Idle state but this transition not have any conditions and, most importantly, have Exit Time with the exit time being 1 second, this is unlike the other transitions i have in my game. What this means with the Exit Time is that after the Player has done the attack animation, it will go back to the idle animation and with that every other animation too.

After that i wanted to see if the Animation works, so i went into play mode in Unity, and finally my Player attacks properly with no problems whatsoever, my Player can attack from any state it is in, whether the Player is running, idle or even jumping which is amazing. Although i do have a small problem or maybe not good to see, when the Player is attacking the enemy it deals damage and kills it but the way it does it looks weird to me personally, since when you attack the "hit box"activates instantly and only last for a instant of a second, so the animation of the Player doing the "ramp" up of the attack shouldn't be the one killing the enemy, it should be when the Player actually stabs using it's weapon, this is very clear in the video to the right. This small visual thing isn't bad, just annoying for me, but i think i have a

solution to this, and that is by delaying the attack mechanic by a fraction of a second, or even for a frame so that killing enemies looks better like in other games.

30/03/2020: I went and did more research after the failure trying to make the attack of my Player delayed, i watched a video called "#2.3 Unity RPG Tutorial - Player attack", this video is more similar in terms of how the delay needed to work like in my game. After watching the video i understood a bit more of how the Coroutine and IEnumerator, the code that involves time, work i basically just have to put the attack mechanic code inside my IEnumerator in order for everything to work. The video that i have watched has helped me a lot, without it i wouldn't known that you need to add WaitForSeconds code first then below it the mechanic of my game, signifying how important the order of the code can be in certain situations and codes, i now know i need to be careful of the order i place my code so that everything works properly.

I also made a public float variable called "delayOf Attack" which can be used to change it in play mode so i can test my delay code better and see the value that is perfect for my Player, making everything more convenient, as i can easily change the value in the inspector window in Unity.

Later i tested my game to see if the attack delay works well, and i'm happy to say that it does work fine. The enemy gets destroyed when the blade of the Player hits it, making the attack and death of the enemy more natural in video games, not only that i could also change how soon or late the delay can be by changing the values in the inspector window. There is also a small problem where if you keep attacking and moving, the Player will look like it's constantly attacking even though the Player is running and jumping all over the place, i won't try to solve this small inconsistency because i don't think i know how to solve it and it's not that much of a big deal. But in the end i'm happy that i added a attack delay to my Player it makes everything better visually in my opinion, although after doing this i believed that making a attack mechanic with box colliders instead would have been better, because with the box cast i have now, you attack one and the hit box only appears for a split second, while on the other hand, i believe that colliders can last much longer and i could make it hit the enemy only once. But i could try to make to that the box cast last longer but that may make problems appear so i'll just leave everything as it is.

01/04/2020: After making the attack delay mechanic system and animation work properly, i wanted to add the crouch mechanic and animation to my Player.

First i started following a video called "Unity 5: 2D Platformer-Crouch[13]", this video will help me make the box colliders in my Player disappear or be disable so that the Players collider and collision is more accurate when he is crouched, and possibly add the crouch animation to my Player. My game will be different to what the video has so i'll need to do some modifications to my game, it's scripts and code.

After watching the video i went to Unity's project settings, went to Input and created a new Input called "CrouchInput" with thee positive button as "s" and Alt positive button as "down", the arrow key, i did this because the person in the video did this following it makes my life and work load easier.

Later i started adding variables and code for my Player, then just when i was about to test my game, to see if the code was working, nothing is broken and if the crouching bool works or not, i got an error in my game that wouldn't let me enter play mode. So i went back to my Player script, looked at my code and realized that the problem was that the crouching == true; and the false one, was suppose to be crouching = true; with only one "=" not two of these "==" , this has made me realized that only certain code and specific situations can have the double equals sign, i believe mainly if and other statements can have that.

After realizing the problem i fixed it, which was very easy and simple to do, then i tested my game to see if there are no other errors or problem and i'm glad to say there are none, so with that i moved on.

Then i added more code to my Player C# script, mainly the Crouchfunction(): and crouch = Input.GetAxisRaw("CrouchInput"); in the void Update so that my Player and Unity game knows if i'm pressing the crouch button or not and if the Player is on the ground or not, thankfully everything works with no problems whatsoever.

Then i created a new C# script called "ColliderControl", and the video explains that this will disable and enable certain colliders depending if the Player is crouched or not.

After adding the code and such, i went and play tested my game, with the Scene window on the side next to the game window, to see if the top box collider really disappears when i press the crouch button, and it does very effectively too. The box collider also reappears when i'm no longer pressing the crouch button and everything work as it should.

I then went and added animator.SetBool("Crouch", crouching); so that i can finally add the crouching animation, which in reality is just a crouch sprite, this will make my Player's sprite and size more accurate to the collider it has.

Then i went to my Animator window and created a Bool parameter called "Crouch", so that the Animator knows if the Player is crouched or not, then i made transitions from Idle state to Crouch state, and vice-versa, and from Running state to Crouch state, and vice-versa. The conditions were simple, if the Player wanted to go into the Crouch state, the Crouch Bool needed to be true, if the Player did not want to go into the Crouch state, then it needed to be false, simple as that.

After that i went to test it and everything works perfectly, the Player now does the crouch animation when i press the crouch button with just the circle collider a=making everything accurate. The only problems i found were that the Player can still sort of move, or better slide when crouching, for example if you start running and then crouch, or stand still and start moving, or run and crouch at the same time, the Player will "slide" while in a crouch position. Another problem is that the Player can still attack i while crouch and i don't want that for my Player, even though it wouldn't affect the game that much if i left it like that, but i want my Player to be more restricted.

One thing i noticed was that my Player can turn around or flip sides when crouched, i found it very interesting and i don't mind having it in my game, i think it fits nicely into my game.

I wanted to fix the moving while crouched and still be able to attack while crouched problem, and to fix these problem i watched a video called "Scripting Crouch Animations(Side-Scroller 006)" this video shows you how to make the Player crouch, and how to make it not jump, move or even attack when crouched, which is very useful to me because i want my Player to no jump, move or attack when crouched. I tried copying a simple line of code, where if the Player is NOT crouching, then the player can move, so i did that and placed the moveCharater(direction.x); in the if statement in the void Update, i tested my game and it works fine now, although i did have to mess around with the code a bit and had to find what code could make the Player stop when crouch.

My Player could also not jump when crouched, i learned this when i was testing my Player and it's movement, now all i needed to fix was the attack animation and system not happening when the Player is crouched.

First i tried to do what i just did but with the PlayerAttack code for attack, but that didn't work because my Player script doesn't have the attack code, so i tried to do the if statement in the PlayerAttack script but that also didn't work because that script doesn't have the crouch code, so i did some other things so that i can fix this problem but nothing. So because of these problems i'm facing i thought about, doing some research on how to change code from one script to another, i found a video called "Unity 5 C# How to pass a variable from one script to another script", this video will help me change and add code that's from one script to another, after watching the video i decided to apply it to my scripts. What i basically did was change the public class name of the PlayerAttack script to be the same as Player so that it can use the codes that the Player script has, but unfortunately, even after trying to make it work the Player could still attack when crouched.

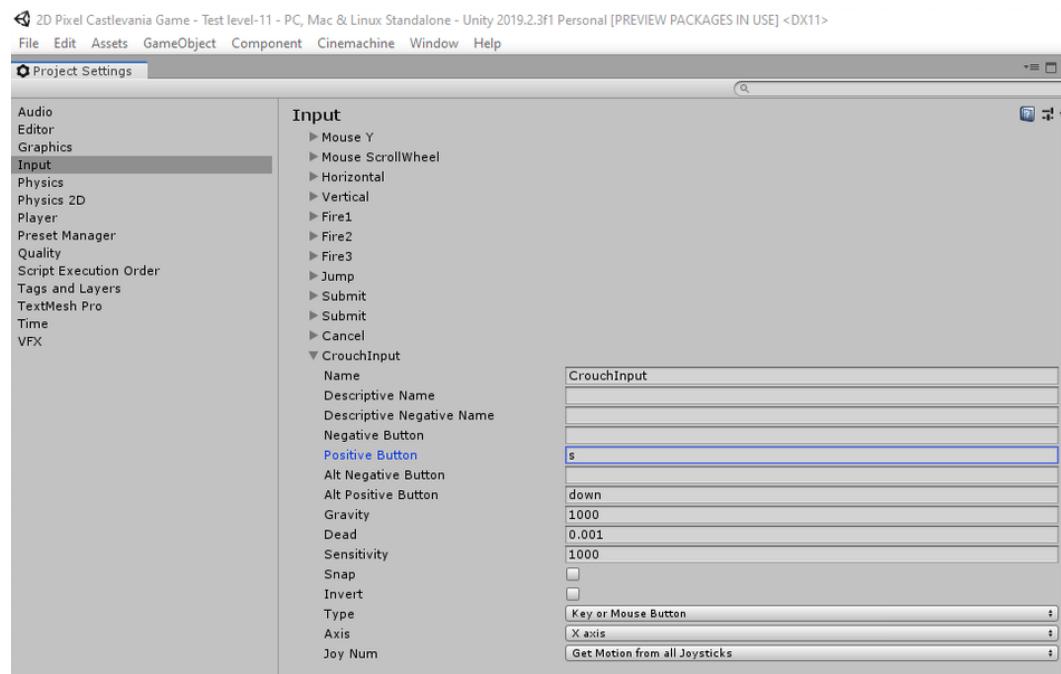
So i thought about it some more and realized that why couldn't i simply try to disable the whole PlayerAttack script when crouched, so i researched about that and found a web site of a person that wants to know how to disable a script by using another script.

Here is the link:(<https://answers.unity.com/questions/207441/how-do-i-disable-a-script-from-a-different-script.html>).

I tried adding GetComponent(PlayerAttack).enabled = false; to my if statement in my void Update in the Player script but C# wouldn't let me do that, so thought about a different way of doing it, after some trial and error i found that, GetComponent<PlayerAttack>().enabled = false; can be placed with no errors whatsoever, after that i went and tested my game and found out that it works perfectly, the Player can't attack or even do the attack animation when crouched, it's amazing, the only things was that the PlayerAttack script would be disable

forever after you crouch but i can easily fix that. The final code that works perfectly and makes the Player not attack when crouched is to the image to the right.

After that i did a test just to see how everything interacts with each other, if there are any errors or not, and after doing the test i'm glad to say there are no errors any more, now the only thing that's left to do to the Player is make the Player get hurt, be stunned by being damaged and make a hurt animation when hurt, it won't be easy i believe.



The new input i created called CrouchInput, with its own buttons.

The screenshot shows a code editor window for a C# file named Player.cs. The tab bar at the top has tabs for Enemy.cs, PlayerAttack.cs, CharacterController.cs, and Player.cs*. The Player.cs tab is active, indicated by a yellow background. The code in the editor is as follows:

```
133     void Crouchfunction()
134     {
135         if(crouch != 0 && onGround == true)
136         {
137             crouching == true;
138         }
139         else
140         {
141             crouching == false;
142         }
143     }
144
145
```

A red squiggly underline is under the assignment operator == in both the true and false assignments. The word "true" is also highlighted in red.

The error i was having that made me not be able to play my game.

The screenshot shows the same code editor window after the fix. The tab bar and file names are identical. The code is now:

```
133     void Crouchfunction()
134     {
135         if(crouch != 0 && onGround == true)
136         {
137             crouching = true;
138         }
139         else
140         {
141             crouching = false;
142         }
143     }
144
```

The red squiggly underline and highlighting from the previous screenshot are no longer present.

The fix i did so that everything works.

The screenshot shows a code editor with the file `Player.cs` open. The code is written in C# and defines a class with methods for collision detection, ground checking, and character movement. Handwritten annotations in blue highlight several parts of the code:

- A bracket on line 32 groups the declarations of `crouch` and `crouching`.
- A bracket on line 60 groups the assignment of `crouch` and the call to `Crouchfunction()`.

```
28 [Header("Collision")]
29 public bool onGround = false;
30 public float groundLength = 0.6f;
31 public Vector3 colliderOffset;
32 private float crouch; //----
33 public bool crouching; //----
34
35 // Update is called once per frame
36 void FixedUpdate() //Update(), this was like that previously.
37 {
38     bool wasOnGround = onGround;
39     onGround = Physics2D.Raycast(transform.position + colliderOffset, Vector2.down, groundLength, groundLayer) || Physi
40
41     if (Input.GetButtonDown("Jump"))
42     {
43         jumpTimer = Time.time + jumpDelay;
44     }
45
46     animator.SetBool("onGround", onGround);
47     direction = new Vector2(Input.GetAxisRaw("Horizontal"), Input.GetAxisRaw("Vertical"));
48
49     modifyPhysics();
50 }
51
52 void Update() // FixedUpdate(), this was like that previously.
53 {
54     moveCharacter(direction.x);
55     if (jumpTimer > Time.time && onGround)
56     {
57         Jump();
58     }
59
60     crouch = Input.GetAxisRaw("CrouchInput"); //----
61     Crouchfunction();
62 }
```

The two variables i created and the Crouchfunction and Input code i added to the void Update,

The screenshot shows the Unity Editor's code editor window with the tab bar at the top containing "ColliderControl.cs", "Enemy.cs", "PlayerAttack.cs", and "CharacterController.cs". The active tab is "ColliderControl.cs". The code editor displays the following C# script:

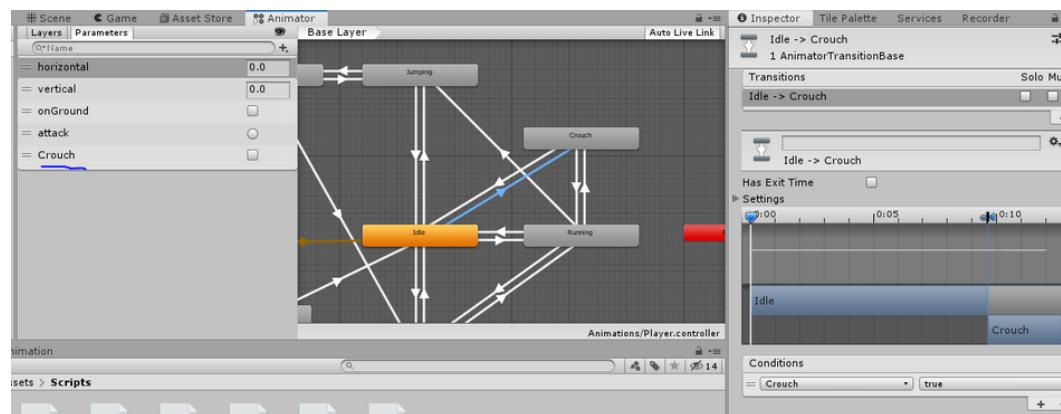
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class ColliderControl : MonoBehaviour
6  {
7      public BoxCollider2D stand;
8      public CircleCollider2D crouch;
9
10     Player playerC;
11
12     // Start is called before the first frame update
13     void Start()
14     {
15         playerC = GetComponent<Player>();
16         stand.enabled = true;
17         crouch.enabled = true;
18     }
19
20     // Update is called once per frame
21     void Update()
22     {
23         if(playerC.crouching == true)
24         {
25             stand.enabled = false;
26             crouch.enabled = true;
27         }
28         else
29         {
30             stand.enabled = true;
31             crouch.enabled = true;
32         }
33     }
34 }
```

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

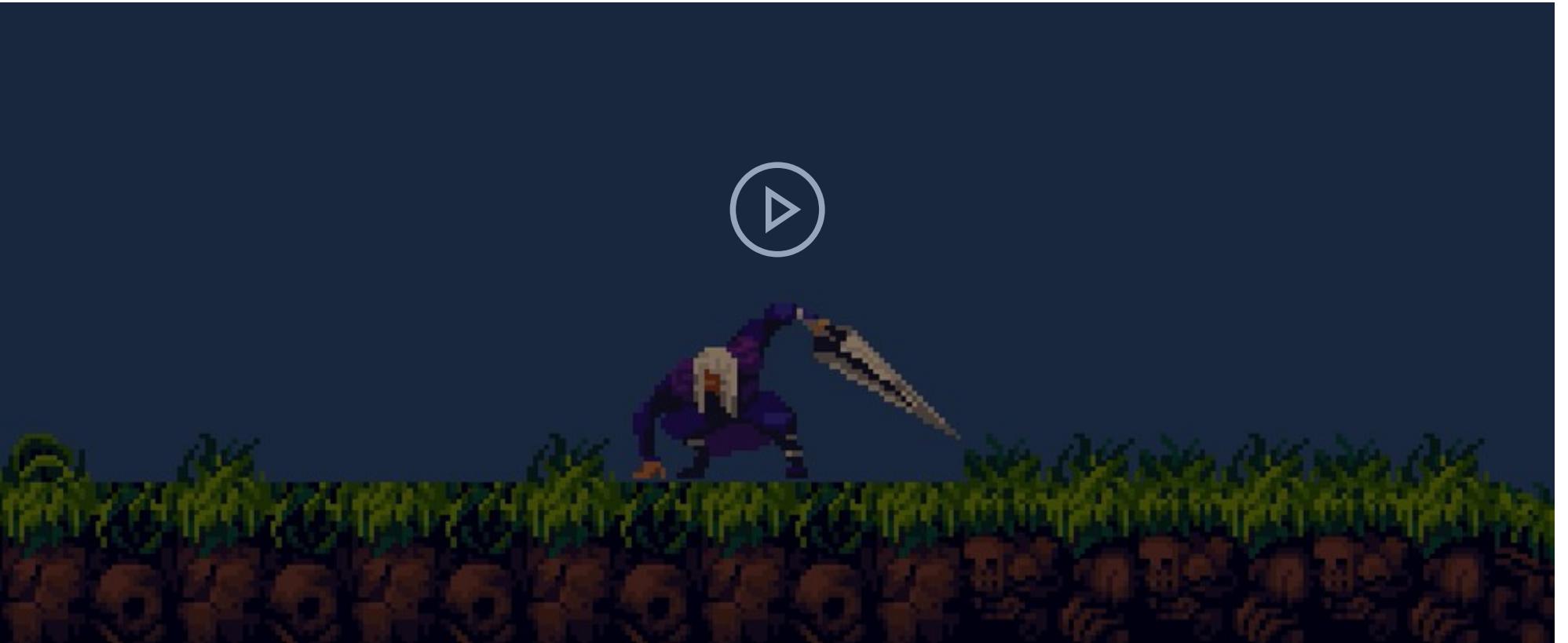
The screenshot shows the Unity Editor's code editor window. The tab bar at the top has tabs for ColliderControl.cs, Enemy.cs, PlayerAttack.cs, CharacterController.cs, Player.cs, and Assembly-CSharp. The Player.cs tab is selected. The code editor displays the following C# code:

```
34
35     // Update is called once per frame
36     void FixedUpdate() //Update(), this was like that previously.
37     {
38         bool wasOnGround = onGround;
39         onGround = Physics2D.Raycast(transform.position + colliderOffset, Vector2.down, groundLength,
40
41         if (Input.GetButtonDown("Jump"))
42         {
43             jumpTimer = Time.time + jumpDelay;
44         }
45
46         animator.SetBool("onGround", onGround);
47         direction = new Vector2(Input.GetAxisRaw("Horizontal"), Input.GetAxisRaw("Vertical"));
48
49         modifyPhysics();
50
51         animator.SetBool("Crouch", crouching);
52     }
```

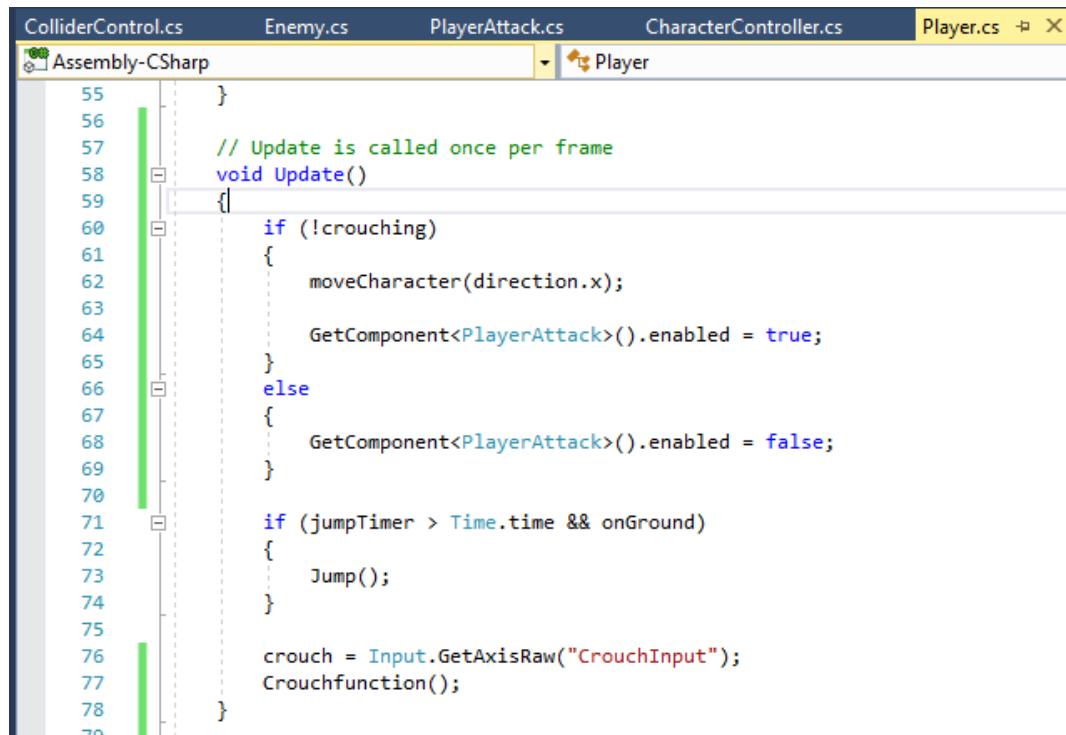
The animator.SetBool code I added to my Player script so that the Player can use the crouch animation.



The Crouch parameter I created and the transitions I added to and from the Crouch state, with its conditions and all.



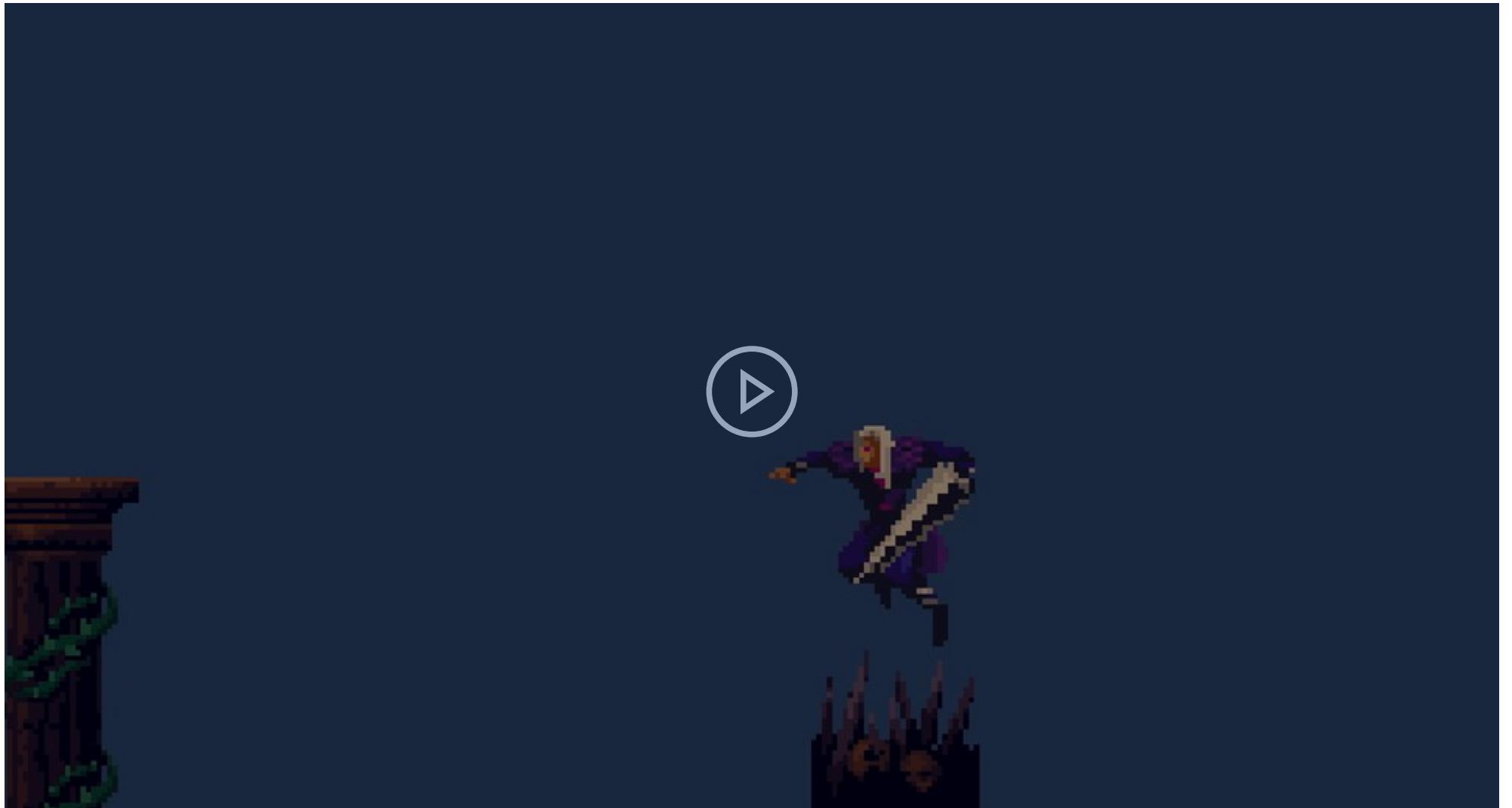
The test i did to see how the crouch mechanic works and if the crouch animation also work if i press the Crouch input.



The screenshot shows a Unity code editor with the tab bar at the top containing five tabs: ColliderControl.cs, Enemy.cs, PlayerAttack.cs, CharacterController.cs, and Player.cs. The Player.cs tab is currently selected, indicated by a yellow background. The main area displays the C# code for the Player script. The code is as follows:

```
55     }
56
57     // Update is called once per frame
58     void Update()
59     {
60         if (!crouching)
61         {
62             moveCharacter(direction.x);
63
64             GetComponent<PlayerAttack>().enabled = true;
65         }
66         else
67         {
68             GetComponent<PlayerAttack>().enabled = false;
69         }
70
71         if (jumpTimer > Time.time && onGround)
72         {
73             Jump();
74         }
75
76         crouch = Input.GetAxisRaw("CrouchInput");
77         Crouchfunction();
78     }
79
80 }
```

The change I did to my Player script in the void Update so that the Player can't slide or attack when crouched.



The test i did after adding the new code, that makes the Player not attack or move when crouched.

06/04/2020: I started watching a video on YouTube called "Knockback & Stomping Enemies - Unity 2D Platformer Tutorial - Part 11", this video will help me make my Player get a sort of knockback stun effect to my Player, this will make the gameplay of my game more exiting

and fun.

After watching the whole video i went and started adding code to my game, what i did first was add four variables of Knockback so that i can change them in the inspector window, so i can see what works well with the Player, making testing much better and easier. These value will be very useful to me.

Then i went and added the code in void Update that will make my Player jump back, like a knockback, when the Knockback Count is bigger then zero, i can test this by manually entering a number in the inspector window for the Knockback Count value slot.

This code will basically have a if statement that will make the Player be able to move if the KnockbackCount is bigger or equal to zero, else it will make the Player bounce back, left or right, depending if the KnockFromRight is true or false. This code is a little different to the code of the video i'm watching because my game is different from the game in the video, my game also has much more stuff to consider and it has more complex mechanics, making a custom code for my game hard, but the code i was given in the video is a good baseline to what my Player script needs to have.

Later, i went and tested the code in play mode and it kinda works but it does have it problems that i would like to fix before finishing the whole knockback mechanic. The test has shown me that right off the bat there are some changes to my Player, my Player now moves faster then normal and can now move when crouched, ruining what i did to my Player before.

In order for me to properly test the effects the knockback has on my Player i have to add values manually in the inspector window of the Player component, so that the Player gets knocked back, like in the video i'm watching. The knockback itself is not bad, i just have to get

the right power, length and how long the knockback should last before you can move the Player again, and for me to fix the problems i have with the Player right now, i have to change my code and i'll have to add more restrictions so that my Player won't move faster, won't be able to attack when knocked back and slide when crouched.

08/04/2020: I started changing my Player script code that involves knockback, so that the knockback is better and the problems it's has are fixed.

I tried to add another code called "modifyPhysics();", thinking that the lake of this code was the reason my Player moved very fast, but when i tested my game, everything was still the same, my Player still moved very fast, still be able to attack when knocked back and the Player even gained a new problem about the jumping mechanic, the jump was much weaker.

Then what i tried to do, was simply take the code that was in the if (knockback <= 0) statement and leave the rest as it is, like the image to the right. This had interesting results, in the test the Player could now move, jump and attack normally, but still had the problem involving the crouch, the Player could still move when crouched, and be able to still attack when in a knockback/stun state.

This is the test i did to see if the changes i made has helped my game to be better and if the problems have been solved, unfortunately, they haven't.

After the failure of the code i added to my Player script in order to fix the problems my knockback had, i decided to change the code a bit, after messing and a lot of tests, the script got what is the image to the right, i thought about the previous test i did and realized that, if having nothing in the if statement fixed the movement of the Player then all i have to do is place the crouching if statement in the knockbackCount if statement. I did that and then i went to my Unity game and tested it, and now the Player no longer slides or moves when crouched.

But there was still a problem with the Player still being able to attack when knocked back, so i thought about it and realized the solution. If nothing happens when the Player is stunned, then i'll simply have to add one line of code that makes the Player not be able to attack when knocked back, KnockbackCount higher then 0, then return to normal when the KnockbackCount is 0 or lower.

The image to the right show the change i made to my Player script in order to fix the problem my Player had.

09/04/2020: After making my Player be able to get knocked back, now all i have to do is make the Player get knocked back after touching a enemy. I was going to start continuing to follow the rest of the "Knockback & Stomping Enemies - Unity 2D Platformer Tutorial - Part 11" video but there was some scripts and codes i didn't know about, involving the HurtPlayerOnContact and HealthManager, so i thought that maybe i needed to know what happened in a previous video about health manager, so i went back to a previous video made by the same creator of the video i was watching and i stared watching a video called "Adding A Health System - Unity 2D Platformer Tutorial - Part 9", After watching this video i realized a few things about what this video shows you.

The video shows you how to make a projectile do small damage and not instantaneous kill an enemy, it al shows you how to make said projectile spin without using the Animation window, but instead with code, which can be useful to me and i'm happy that i watched the whole video instead of skipping a few minutes to the part i want to know. This video also tells you how to implement a EnemyHealthManager, a HealthManager, for the Player, and a HurtPlayerOnContact script into your game, these scripts will help you manage the enemies and Player's health and the HurtPlayerOnContact script, will make the Player receive damage by being in contact with the enemy. The video not only tells you about this but also much more, all of this is very useful information that i can use for my future projects and increasing my knowledge of code and Unity, making everything easier for the future.

After watching the whole video i went and started basically copying the codes and script of that video involving the health and hurting the Player on contact.

First i did everything in order, i created a EnemyHealthManager, added it to my Enemy idle gameObject, and copied the code that was shown in the video, with a few adjustment and changes to suit my game, since it's different to the game in the video.

Then i did an update to my Enemy script because i realized that the "public Animation deathAnimation;" with the Instantiate method code, can be used to create a animation before the enemy truly dies, like a death animation that i would like to put into my enemies so that the enemies wouldn't simply disappear after the Player destroys them.

Later i created a HealthManager C# script, this will make sure the Player can be damaged and this will also connect everything together.

After that i created a animation so that i can add it to the deathAnimation slot in order for the enemy to "explode" when it's killed by the Player.

I then tried to drag the animation to the deathAnimation slot in the Enemy script component that the Enemy idle has, but for some reason it wouldn't let me, it would not accept it. In fact it wouldn't even let me play the game, so for now i took out the public animation variable and the instantiate method so i can at least play my own game.

After struggling with that i decided to move on and create a HurtPlayerOnContact C# script, but this time since i practically have all of the necessary manager scripts that will help me, i added not only the code from the "Adding A Health System" video but also from the "Knockback & Stomping Enemies", with the knockback code as well, hopefully this will make the Player bounce back when in contact with the enemy.

So with all that i applied the scripts to it's designated gameObjects and went to testing if everything works, and unfortunately the Player didn't bounce back when it touched the Player, this had me very confused because i went back to the videos i watched to see if i did anything wrong but no, i did everything corrects and my knockback on contact with an enemy would still not work of whatever reason. I tried to do multiple things to fix this problem, like changing the if statement in the HurtPlayerOnContact, with other.tag instead of other.name, but nothing, i tried making some lines of code into notes so they wouldn't affect the script and other code, nothing. I even tried to disable some scripts components, thinking that that might fix anything, i tried adding the HurtPlayerOnContact to the Player but

that didn't help, i tried adjusting some values here and there and but that did nothing. This was stressing me out since i did everything in the videos yet i was having problems with my game and didn't know how to fix anything since everything seemed fine.

I thought about how maybe i did something to the enemy to cause this problem, so i created a empty gameObject with a box collider, the HurtPlayerOnContact and the EnemyHealthManager scripts,with the Enemy tag on it. I then went and did testes and the problem still persisted, i tried to do some changes like adding another script or taking a script or two, i added a Enemy Layer to it, took out the Enemy tag, i did all that and more and the problem was still there.

After all that i was still having trouble with the animation slot not working, so i thought about this problem and rewatched the HealthManager video and realized that i should have a GameObject variable instead of a Animation variable, because that were the instantiate will come from, so i made a GameObject variable called Enemy, replaced the deathAnimation in the instantiate with Enemy to represent where it will come from and then i dragged the enemy object in the hierarchy to the Enemy slot in the EnemyHealthManager component slot and it works, i don't know if that will do anything abut at least i know what works and what doesn't.

Next i added a `Destroy(gameObject);` code to see if even the HealthManager was working or if the collision with the HurtPlayerOnContact was the problem.

I then went and tested my Player and even after multiple contacts with my enemies, the Player was still alive, this gave me the conclusion that the main cause of my Problems must have to do something with the collision or the HurtPlayerOnContact script.

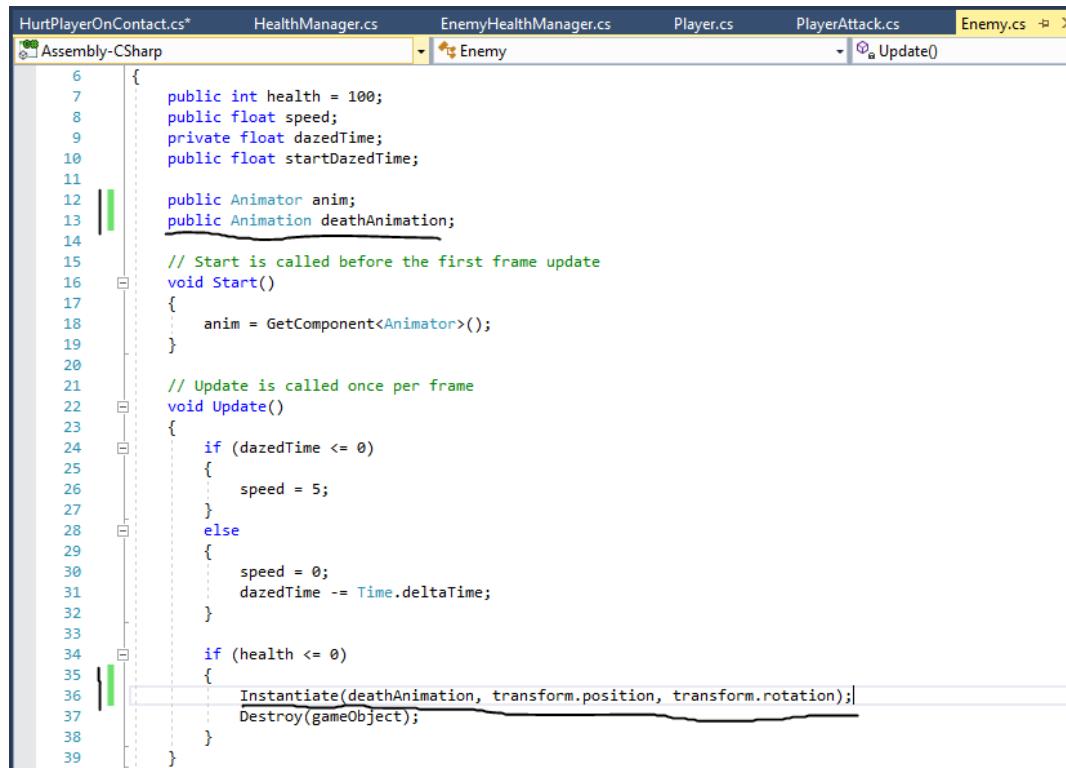
Out of desperation i went and looked at the circle collider 2D and decided to enable the Is Trigger box, i then tested my game and now everything works perfectly, the Player now gets knocked back when it touches the Enemy idle and the Player dies after touching the enemy a certain amount of times, all the problems i had are now fixed because of a simple box i ticked in the circle collider 2D of the Enemy idle object.

I even did more tests just to make sure that everything was still working, i tested the knockback and it works perfectly, i can adjust it to my liking so that it feels "right" when playing my game. During this test i also tested the HealthManager to see if it also works properly and i'm glad to say that it works fine, after touching the enemy four times the Player is destroyed, making you unable to control the Player anymore.

Later i made my empty GameObject also have Is Trigger enabled with it's box collider 2D, only with the HurtPlayerOnContact and EnemyHealthManager script, i then tested my game and this object also works perfectly. I even took out the EnemyHealthManager script and did the test again, and the Player still gets knocked back left or right depending where it's position is according to the empty object, with this i realized that the EnemyHealthManager may not be necessary at all.

Then i went to do more tests to see if i can get the knockback just right, i wanted the players to feel that the knockback they receive from enemies was fair. After some testing this is the values i got by the end of all the tests, you can see this on the image to the right.

I also did some test to see what script was necessary or useful to my Player, and i realized that my Player didn't need the HurtPlayerOnContact script or this would increase the amount of damage the Player would receive, the only thing the Player needs is the HealthManager so that the Player is killable. I then deleted the empty GameObject i had since it's no longer necessary for me to test on it.



The screenshot shows the Unity Editor's code editor window with the tab bar at the top containing "HurtPlayerOnContact.cs*", "HealthManager.cs", "EnemyHealthManager.cs", "Player.cs", "PlayerAttack.cs", "Enemy.cs", and "Update()". The "Enemy.cs" tab is active. The code in the editor is:

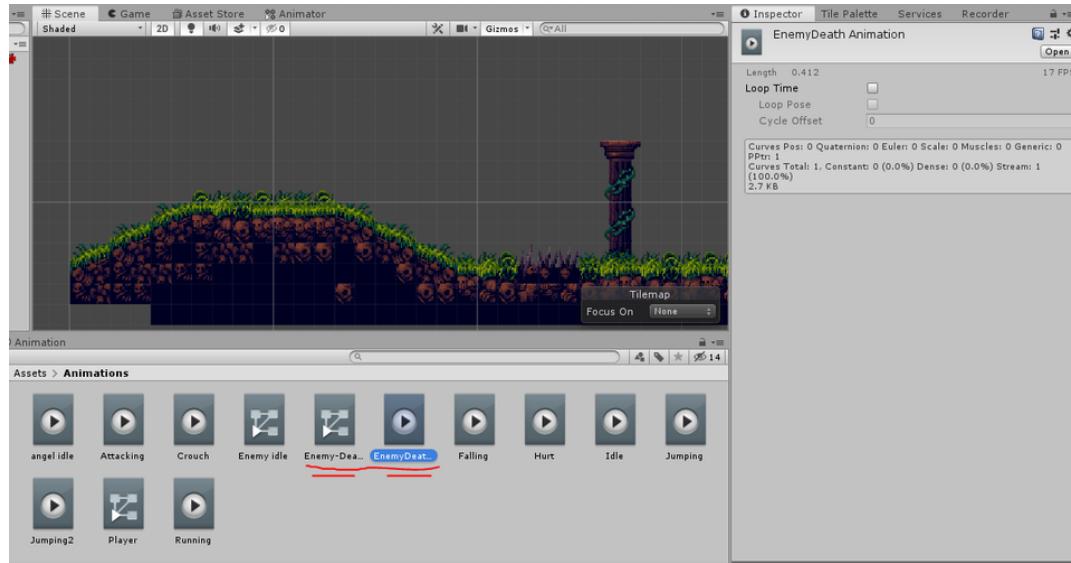
```
6  {
7      public int health = 100;
8      public float speed;
9      private float dazedTime;
10     public float startDazedTime;
11
12     public Animator anim;
13     public Animation deathAnimation;
14
15     // Start is called before the first frame update
16     void Start()
17     {
18         anim = GetComponent<Animator>();
19     }
20
21     // Update is called once per frame
22     void Update()
23     {
24         if (dazedTime <= 0)
25         {
26             speed = 5;
27         }
28         else
29         {
30             speed = 0;
31             dazedTime -= Time.deltaTime;
32         }
33
34         if (health <= 0)
35         {
36             Instantiate(deathAnimation, transform.position, transform.rotation);
37             Destroy(gameObject);
38         }
39     }
}
```

The additional code I added to my Enemy script, in order to be able to add a sort of death animation to my enemies.

The screenshot shows the Unity code editor with the tab bar at the top containing "HurtPlayerOnContact.cs", "HealthManager.cs", and "EnemyHealthManager.cs". The "HealthManager.cs" tab is active. Below the tabs is a dropdown menu showing "Assembly-CSharp" and "HealthManager". The main area displays the C# code for the "HealthManager" class:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class HealthManager : MonoBehaviour
6  {
7      public int maxPlayerHealth;
8      public static int playerHealth;
9
10     // Start is called before the first frame update
11     void Start()
12     {
13         playerHealth = maxPlayerHealth;
14     }
15
16     // Update is called once per frame
17     void Update()
18     {
19     }
20
21
22     public static void HurtPlayer(int damageToGive)
23     {
24         playerHealth -= damageToGive;
25     }
26
27     public void FullHealth()
28     {
29         playerHealth = maxPlayerHealth;
30     }
31 }
```

The code I added after creating the HealthManager C# script.



The animation i created so that the enemies could explode after being destroyed.

14/04/2020: The next thing i need to do for my Knockback system and mechanic to be complete is to add a hurt animation for my Player so that people may notice that they have been hit and knocked back.

I first start by adding a new private float variable called "hurt", thinking that i will use this in a way that will help me set up a animation like the other animation code in the Player script and in other scripts.

I then go to the Animator window and created a new Parameter called "Hurt", i then move the hurt state near the Any State, then i create a transition from the Any State to the Hurt animation, later in the inspector i disable the exit time, disabled the fixed duration and made all the values to zero so they wouldn't effect the animation. After that i created a condition, of the Hurt parameter, that has to be greater then 0.0001, this will basically represent the KnockbackCount, so if the KnockbackCount is above zero, then the Player can play the hurt animation, else it will play something else.

Then i went to my Player C# script and added a new line of code that is "animator.SetFloat("Hurt", KnockbackCount);", i knew what to add for the Hurt animation to work for my Player as i have learned from the videos i have been watching throughout my project and through doing it but in a different way, for example i made the crouching animation work by doing it this way but instead with SetFloat i used SetBool.

After i did that i realized that the private float hurt variable i created isn't necessary, i originally thought i needed some variable that would connect with a parameter in the Animator, but then i realized that that variable would be doing nothing and i needed something based on what the Player is doing, so i added KnockbackCount of what would have been the hurt variable.

Throughout the progress of this project that i'm doing i have gained some knowledge and some cool shortcuts and additional help i can get involving code.

I have realized that i can press alt + arrow keys to move single lines of code, this is helpful for me to organize my code better, faster and cleaner, this is better then copying, deleting and then pasting the code to where i want it to be.

I have realized another thing, i can press ctrl + left click a word in my script that visual studio knows about, this will open a separate tab in visual studio that will have that word, other words that are similar, what i need to add in it's brackets and more. This is useful to me because with this i can learn about a specific word, variable or anything involving code and find out what i need for the brackets since i might not know what i need to add and with this method i can figure that out very quickly without searching it up on the internet.

Later i did a test to see if i did everything right, in the script and Animator, so that my Player can properly do a Hurt animation when Knocked back. Unfortunately, after the test i realized that the animation isn't working properly as the Player is staying in a Hurt position after getting hit when in fact, the Player should look hurt but then return to a normal state, so i need to fix this problem.

I believe that the problem has something to do with the Animator window and the Hurt state, i look at it and realized that i forgot to add a transition from the Hurt state to the idle state so that the Player can go back to a normal animation after the Hurt animation is done. I add the transition and make sure that the Exit Time is on, i make the Exit time value to be 1 and everything else to be zero, it will also have no conditions.

After that i went to test my Player and it's Hurt animation, and the previous problem was fixed but now i have a new problem, the Player does a sort of weird jostle of animation where the Player, after it does the Hurt animation it will go through all of the other animations, idle, Running, Jump1 and Jump2 before it goes to the Falling animation. This doesn't look good or nice at all to look at after the Player gets hit by an enemy, so i want to try to remove this small problem that is noticeable.

I went back to the Animator and try to figure out a solution, then i thought of one i can instead move the transition from the Hurt state to the Falling state, so that after the Hurt animation is done the Player goes to a falling state which is normal after getting knocked back into the air. Then i also added a condition to that transition to act like a sort of restriction or limiter so that the Player will only do the Falling animation after doing the Hurt animation.

Then i go to my game scene and go to play mode to see if everything is fixed, and fortunately for me the small problem i had is now fixed but now i have these other small problems that i didn't know they existed until now, these problems are: if the player presses up or any jump button, the Player will be knocked back higher and slightly longer then usual and i don't know why, another problem is that is the Player is knocked back and pressing left or whatever is the opposite direction to where the knockback came from the Player will do the hurt animation and right before it does the Falling animation the Hurt animation flips, like it's turning around just before falling and it looks kinda weird in my opinion, the last problem i found, which isn't a problem just a thing a notice, is that if the player attacks an enemy and right after that gets knocked back the enemy will die while the Player does the hurt animation, which looks weird. you can see all of these problems in the video to the right.

All of these problems are minor and don't affect the gameplay at all and aren't even that noticeable to begin with, so because of that i will not be trying to fix these problems, as in my opinion they are a not in the highest of priorities, as i have other things to do and add to my game that are more important.

16/04/2020: First i dragged my background sprites i had in a folder, i changed their layers accordingly so that the moon background would be in the back, the mountains would be in front of the moon and the fields would be in front of the mountains, then i created two additional copies of the mountains and fields so that i would be easy to do a looping background. Then i created a empty object called "Background" and inside that i created three more empty objects so i can place my individual different backgrounds with it's different layers so that everything is organized. What i did was all in order thanks to a video i'm following that will help me do a infinite looping background with a parallax effect, the video i'm using is called "Unity Parallax Tutorial - How to infinite scrolling background", i believe that this video will help me make this part into a reality.

Next i started copying the code from the video i was watching into the new script i created called "Parallax", this specific code will help me make my background follow the Player in a way and also have a parallax effect, making everything visually better and nice to the eye.

In order for every looping background to work i have to add this script to the Layer -4 and Layer -3 empty objects that have my multiple backgrounds.

After i made sure that i copied everything correctly i went to testing this code in my game. The backgrounds didn't quite follow the Player as i wanted it to, and another thing the big purple background with the moon was nowhere to be seen in the game scene which is odd. Another thing i noticed in this test was that the parallax effect was active although it wasn't well showed, but it's still progress nonetheless.

Later i added more code to my Parallax script, hoping that this code will help the background loop infinitely so that the background, the mountains and fields will always been on the screen when you play my game.

When i was testing my game i kept having these warning messages as seen in the image to the right, so i looked into them and realized that these warnings were appearing because my empty object Layers didn't have a Sprite Renderer component in them, so to remove these error messages i simply added the Sprite Renderer component to my empty gameobjects with the Layers as names and now the red messages don't appear any more.

I then went and properly tested my game to see if the new code has made my background loop forever no matter if the camera is moving to the far left or right, but unfortunately that didn't happen and that error i fixed before also didn't do anything to improve my game or fixed the problems i was having, the new code i added didn't do anything to my background.

There was another problem i was having for some reason and i didn't know the reason for it, the purple moon background wasn't appearing in the game view at all, no matter where the Player was going. I wanted to fix that, so i thought about it and i tired the simplest answer, i simply moved the background itself to a better position, all of them, making it perfectly aligned with the camera rectangle box.

After that i went to test if it worked and it didn't, i'm a little surprised that it worked because i thought that the code would automatically move the background image to the game view but it turns out i was wrong.

Now that i moved the images everything looks much nicer, but there are still some problems, for example the background images only follow the camera horizontally not vertically, so if the player is very high or very low you won't be able to see the background. Not only that but the mountains and field sprites still don't loop around infinitely and they also don't follow the Player vertically.

I tried many thing to fix these problems i was having with my background but nothing was working, i tried changing the code a little bit to see if that works but nothing, i tried to modify the hierarchy and inspector window, moving the script around, disabling and re-enabling the script, to see if that does anything but nothing, i even thought about adding more code from a different video to see if that will help my issue but i decided against it.

Then i thought about my overall issue and how i was having trouble with it, after some thought i decided to delete my old code and add a new code from a different video called "Simple Parallax in Unity (with Tilemaps)" this has a much more simple code, that will make a background follow the camera horizontally and vertically, with additional options that makes a background locked on the Y axis, so it doesn't follow the Player vertically.

I then went to test the new code and now my backgrounds finally follow the Players movement perfectly, horizontally and vertically, The moon background will always be in frame and in the game view, and the parallax effect also works great with the mountains and fields, the only this is missing is those two infinitely looping in the scene. Another thing i tested was the LockY option that i gained from copying the code from the video i was watching, i could lock, in a way, the Y axis of a certain background so that it doesn't follow the camera vertically, i

did this to the mountains and fields sprite to see how they would look in game, and they don't look good in my opinion, so i would rather have all of the background follow the Player in a way then to have it look weird and less natural, i made all of the backgrounds have a different relative move value so that it will make the parallax effect work.

After all that i wanted my mountains and fields to infinitely loop so that they are always present in the game view, to do that i started watching a video called "Scrolling Repeating Background in Unity" this video will help me make my background much better in every way, not only that but this video has a link that sends me to a site with the code from the video, this is amazing because with that i can save a lot of time by simply copy and paste the code to my own script.

What i did first was make a new C# script called "backgroundLoop" and then i copy and pasted the code i got from the link below the video, but i did do some small changes to the code, for example i took out the code about moving the camera since i don't want my camera to move on it's own. Hopefully this script will make my background loop forever.

Then i went to Unity and added my script to my Main Camera, i made the choke zero, in the backgroundLoop component in the inspector, since i already made the background connect very well together so there are no gaps between backgrounds, i also increased the Levels size to two so i can add my two empty Layers gameobjects that have my mountains and field sprites, after that i made sure everything was in place, i cleaned my scripts to make sure everything is organized by going to Edit > Advanced > Format Document, i also moved the backgrounds a bit so that it will be properly placed when you play the game, i tested a little to see if this works and the background and well placed inside the game scene.

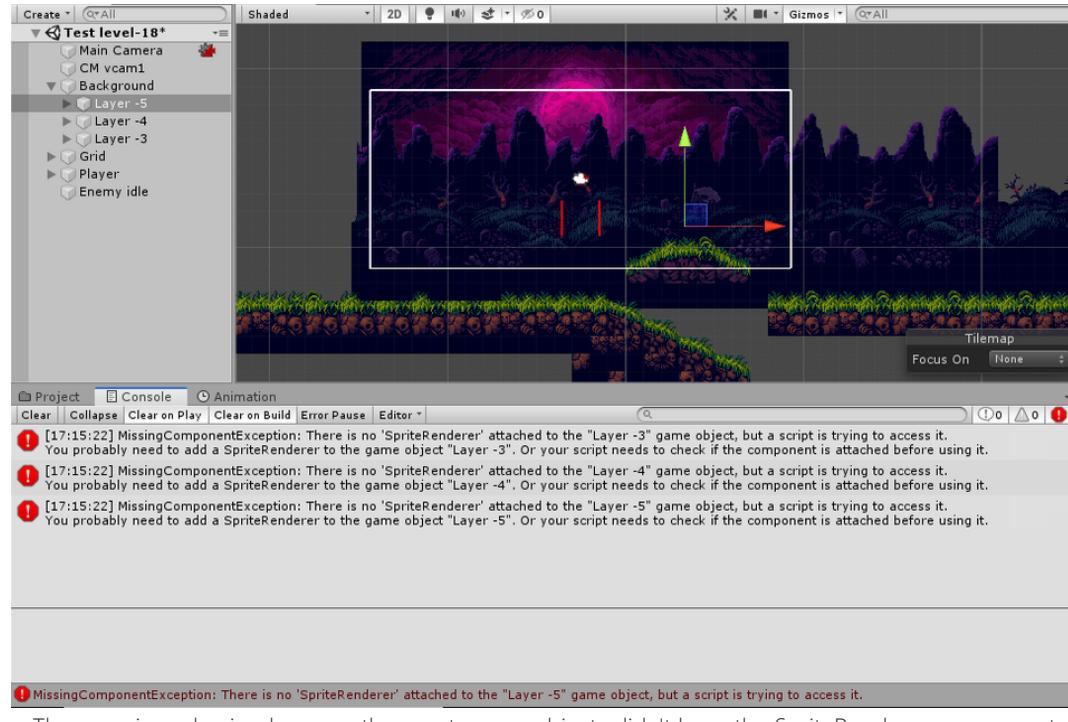
Later i went and started to test my background to see if they loop and i'm glad to say that they do loop infinitely and it looks very nice, i moved the Player around to see how everything look and i'm happy with the results, i did more tests to see if the speed of the parallax effect was ok, but i ended with the Relative Move being 0.6 with the Layer -4 empty object and 0.3 with the Layer -3 empty object, i think they look fine like that, but i could always that that very easily in the inspector.

Everything is positioned well, after doing the tests i looked at a video from what my game is based upon, i have the same art but i have different mechanics, the video that has helped me in terms to see what i need for the game to look nice and how my Player and enemies should act is "Gothicvania-Cemetery Pixel Art Game Play" but i noticed something in this video and that was that the background was blurry in some way, i thought about adding this to my game, and i even did some research on this topic, but in the end i decide to not do it just because i thought that i had more important features to add to my game, and making my game look a little bit more pretty wasn't in the highest of my priorities. I learned a lot from just adding a sort of moving background to my game, like what code works and how it works, and that testing is very important.

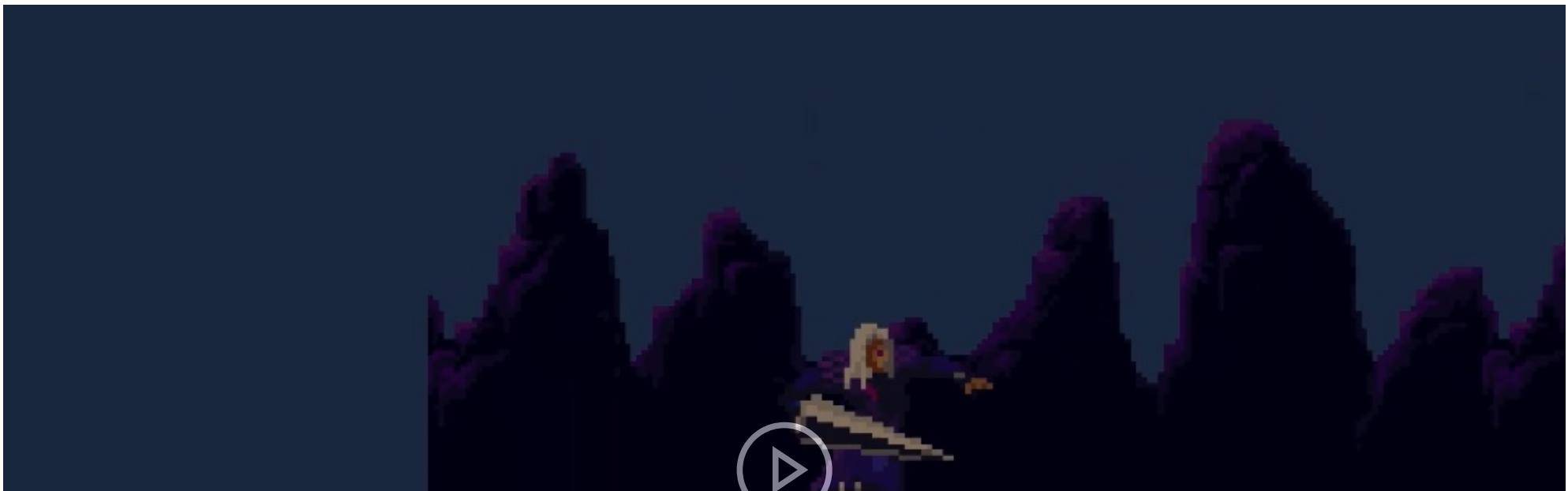
The screenshot shows the Unity Editor's code editor window. The tab bar at the top has several tabs: Parallax.cs (which is active), HurtPlayerOnContact.cs, HealthManager.cs, EnemyHealthManager.cs, Player.cs, PlayerAttack.cs, and EnemyAttack.cs. Below the tabs, there is a dropdown menu for 'Assembly-CSharp' and a search bar containing 'Parallax'. The main area displays the C# code for the Parallax script:

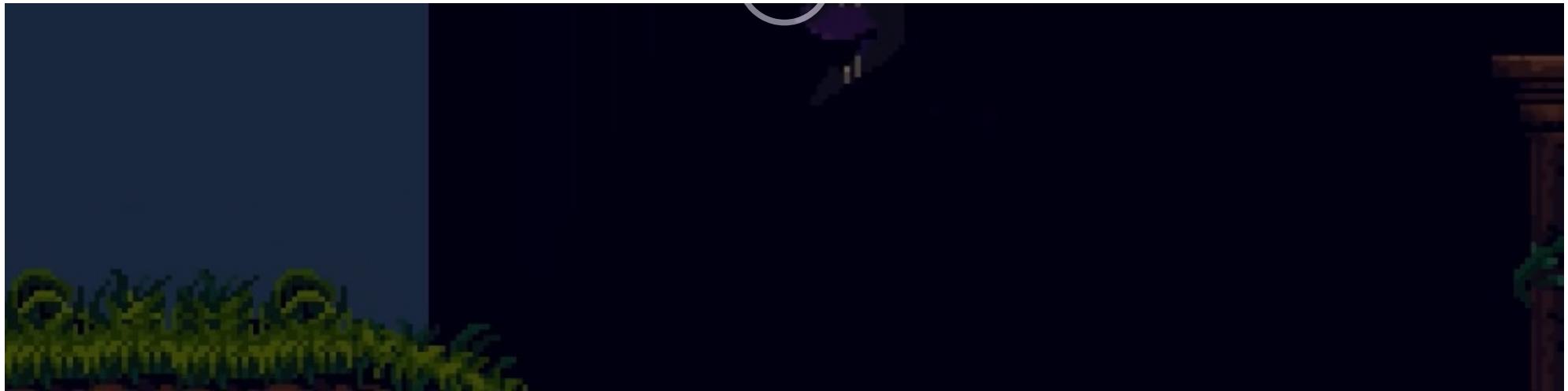
```
4
5  public class Parallax : MonoBehaviour
6  {
7      private float lenght;
8      private float startpos;
9      public GameObject vcam;
10     public float parallaxEffect;
11
12     // Start is called before the first frame update
13     void Start()
14     {
15         startpos = transform.position.x;
16         lenght = GetComponent<SpriteRenderer>().bounds.size.x;
17     }
18
19     // Update is called once per frame
20     void Update()
21     {
22         float temp = (vcam.transform.position.x * (1 - parallaxEffect));
23         float distance = (vcam.transform.position.x * parallaxEffect);
24
25         transform.position = new Vector3(startpos + distance, transform.position.y, transform.position.z);
26
27         if (temp > startpos + lenght)
28         {
29             startpos += lenght;
30         }
31         else if (temp > startpos - lenght)
32         {
33             startpos -= lenght;
34         }
35     }
36 }
```

The additional code I added to my script in order for the background sprites to loop infinitely no matter where the camera or Player is, making the images always on the game scene.

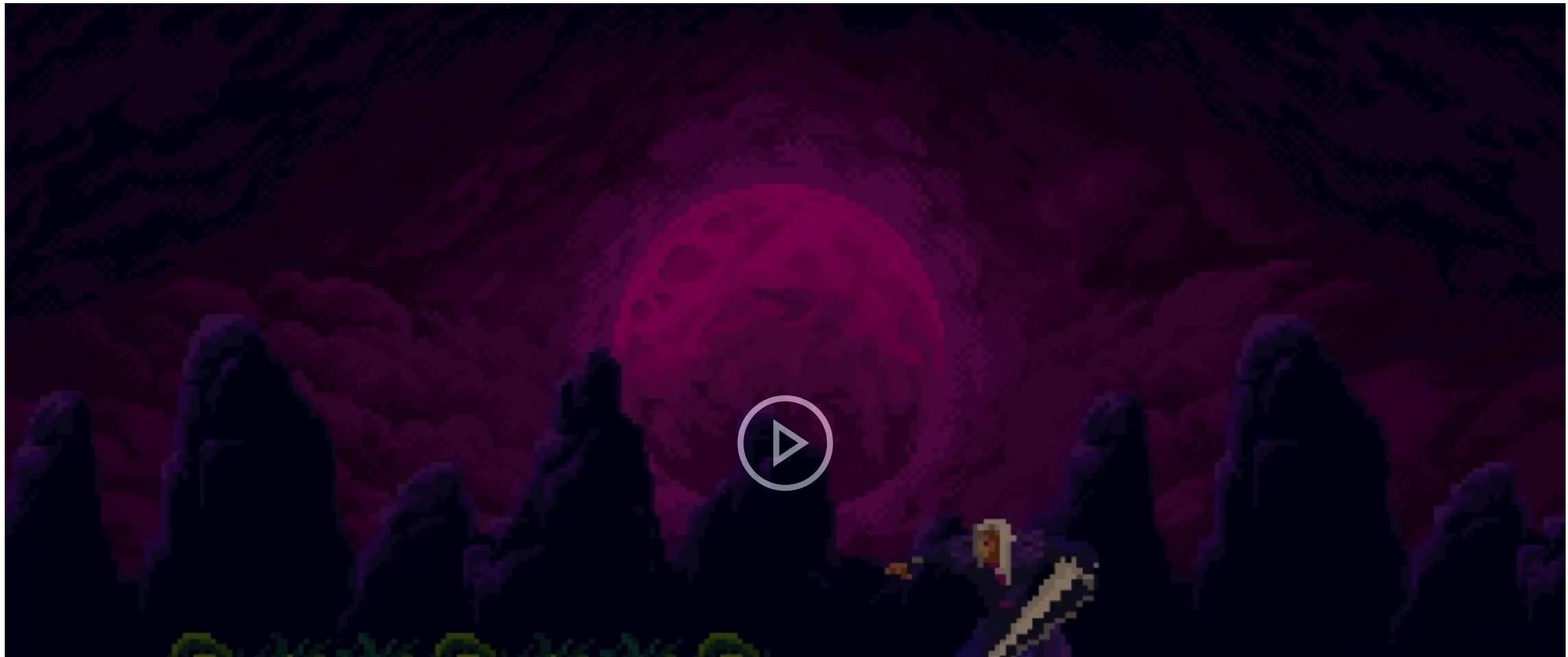
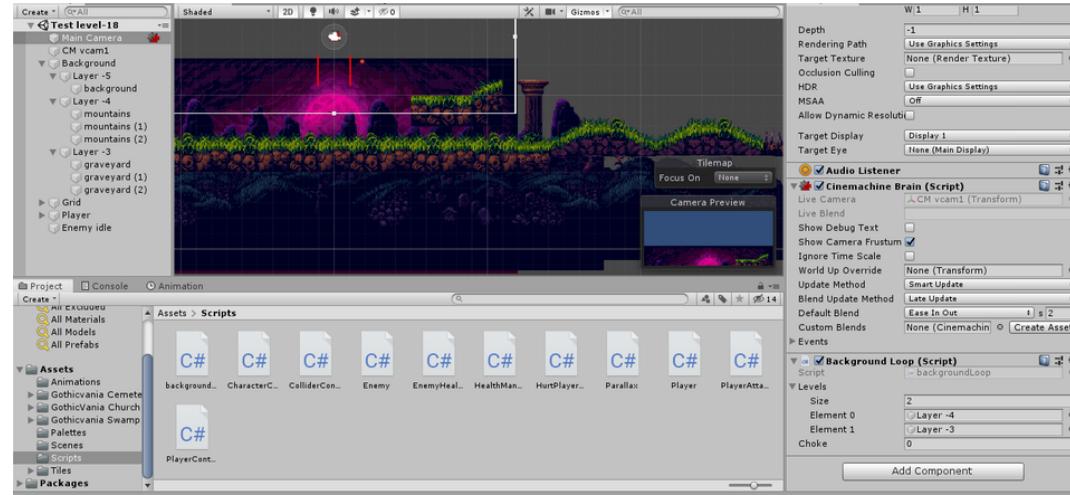


The error I was having because the empty gameobjects didn't have the SpriteRenderer component.





This video shows you how the sprite, the mountains and fields, are still doing a sort of parallax effect but not looping infinitely.





The test i did after adding the new script to my Main Camera, and the results show everything working fine and the parallax and moon image following the camera just fine.

20/04/2020: I started adding enemies and it's mechanics to move and such after doing the background parallax. the first thing i did was add three enemies by going to my enemy script, grab multiple of them and then place them into the hierarchy, by doing that i'll have created three enemy animations, which is the only animation that they have, which is moving or some short of idle animation. Later i placed the new animation i created in the Animation folder and renamed the enemies in the hierarchy and placed them in different positions in my scene.

After that i added box colliders to my new enemies, to my cat and sekeleton enemies i added a Rigidbody 2D, a Box collider 2D, with it's Trigger on, and the HurtPlayerOnContact script so that they can hurt and kill the Player. I also adjusted their box collider to represent their animation better, i also frezzed their Z rotation so that nothing weird can happen and i added the Enemy tag and Enemy Layer to all of my enemies.

Then i went to my Ghost enemy which will a bit different them my other enemies, this enemy will only have thee box collider 2D, with Trigger on, and the HurtPlayerOnContact becasue i want this enemy to patrol up and down above the ground, like a ghost, i'm doing this so that my game won't have the same type of enemies all over the level, with different types of enemies in my game, it makes my game more interesting and better.

I then did a test to see how my Player reacts to my enemies so far and how they look in game and such. The ghost enemy looks well animated and does make my Player get knocked back but i can't damage it, mainly because it doesn't have enemy script on it, without it the Enemy ghost can't be damaged or killed. Another thing i noticed in this test is that my skeleton and cat enemies car nowhere to be found, after some thought i realized that they have fell through the ground, the reason for this is because they have Rigidbody 2D's and their colliders have Is Trigger enabled, that's way, but it's not a big problem because i know a easy solution to solve this.

While i was continuing testing my game and made my screen big to the maximum i realized something, and that was my game look a bit too be for my comfort, my Player, enemies and even background look too big in my opinion, i even checked the reference video i was using and in that video the Player, the background and everything were smaller, and thus a better experience to play it instead of playing a game where the Player covers one third of the screen. I thought about fixing this and making everything in my game smaller, by either increasing the pixels per unit, or by decreasing the scale of everything by a few values, but then i thought about it more and realized that there would be more problems if i did that, the colliders may not match with how the Player and enemies look so then i'll have to adjust them which may take a lot of time and there may be other problems.

To solve this small problem i was having, about the enemies falling to the bottom, i added another box collider 2D to my cat and skeleton enemies, a very small and thin box collider that would be at the bottom of the enemy, so that it wouldn't interact with anything necessary. The Box Colliders 2D will also have their Is Trigger disabled so it doesn't fall to the bottom.

Later i went to test this and now my enemies will not fall to the bottomless pit when i start my game because they now have sometime preventing them from falling, which is the extra box collider 2D that has the Is Trigger disabled.

But when i went to touch one of my enemies, mainly the cat monster, the Player pushed the enemy and didn't get knocked back, while the rest of the enemies made the Player get knocked back, i looked at the enemies inspector and then i realized that the cat's box colliders had Is Trigger disabled when the box collider that is the biggest should have is Trigger enabled, but that is a easy fix that i solved later.

When i'm attacking my new enemies i created i get this weird warning error in the console that says the object reference is not set to an instance of an object, i don't know what this is happening but i think i can easily fix this just by adding the Enemy script to my new enemies so that the Player can damage them and kill them.

I then started making a new script called "enemy_move" this will make my new enemies move left and right, where i can adjust their speed and make them flip, depending on where they are moving, left or right. All of this script and code will be accomplished thanks to a video am watching and will copy the code to make this all possible, the video is "How to Make Unity 2D Enemy Patrol Movements Unity3D Tutorial", i'm using this video because is has code where you can add invisible wall that will force a enemy to turn, which i like a lot because i can change and adjust where the enemy can go better as i have more options and less limits then if i were to use a different code like a code where the enemy move depending if there is a cliff or not, making them move depending on a platform and not the enemy or a invisible wall.

This is the first part of the code i copied.

This is the second part of the code i copied, since the code is a bit too big for me to take a screen shot of it all, so i'll have to take multiple screenshots, since if i simply make the numbers and letters smaller then it will be too hard to see them.

After that i created four new empty GameObject with Box Colliders 2D and placed them to where i want them to be, with Trigger enabled, and having a tag called " enemy turn Wall" so the enemies know what box colliders will make them turn. then i made a empty gameObject where i can place my box collider Walls so that my hierarchy becomes more clean and organized.

Later i went and tested if my enemies will move and turn when they hit the wall and i'm happy to say that my enemies do move, they can be killed by the Player, they can make the Player get knocked back and do turn when they hit a wall but my enemies look like they are moving backwards, i believe this has something to do with the scale in the script, i can fix this by making the values from positive to negative and vice-versa in the if (MoveRight) statement, the first value in the new Vector2.

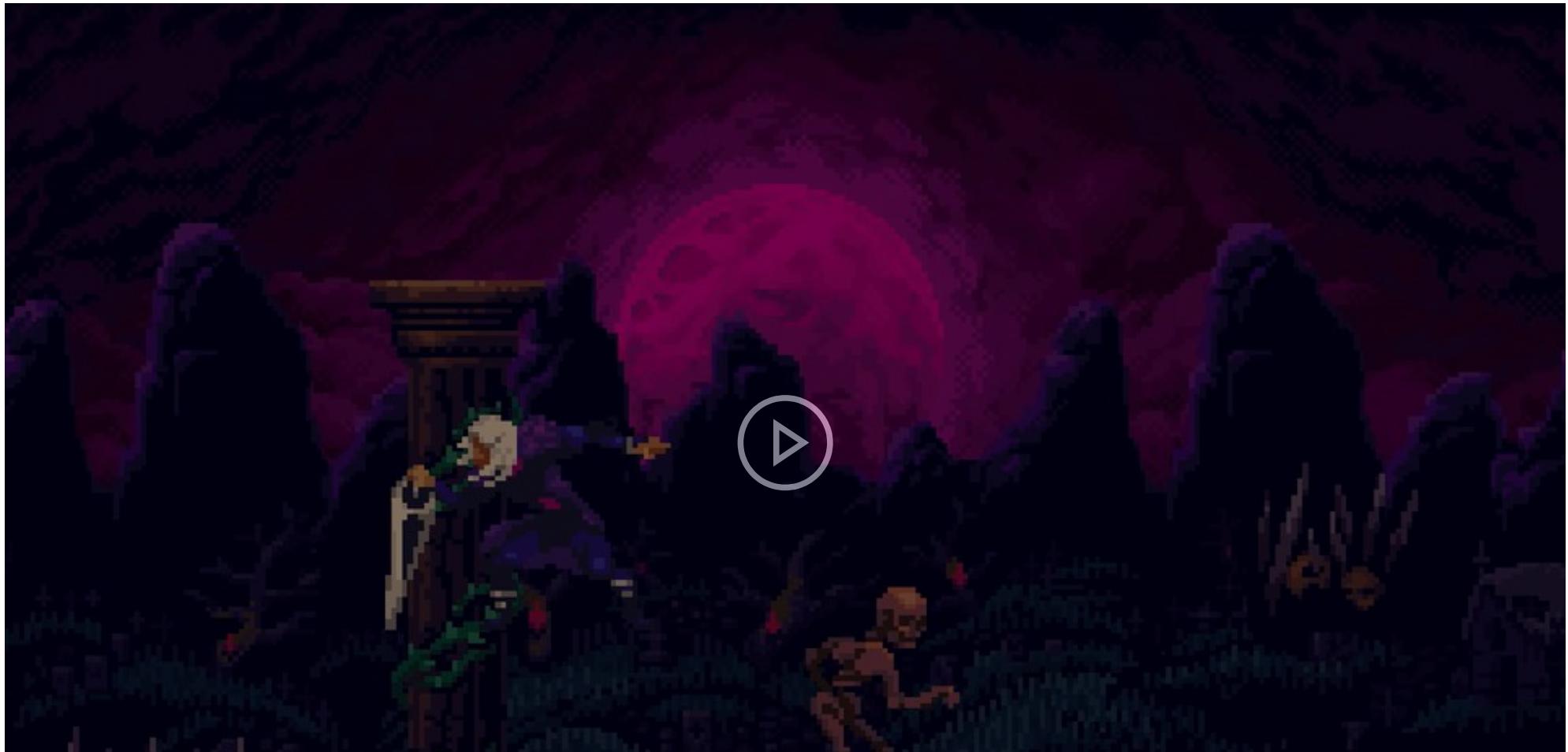
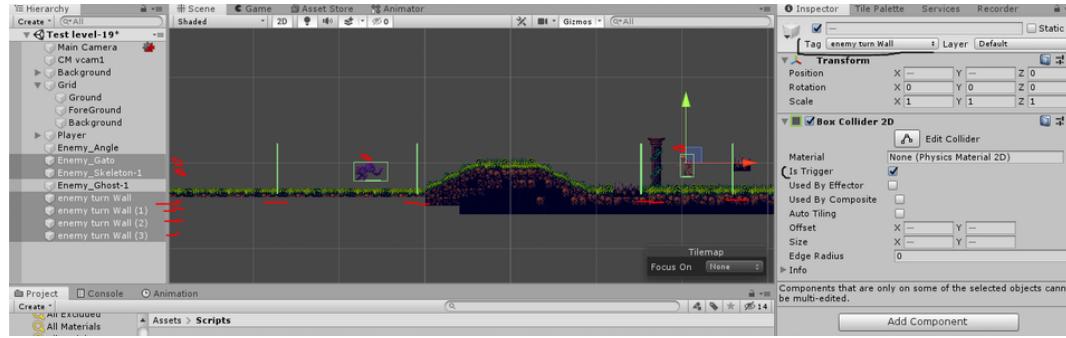
After i did that the problem has been fixed and my enemies look fine, they can be destroyed by the Player and the Player can even be killed and move back when it touches the enemies. The animation on the enemies also still look nice.

While i was testing my game i noticed something, my fields background didn't fill the whole screen, you can see where the fields end, like someone has made a cut in the background. I don't know why that is, this means that my bockgroundLoop script didn't work, so now i have to fix it in a different way because i'm running out of time before i have to present my game in the Final Product tab in SharePoint. I can fix this by simply adding more filed background so that they cover a wide range, it does make my game move heavy but it is what i need to do to make my game look nice.

The screenshot shows the Unity code editor with the script `enemy_move.cs` open. The code is written in C# and defines a class with three methods: `Start()`, `Update()`, and `OnTriggerEnter2D(Collider2D trigger)`. The `Update()` method handles movement based on the `MoveRight` variable, which is set to `false` when the enemy enters a trigger area tagged "enemy turn Wall".

```
11 void Start()
12 {
13 }
14 
15 // Update is called once per frame
16 void Update()
17 {
18     if (MoveRight)
19     {
20         transform.Translate(2 * Time.deltaTime * enemySpeed, 0, 0);
21         transform.localScale = new Vector2(1, 1);
22     }
23     else
24     {
25         transform.Translate(-2 * Time.deltaTime * enemySpeed, 0, 0);
26         transform.localScale = new Vector2(-1, 1);
27     }
28 }
29 
30 void OnTriggerEnter2D(Collider2D trigger)
31 {
32     if (trigger.gameObject.CompareTag("enemy turn Wall"))
33     {
34         if (MoveRight)
35         {
36             MoveRight = false;
37         }
38         else
39         {
40             MoveRight = true;
41         }
42     }
43 }
44 }
```

The new script i created and the code i added to it, the bottom part of it.





The test i did to see how the new script "enemy_move" make the enemies act and move, after that i realized that there was a problem.



The test i did after fixing the problem the enemies had with moving one way but looking like it moving the other way.

22/04/2020: I wanted a different type of enemy so i went and made a new script called "enemy_move_up" this script will make a enemy move up and down, so it will look like it's blocking a area, not only that but i'll have more enemies that are more interesting and diverse,making my game better and interesting to play.

The first thing i did was make a new C# script called "enemy_move_up" that will have the code to make my new enemy move up and down. The code i will use will be similar to the enemy_move script, just with some slight changes, like the enemySpeed being in the Y axis value and not the X axis and the tag having a different name so i can tell the difference, but the rest should be the same.

Later, i went to my Unity editor and added two empty game objects, with box colliders 2D, called "enemy turn Wall UP" these will represent the invisible wall that the ghost enemy will touch and that will make the enemy move the opposite way, making the enemy move up and

down. I made sure everything was correct, i looked at the box collider 2D and made sure the Trigger was on and i made sure that the ghost enemy and new walls had the right tag.

After that i tested my enemy and for some reason i wasn't working, the ghost enemy would just go up and not stop, i thought that maybe if i added another box collider 2D with the Trigger off, that would fix the problem i was having.

I went and tested my game to see if the Ghost moves up and down but unfortunately it doesn't, not even after i added a new box collider to it, i did this because i thought that maybe a collider with Is Trigger off would connect with the Is Trigger vertical walls, but that didn't help. For some reason the enemy is only going up, i even made the enemy move slower to make sure it was that and not the enemy simply disappearing.

For some reason i had a weird issue about the background, when i entered play mode, the mountains background was flickering and i believed that this was because of where the Player was, the background is trying to follow the Player and since i placed the background in the 0, 0, 0 position and the Player is somewhere else it's trying to follow it, which causes the problem, but i believed this is easy to fix i just have to make the Player in the same position as the background.

To fix the problem i was having with the ghost enemy i thought that maybe if the code of the enemy_move_up was the same as the enemy_move, then the problem would be fixed. So what i did was copy all of the code from enemy_move and add it to enemy_move_up, the only difference the new script will have is that the float enemySpeed will be in the Y axis value and not the X axis, and the localScale will be the same no matter what, since i don't want my ghost enemy to turn left and right.

Another change i needed to do was change the tag of my new walls to be enemy turn Wall.

After all that i went and tested this and still my enemy would go up and not stop, but thanks to the that i believe i now know the problem, because if the problem wasn't the code then the problem must be with my ghost enemy and what it has in the inspector window.

So with that logic i thought about it and decided to add a Rigidbody 2D but with it's drag, gravity and mass zero, with it's Z axis frozen of course.

Then i tested the enemy and now everything works, the enemy moves up and down when it touches the invisible walls. It just shows you that even if you want a enemy to not be affected by gravity or you want a enemy to fly it's might be good to add a Rigidbody even if you plan on not using it.

After the success i did with the enemy i wanted to change the enemy_move_up one last time so that it can become more clean, organize and represent what the code will do to a object.

I also changed the tag again to be "enemy turn Wall UP", but now that i think about it, it probably was unnecessary to make a new tag just for enemies that will move up and down, but what done is done and i don't want to wast time undoing things, i just want to move on and complete my project.

I went to testing my game to see if everything else was working fine and that there was no problems what so ever, and i'm glad to say that everything works well and there are no issues, although i think i found something that would be annoying for players, and that is the attack mechanics can be unreliable as seen here in the video of me attacking a skeleton, i did the attack animation multiple times even though i just needed three hits to kill the enemy, i might increase the attack range so that the attack is more reliable and consistent.

I was worried that the background wasn't infinitely looping, so to see if my worries were true, i decided to do a test of sort, i made a very long path way so the player can run for a long way, to see if there is an end to the background. I did the test and the background was still in the screen, the mountains, fields and the moon, i even went to the scene window in play mode and tested if the loop was working by moving the Player and the background was following the Player and the fields and mountains were looping.

With all that i decided what i wanted to do next, and that was make a death animation for my enemies when they die.

I did know how animations work and how to implement them to a gameObject, sort of, but i did use a video to help me in a way and that video was "MELEE COMBAT in Unity".

Then i went to the Animator tab and dragged the death animation i created, which is a explosion animation from a asset i downloaded, to all of my enemies Animator tab, after that i created two parameters, a "Hurt" trigger and a "IsDead" bool, what i did first was added a transition from the enemies normal state to thee death state, what's in the transition is a condition that can only happen if IsDead is true, and i also made sure that exit time is off and so is fixed duration.

I wanted to know how to add a hurt animation to my enemies like a white flash when the Player hits them, so i did some research and found a video called "GameDev Tutorial: Howto make enemies Flash and Explode in Unity", but the way the person makes the enemy flash white is quite different then i expected, i thought he would use the animations and the animator, but i was wrong and because of that i'm a bit scared of adding a hurt animation of some sort to my enemies, i'm afraid i might break thing making waste a lot of time, and that is not good since the end of the final major project is near the end. But it might be a good addition to add, so if i get some ideas on a better way to implement this i'll do it because right now it's quite hard to tell if the Player has hit the enemy or not.

After some thinking on how to fix the death animation problem i was having, i rewatched the "MELEE COMBAT in Unity" video and thanks to that i realized on to fix the problem, i simply had to add a delay on the Destroy(gameObject) so that the animation can play before the enemy is destroyed, i just had to add a number next to the gameObject for how long i want to wait before the enemy is destroyed.

I then went to test my game to see if the animation was working properly and i happy because it works, the enemy explodes when it's health reaches zero or below. But now i noticed another problem with this death animation and that is that the animation moves when you kill a enemy, for example, you destroy a skeleton the explosion will move like if the skeleton was still alive, not only that the box colliders and scripts are still active even when the explosion is happening, so the Player can still get knocked back and get damaged after the enemy is presumably dead. This is not a good thing to have in your game and should be removed, so now i will try to make sure when the enemies health is zero or below that the enemies can't do anything, just do the death animation and then be completely destroyed.

After doing that test i tried doing some stuff to my Enemy C# script, at the if (health <= 0) statement where stuff will happen if the enemies health is zero or below, what i knew i needed was some code that will disable or prevent the enemy from moving and being unable to affect the Player when kill and then do the death animation and then the last code is it being destroyed.

First, i tried to add some code that will disable the scripts the enemy is using but for some reason, that didn't work, i couldn't do that with multiple scripts, i could only do it with either enemy_move or enemy_move_up and i wanted both of them and not just one. So i decided to try something else, i even did some research on how to prevent a enemy's movement after they die, i researched about disabling rigidbody2D via script, the Rigidbody itself, how to Freeze a objects position in 2D in the script, but nothing worked, the only thing i managed to get to work was a code about disabling the Collider2D of the object with the Enemy script, which made the enemy not affect the Player when they die, but even with that they can still have a moving explosion animation when they die and i wanted to fix that.

I did even more research and thought about different ways to do some things, especially how to make some of the scripts the enemies have not work anymore so they stopped moving when they die. I then thought about destroying the script instead of disabling it, i did some research and found something which is Object.Destroy also known as the code which is: Destroy(GameObject obj, float t = 0.0F); this is the code you use if you want to destroy something. Then i thought about if i can destroy the script component my enemies are using, i tried using Destroy(GetComponent<enemy_move>()); and Destroy(GetComponent<enemy_move_up>()); even with both of them in the Enemy script it works when i tested this out.

I then made the full code complete, in the if statement about if the health is zero or below the enemy will have it's movement scripts destroyed, then the box collider 2D will be disabled, then the death animation will appear and only after all that will the enemy be destroyed after 0.45 seconds as i added a delay to the Destroy(gameObject), without the delay, the enemy would be destroyed instantly without doing the animation.

Then i fully tested my game to see if everything works out properly and there are no errors or problems anymore. And i'm glad to say that now my enemies have a working death animation, where they actually die and don't affect the Player after they die. I even tested if the enemies can still damage the Player and kill it, and they still can when the Player's Health Manager is enabled.

I took some time to get this death animation to work properly, a lot of mistake happened but i'm happy with the results. Now all that is left to do to make my game complete for me is have one last new enemy that shoot projectiles, make a proper level, add at least one looping sound track, and one or two sound effects of attack and getting hit by enemies, and maybe add a checkpoint mechanic where if the Player dies you respawn at the start of the level. Another thing my game could have is a start screen, when you start the game before actually play the game, and a end screen, when you finish the game.

The screenshot shows the Unity Editor's code editor window with the script `enemy_move_up.cs` open. The code is a C# script for a Unity GameObject. It includes methods for `Start()`, `Update()`, and `OnTriggerEnter2D`. The `Update()` method contains logic to move the transform vertically based on the `MoveUp` boolean. The `OnTriggerEnter2D` method checks if the trigger's game object has a specific tag and toggles the `MoveUp` value.

```
public float enemySpeed;
public bool MoveUp;

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

// Update is called once per frame
void Update()
{
    if (MoveUp)
    {
        transform.Translate(2 * Time.deltaTime * 0, enemySpeed, 0);
    }
    else
    {
        transform.Translate(-2 * Time.deltaTime * 0, enemySpeed, 0);
    }
}

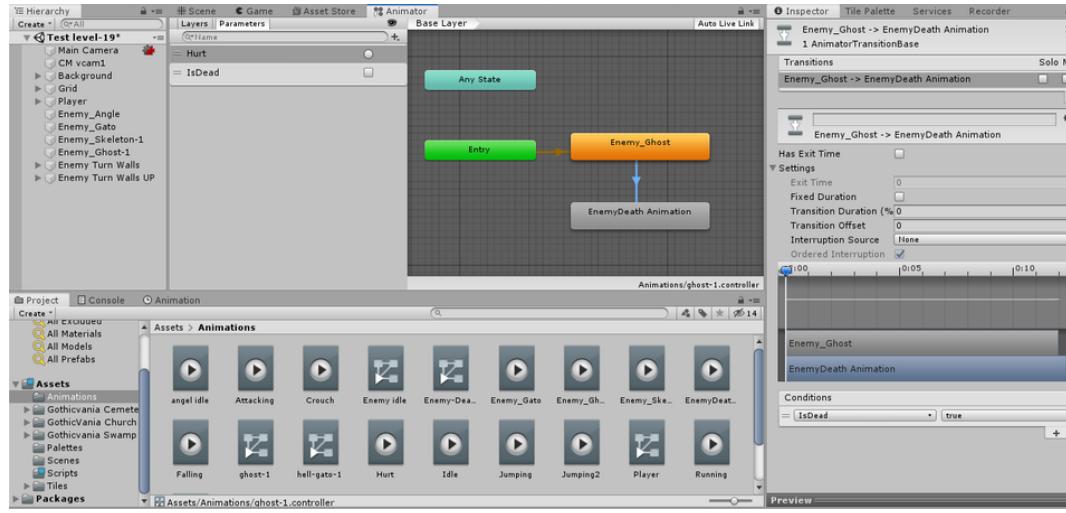
void OnTriggerEnter2D(Collider2D trigger)
{
    if (trigger.gameObject.CompareTag("enemy turn Wall UP"))
    {
        if (MoveUp)
        {
            MoveUp = false;
        }
        else
        {
            MoveUp = true;
        }
    }
}
```

The new script, "enemy_move_up", i created and the code i copied from the from the "enemy_move" script so that it is similar to it but it has a few changes mainly the enemy will move vertically not horizontally.

The screenshot shows the Unity code editor with the tab bar at the top containing files: Player.cs, enemy_move_up.cs, enemy_move.cs, HealthManager.cs, PlayerAttack.cs, and Enemy.cs. The Enemy.cs file is currently selected and has its title bar highlighted. Below the tabs, there is a search bar with the placeholder text "Update()". The main area of the editor displays the C# code for the Enemy script:

```
6  {
7      public int health = 100;
8      public float speed;
9      private float dazedTime;
10     public float startDazedTime;
11
12     public Animator anim;
13
14     // Start is called before the first frame update
15     void Start()
16     {
17         anim = GetComponent<Animator>();
18     }
19
20     // Update is called once per frame
21     void Update()
22     {
23         if (dazedTime <= 0)
24         {
25             speed = 5;
26         }
27         else
28         {
29             speed = 0;
30             dazedTime -= Time.deltaTime;
31         }
32
33         if (health <= 0)
34         {
35             anim.SetBool("IsDead", true);
36
37             Destroy(gameObject);
38         }
39     }
40 }
```

The small changes i did to my Enemy script so that my enemies can do a explode animation when th...
die.



The changes i did to all my enemies in the Animator, by added a Hurt Trigger and a IsDead Bool. I also made a transition to idle and death animation, with exit time off and 0 transition duration.

The screenshot shows the Unity code editor with the tab bar at the top containing files: Player.cs, enemy_move_up.cs, enemy_move.cs, HealthManager.cs, PlayerAttack.cs, and Enemy.cs. The Enemy.cs file is currently selected and has its title bar highlighted. Below the tabs, there is a search bar with the placeholder text "Enemy". The main area of the editor displays the C# code for the Enemy class:

```
Assembly-CSharp
else
{
    speed = 0;
    dazedTime -= Time.deltaTime;
}
if (health <= 0)
{
    anim.SetBool("IsDead", true);
    Destroy(gameObject, 0.5f);
}
```

The small change that made the death animation work before the enemy is destroyed, before the enemies would just be destroyed and not do a death animation.

The screenshot shows the Unity Editor's code editor window with the tab bar at the top containing "Player.cs", "enemy_move_up.cs", "enemy_move.cs", "PlayerAttack.cs", "HealthManager.cs", "Enemy.cs", and "Assembly-CSharp". The "Enemy.cs" tab is selected, and the code editor displays the following C# script:

```
20 // Update is called once per frame
21 void Update()
22 {
23     if (dazedTime <= 0)
24     {
25         speed = 5;
26     }
27     else
28     {
29         speed = 0;
30         dazedTime -= Time.deltaTime;
31     }
32
33     if (health <= 0)
34     {
35         Destroy(GetComponent<enemy_move>());
36         Destroy(GetComponent<enemy_move_up>());
37         GetComponent<Collider2D>().enabled = false;
38         anim.SetBool("IsDead", true);
39
40         Destroy(gameObject, 0.45f);
41     }
42 }
43
44 public void TakeDamage(int damage)
45 {
46     dazedTime = startDazedTime;
47
48     health -= damage;
49     Debug.Log("damage TAKEN!");
50 }
51
52 }
```

The modification I did to this script in order for the death animation not move when the Player kill a enemy and not still make the Player get knocked when touching the explosion animation.

24/04/2020: After i made the basic simple enemies, the ones that move up and down, left and right and the ones that stand still, i wanted to make a enemy that shoots projectiles at the Player, even though i believed it would be very hard to do, because it's quite complicated to make a enemy spawn a projectile, make the projectile move and then do a specific animation for the wizard enemy that i have and plan on using. While every other enemy was simple to make, they just moved left and right, up and down or stay idle, even though i know it will be hard to make a enemy that shoot at the Player, i still want to do it because that will make my game and enemy more diverse, interesting and, more importantly, fun.

I added a Enemy_Wizard to my scene, animated and all, with a Rigidbody 2D, and two box colliders 2D, one is a Trigger so that it can be damaged and can hurt the Player, the other will be used as a way to prevent the enemy from fall to the bottom. I also added it the Enemy and HurtPlayerOnContact scripts, another thing i added to the scene was the projectile itself, which is a fireball of some sort, which has a circle collider 2D, with Trigger enabled, and it has the HurtPlayerOnContact component script so it can hurt the Player. Both of the fireball and the wizard enemy have their respective Layers, the fireball has a new Layer called "Projectile".

Then i created a new C# script called "Projectile", this will have all the code for the fireball needs for it to move, left or right and be able to damage the Player. The code where i got this is from a video called "How to Make Your Enemies Fire Projectiles - Unity 2D Platformer Tutorial - Part 23", this video will help me make all of this possible, has it's a video that has a enemy that behaves has i want my enemy, i want my wizard enemy to shoot fireballs when the Player gets nearby and shoot projectiles once every second or so, the projectiles also move in a straight horizontal line.

This is the test i did after i added the Projectile script to my fireball and entered play mode to see if the fireball moves. I am happy to say that the fireball move as intended although there is a small problem, when the player touched the fireball the Player died instantly, i believed there is something wrong with the Player's HealthManager or with the HurtPlayerOnContact script, but i won't change them, what i'll probably do is increase the Players health or make the enemies and projectiles do less damage.

After that i made a new C# script called "ShootAtPlayerInRange", which is from the "How to Make Your Enemies Fire Projectiles - Unity 2D Platformer Tutorial - Part 23" video i'm watching. This script will make the wizard enemy be able to spawn the fireballs, from a LaunchPoint empty object, that will go left or right depending if the Player is left or right from the enemy.

Later i added the script i just made to the wizard enemy and then i made some modifications to the scripts component inside it, the values and all. I also created a LaunchPoint empty gameobject so that the fireballs spawn from there, the empty object will also be a child of the wizard as if i create more wizard i don't need to duplicate more LaunchPoints separately, it will make my life easier.

Then i made my enemies into prefabs, the reason i did this is because in the video that i am watching it has enemy and projectile prefabs, so because of that i did what i did otherwise my script might not work because my object are different, better to be safe then sorry.

Another thing i notice was when i have been testing my game i have been getting these error about NullReferenceException, i don't really know what this is about, i just know that a line of code is calling something and that something is null, i believe. I'll have to try and fix this because when i try to test my game and i get close to my wizard, the wizard doesn't shoot any fireballs.

This is a small test i did, after adding fireballs to the scene and doing some changes the Projectile script, to see how the fireballs react to the Player, what i found was that when i disable the player's Health Manager component so the Player wouldn't be damaged, the projectiles would duplicate itself when in contact with the player, maybe that's the reason the player was dying instantly when it came in contact with the fireball, the fireballs would damage and duplicate itself dealing a lot of damage to the Player. I need to fix this issue so that my game is fixed and balanced.

After quite some time trying to fix the issue of the Projectile, by thinking, changing some values and more. But then i thought about something, since i knew what some types of code did what, i knew that the instantiate code is used to call or invoke something, and i was using that to call the fireball object, and then i realized that maybe my projectile doesn't need to instantiate anything, because i don't have anything to instantiate with my fireball object unlike the video that i'm watching where the person doing the game in the video has GameObject which is a particle effect where it's used for it's projectile when it comes in contact with something, showing people that it's destroyed. I don't have anything like that so i don't need the instantiate code.

So after i realized that i removed the fireball variable, the instantiate code and some other stuff that was unnecessary to my projectile.

I then went to my Unity editor and did some changes to my fireball and then i added three fireball projectiles to my scene so that i can test them and see how they react to my player.

In this test i saw the projectile move fine, and when i touched them they were able to knock the Player back, not only that but when they touched anything else like the ground they would disappear, or better be destroyed, which is how i wanted my fireballs to react.

After i fixed some more modifications to my game and it's scripts, i went and tried to finally test if the Enemy_Wizard can spawn the fireball as i want it to, and i'm happy to say that everything works properly, not only does the enemy spawn fireball, deal damage to the Player by itself or with the fireballs but it can also be destroyed, by the Player and not anything else.

Even though i like what i did and what i'm seeing, there are a few problems, these problems don't bother me since the Player might not notice or it's not important for me to fix it but these problems are, if the Player is to the right of the wizard and gets very close to it and ducks, the fireball will spawn from the LaunchPoint, but then it will move to the right, like it's trying to follow the Player. Another problem is if the Player is to the right of the enemy wizard the wizard won't shoot any projectiles at all. I won't fix any of these small problems because i don't want my enemy to shoot fireballs left and right and the problem with the fireball moving to the right when the Player is close to the enemy isn't a big problem for me, it might look weird when you play the game, but it's not a game breaking but, and i have more important thing to work on then this small problem.

For some reason, whenever the fireballs would spawn, in the console, there would be a error message saying:

"Destroying assets is not permitted to avoid data loss. If you really want to remove an asset use DestroyImmediate (theObject, true);", the thing is the fireball would still be destroyed after 3 seconds as i made it that way in the script, so this warning isn't preventing that, but i wanted this annoying warning to stop so i tried this way of destroying, but this way won't let me delay the destroy code, and i don't want to make a coroutine code just to delay a destroy code, so i'll leave this my old destroy code, since it's not ruining or breaking anything that i know of.

I then wanted my wizard enemy to be able to do the death explosion animation, so i went to it's Animator and added the Death animation and made the parameters and did it's transitions, with a condition and exit time off, simple as that.

Later i went to test if the new enemy does do the death animation when the Player kills it, and it does work perfectly, the enemy explodes after the Player hits it with it's attack. There were also no problems or issues what so every during this test.

Then i wanted my Enemy_Wizard to be able to do it's attack animation before the actual projectile is spawned, because the way the fireball is spawn from the wizard is weird, the enemy is just standing there idle while fireballs are appearing in front of it whenever the Player gets close. So because of that i want to fix that small problem, by adding additional code to my ShootAtPlayerInRange script, i don't know exactly how to do that but thanks to past experience in doing other animations to my Player and it's attack animation, i know the general idea of it.

The first thing i need to do is make a public Animator, then get the component in void Start, after that is make the animation happen when the enemy does something. Then i'll just have to add the attacking animation to the enemy's Animator, make a new Parameter called "EnemyAttack" and make transitions.

I also made one part of the if statement it had because i realized that it didn't affect the gameplay of my wizard so i decide to note it out, it's the green if statement that is noted out.

I then went to test if the wizard enemy does the animation properly, and thankfully the animation does play when the Player enters the enemy's "field of vision" that will make the enemy shoot at the Player, the only problem with the animation is that it doesn't sync well with the fire ball spawning in, it looks weird, i need to find a way of delaying the spawning of the fireballs so that it syncs well when the enemy is about to do a sort of launching something with it's hand on fire, because this will make my game more natural and not weird looking.

After realizing the problems my enemy attack animation still has, i decided to do some modifications to the new script my Player has in order to make my Enemy_Wizard attack animation be sync to the fireballs spawning, using the coroutine code, and now the enemy attack animation is slightly more in sync but now i have a problem with the projectiles not working properly, it's spawning at the right time but it's not moving, and that's not the only problem i'm having because the enemy is now doing attack animation twice before returning to the idle animation when it's should be doing it once and then return to normal, not only that but it would seem that the syncing won't be the same every time because at the start when i got near the enemy the syncing wasn't as close as i wanted but then later it got better and if i have stayed longer in the test just watching the enemy i believed it would have gone worse again.

To solve all of the problems i was having with the Enemy_Wizard i went back to the ShootAtPlayerInRange and tried to do some changes in order to fix the problems i was having, The first thing i tried to do was change thing around the coroutine, i though about what the

problem was exactly and how it did what it did, i was trying to understand it and then it hit me, i was trying to delay multiple things like the shotCounter and the Destroy(fireball, 3f) code when in fact it should just be the Instantiate code while the rest are left in the if statements just as they are, not impacted by the delay.

I also made some changes to this script such as, making everything more organized and bringing back the other if statement that was noted out, it may have been doing nothing in helping my Enemy wizard to be better nor was it doing any harm, and because of that it's in my script now, not only that it make my script more clean and organized.

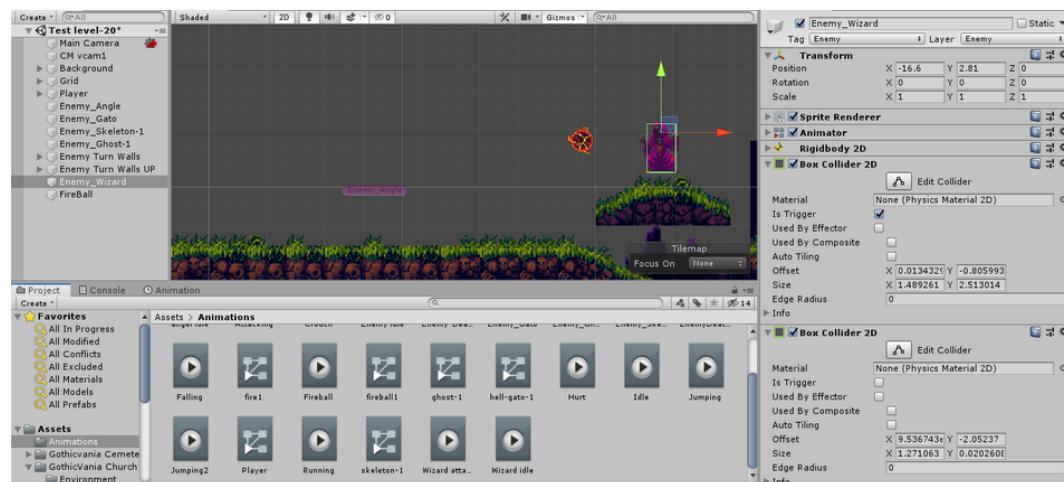
Then i made some changes to my Enemy wizard and other stuff so that my enemy is better at shooting fireballs at the Player and at other stuff, i also made sure everything was in place, the enemy objects had the right scripts, components, values, tags, Layers and all.

Later i went to the wizard's Animator to make some final changes so that the attack, idle and death animations work properly, i made sure that the transitions had the right conditions and everything.

After everything, i did a test to see if everything reacts the way it should, and it does, the enemy wizard does the animation perfectly which syncs well with the spawn time of the fireball spawning, even though it still has the problem with the enemy not fire the fireball to the right and the fireball moving to the right if the Player is too close the wizard, as seen in this video to the right.

Later i did some more improvements to my code to make it better and cleaner, i even added a line of cod in the Enemy script where now the enemy wizard can also lose it's ability to shoot fireball after it dies just to make sure it can't impact the game after it's death.

And that is pretty much all of the different types of enemies i want to add to my game, hopefully this will make my game good enough for players to enjoy it, after this i'm going to start making a proper level, then some sound effects and music, and maybe even add a new feature or two, like a white shade or material showing the players that they have hit the enemy, then after that make a few adjustments to my game and then finally make a start and end scene so it can be a somewhat proper game, if i have time to do all of these.



The new Wizard enemy i added, with the components i added to it, to my scene with the fireball obje...
that it will shoot from, again with the components that it needs.

The screenshot shows the Unity Editor's code editor window with the tab bar at the top containing "Projectile.cs", "Player.cs", "enemy_move_up.cs", "enemy_move.cs", and "PlayerAttack.cs". The active tab is "Projectile.cs". The code editor displays the following C# script:

```
6  {
7      public float speed;
8      public PlayerController player;
9      public int damageToGive;
10     private Rigidbody2D rb2D;
11
12     public GameObject fireBall;
13
14     // Start is called before the first frame update
15     void Start()
16     {
17         player = FindObjectOfType<PlayerController>();
18
19         rb2D = GetComponent<Rigidbody2D>();
20
21         if(player.transform.position.x < transform.position.x)
22         {
23             speed = -speed;
24         }
25     }
26
27     // Update is called once per frame
28     void Update()
29     {
30         rb2D.velocity = new Vector2(speed, rb2D.velocity.y);
31     }
32
33     void OnTriggerEnter2D(Collider2D other)
34     {
35         if(other.name == "Player")
36         {
37             HealthManager.HurtPlayer(damageToGive)
38         }
39
40         Instantiate(fireBall, transform.position, transform.rotation);
41         Destroy(gameObject);

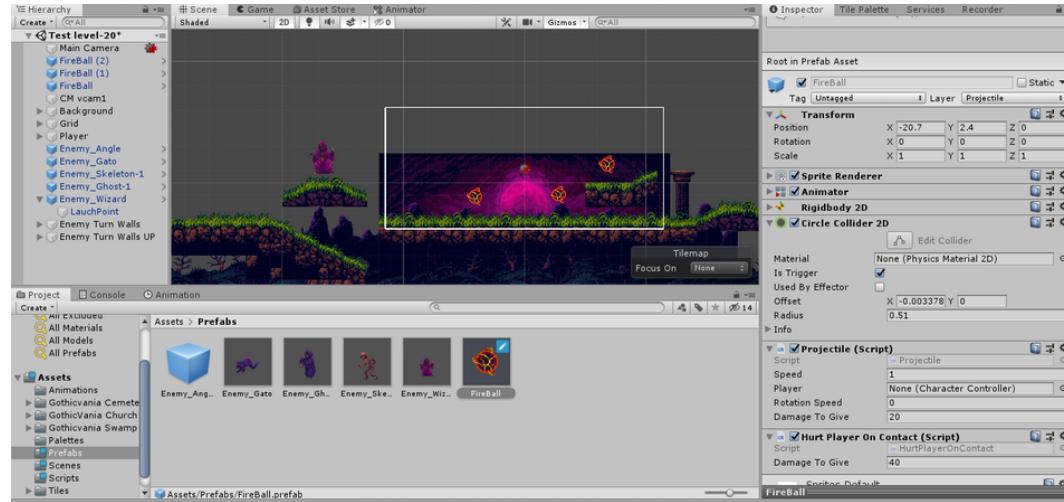
```

The "Projectile" C# script I created with the code I added from a video I was watching, this script will be given to the fireball object, the projectile that the new enemy will shoot.

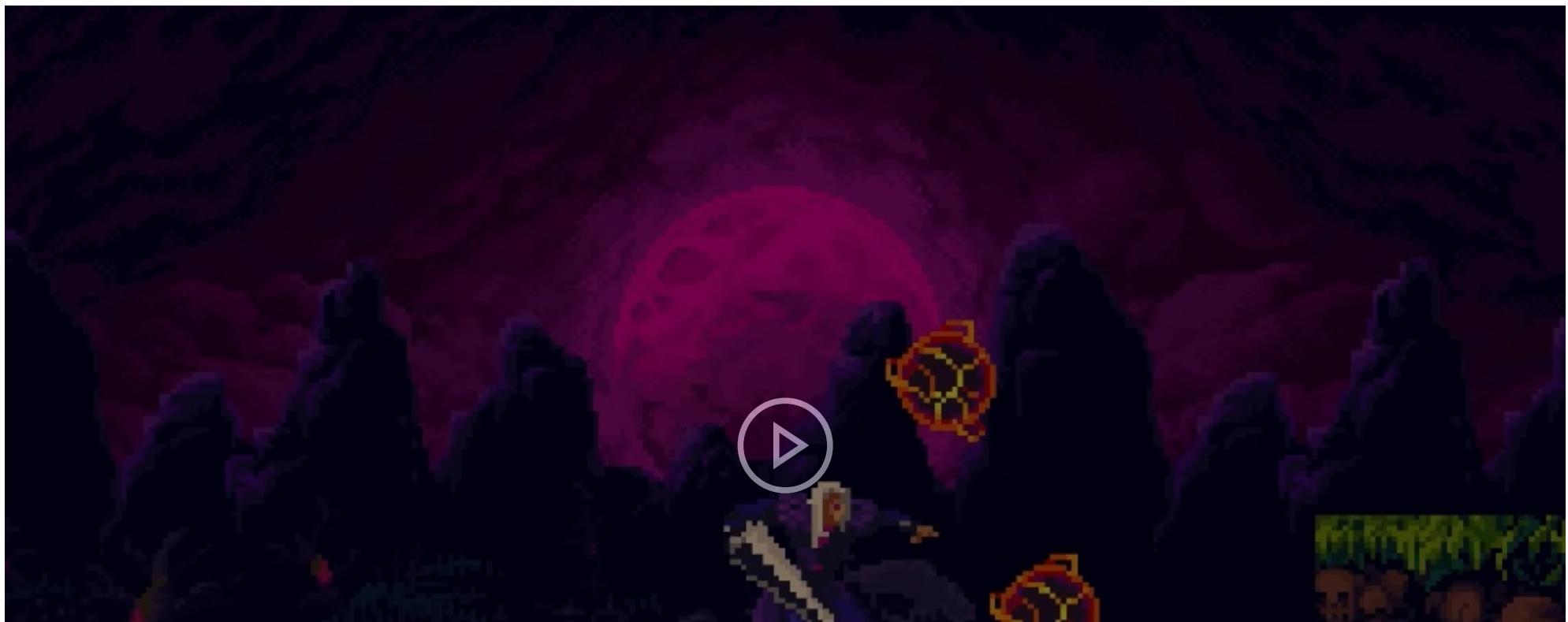

```
ShootAtPlayerInRange.cs    Projectile.cs  X  enemy_move_up.cs  enemy_move.cs
Assembly-CSharp  Projectile
7     public float speed;
8     public CharacterController player;
9     private Rigidbody2D rb2D;
10    public float rotationSpeed;
11    public int damageToGive;
12    //public GameObject fireBall;
13
14    // Start is called before the first frame update
15    void Start()
16    {
17        player = FindObjectOfType<CharacterController>();
18        rb2D = GetComponent<Rigidbody2D>();
19
20        if (player.transform.position.x < transform.position.x)
21        {
22            speed = -speed;
23            //rotationSpeed = -rotationSpeed; //----
24        }
25    }
26
27    // Update is called once per frame
28    void Update()
29    {
30        rb2D.velocity = new Vector2(speed, rb2D.velocity.y);
31        //rb2D.angularVelocity = rotationSpeed; //----
32    }
33
34    void OnTriggerEnter2D(Collider2D other)
35    {
36        if (other.name == "Player")
37        {
38            HealthManager.HurtPlayer(damageToGive);
39        }
40
41        //Instantiate (fireBall, transform.position, transform.rotation);
42        Destroy(gameObject);

```

The changes I did to the Projectile script so that the fireballs don't do weird stuff and actually work when in contact with the player.



The modifications i made to the fireball prefab and the addition fireballs i added to the scene so that i can test them better.





The test i did to see if the fire balls now work properly, if they do damage to the Player and if they disappear when in contact with any collider.

The screenshot shows the Unity code editor with the script `ShootAtPlayerInRange.cs` open. The code is written in C# and defines a class `ShootAtPlayerInRange` that inherits from `MonoBehaviour`. It includes methods `Start()` and `Update()`. The `Update()` method contains logic to check if the enemy is within range of the player and to trigger an attack animation before shooting a fireball.

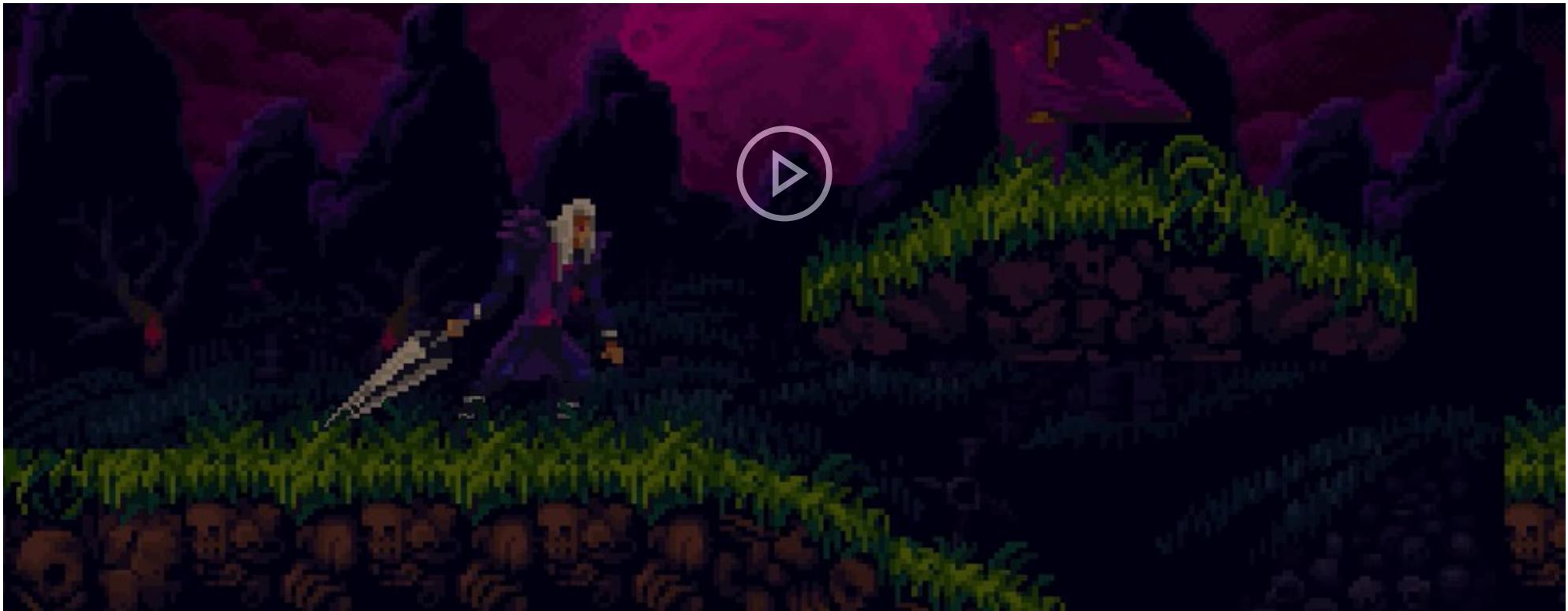
```
17  public Animator enemyAnim;
18
19
20 // Start is called before the first frame update
21 void Start()
22 {
23     enemyAnim = GetComponent<Animator>();
24
25     player = FindObjectOfType<Player>();
26     shotCounter = waitBetweenShots;
27 }
28
29 // Update is called once per frame
30 void Update()
31 {
32     Debug.DrawLine(new Vector3(transform.position.x - playerRange, transform.position.y, transform.position.z), player.transform.position);
33     shotCounter -= Time.deltaTime;
34
35     //if (transform.localScale.x < 0 && player.transform.position.x > transform.position.x)
36     //{
37
38         //  Instantiate(fireball, launchPoint.position, launchPoint.rotation);
39         //  shotCounter = waitBetweenShots;
40         //  Destroy(fireball, 3f);
41     //}
42
43     if (transform.localScale.x > 0 && player.transform.position.x < transform.position.x)
44     {
45         enemyAnim.SetTrigger("EnemyAttack");
46
47         Instantiate(fireball, launchPoint.position, launchPoint.rotation);
48         shotCounter = waitBetweenShots;
49         Destroy(fireball, 3f);
50     }
51 }
```

The code I added to my ShootPlayerInRange so that the wizard enemy could do it's attack animation
before it actually shoots the fireball projectile.



The video that shows you the enemy doing the attack animation but it's doing it too soon.





The test i did after doing some modifications to my wizard and it's scripts, showing the enemy spawning the fireballs but it's not moving and for some reason the enemy is doing the attack animation twice then doing the idle animation.

The screenshot shows the Unity code editor with the script `ShootAtPlayerInRange.cs` selected. The code defines a class `ShootAtPlayerInRange` that inherits from `MonoBehaviour`. It contains variables for a player, a fireball, a launch point, player range, wait between shots, shot counter, delay of enemy attack, and an animator. The `Start()` method initializes these variables. The `Update()` method updates the shot counter and checks if the enemy has moved out of range. If so, it triggers the "EnemyAttack1" animation and starts a coroutine for a delayed attack. The code uses `Debug.DrawLine` for debugging.

```
public class ShootAtPlayerInRange : MonoBehaviour
{
    public Player player;
    public GameObject fireball;
    public Transform launchPoint;

    public float playerRange;
    public float waitBetweenShots;
    private float shotCounter;
    public float delayOfEnemyAttack;

    public Animator enemyAnim;

    // Start is called before the first frame update
    void Start()
    {
        enemyAnim = GetComponent<Animator>();
        player = FindObjectOfType<Player>();
        shotCounter = waitBetweenShots;
    }

    // Update is called once per frame
    void Update()
    {
        Debug.DrawLine(new Vector3(transform.position.x - playerRange, transform.position.y, transform.position.z), new Vector3(transform.position.x + playerRange, transform.position.y, transform.position.z));
        shotCounter -= Time.deltaTime;

        if (transform.localScale.x < 0 && player.transform.position.x > transform.position.x && shotCounter <= 0)
        {
            enemyAnim.SetTrigger("EnemyAttack1");
            StartCoroutine(EnemyDelay());
            shotCounter = waitBetweenShots;
            Destroy(fireball, 3f);
        }
    }
}
```

The changes I made to `ShootPlayerInRange` so that the enemy could properly do the attack animation and spawn a fireball on time so that the animation syncs with the spawning of the projectiles. This is the top part of the script and the code it has, showing the variables and everything.

The screenshot shows the Unity code editor with the script `ShootAtPlayerInRange.cs` open. The code is written in C# and defines a class `ShootAtPlayerInRange` that implements the `IEnumerator` interface. The script includes methods for finding the player, setting up shot counters, and spawning projectiles. It also contains logic for attacking enemies based on their local scale and position relative to the player.

```
22     player = FindObjectOfType<Player>();
23     shotCounter = waitBetweenShots;
24   }
25 
26   // Update is called once per frame
27   void Update()
28   {
29     Debug.DrawLine(new Vector3(transform.position.x - playerRange, transform.position.y, transform.position.z), new Vector3(transform.position.x + playerRange, transform.position.y, transform.position.z));
30     shotCounter -= Time.deltaTime;
31 
32     if (transform.localScale.x < 0 && player.transform.position.x > transform.position.x && player != null)
33     {
34       enemyAnim.SetTrigger("EnemyAttack1");
35       StartCoroutine(EnemyDelay());
36 
37       shotCounter = waitBetweenShots;
38       Destroy(fireball, 3f);
39     }
40 
41     if (transform.localScale.x > 0 && player.transform.position.x < transform.position.x && player != null)
42     {
43       enemyAnim.SetTrigger("EnemyAttack1");
44       StartCoroutine(EnemyDelay());
45 
46       shotCounter = waitBetweenShots;
47       Destroy(fireball, 3f);
48     }
49   }
50 
51   IEnumerator EnemyDelay()
52   {
53     yield return new WaitForSeconds(delayOfEnemyAttack);
54     Instantiate(fireball, launchPoint.position, launchPoint.rotation);
55   }
56 }
```

The changes I made to `ShootPlayerInRange` so that the enemy could properly do the attack animation and spawn a fireball on time so that the animation syncs with the spawning of the projectiles. This is the bottom part of the script and the code it has, showing the `IEnumerator` code and everything.

27/04/2020: The next thing i wanted to do was make a proper level of players to enjoy, this is where the video reference "Gothicvania-Cemetery Pixel Art Game Play" really helps me, since i don't know much about level design i will use this video to help me make a good level, at least.

What i first started to do what remove my old level and then start making a new one from scratch, i then made two new tilemaps called "Spikes Damage" and "One way grass roof", the spike tilemap will be used for the spikes sprites, i will make them damage and get knock back the Player. Another thing i did was make the "One way grass roof" have the ability to make the Player go below it to the top of it, like in some video games, these two new tilemaps will both have the Tilemap collider 2D, although i am reconsidering to change the Ground, One way grass rood and maybe even the spike damage tilemap to a different collider like, the box collider or edge collider 2D, because i heard that the edge collider 2D is good for something like the ground. The reason i wanted to add a one way collision to my game is because it's a nice feature to have but also the video reference i have also has that, so i want that in my game.

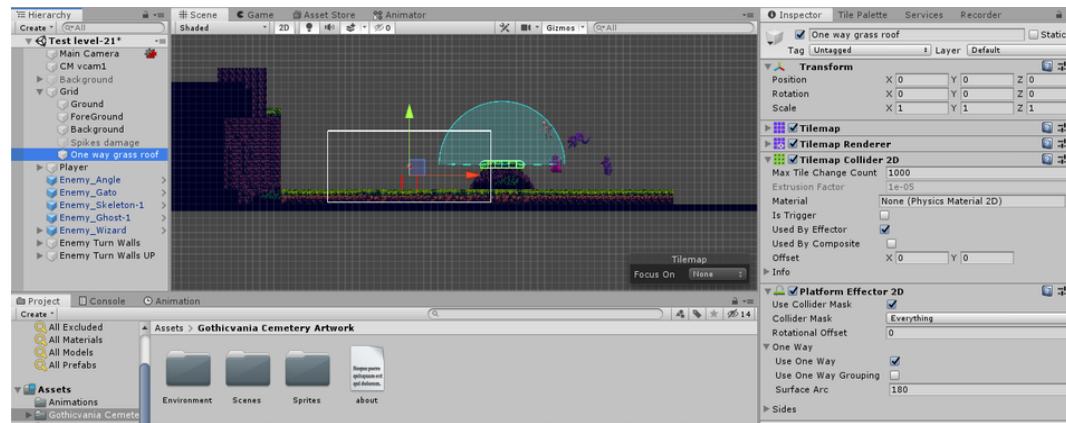
Later i went and entered play mode to see how my Player reacts to it's new level, the first thing that happened was a enemies started moving, i attack the Enemy_gato enemy and it died but for some reason the death animation didn't seem fully complete, only a white spark appeared before the enemy was destroyed, i noticed that this some times happens and sometimes it doesn't, if this becomes a bigger issue i might look more in depth into it. After defeating the enemies, i went to go on to of the big rock that was next to the player to the right, from below, and there was some problems i noticed like for example: the one way platform is in front of the player, when it should be behind the Player, the Player can't go through the sides of the platform and the Player can't jump when it's on top of the grass platform.

I then went and decided to add more stuff to my level and try to fix some issues i was having. To solve the one way platform issue i went and made the One way grass roof's tag and Layer into Ground, so the Player can jump when on top of it, i also decreased the Platform

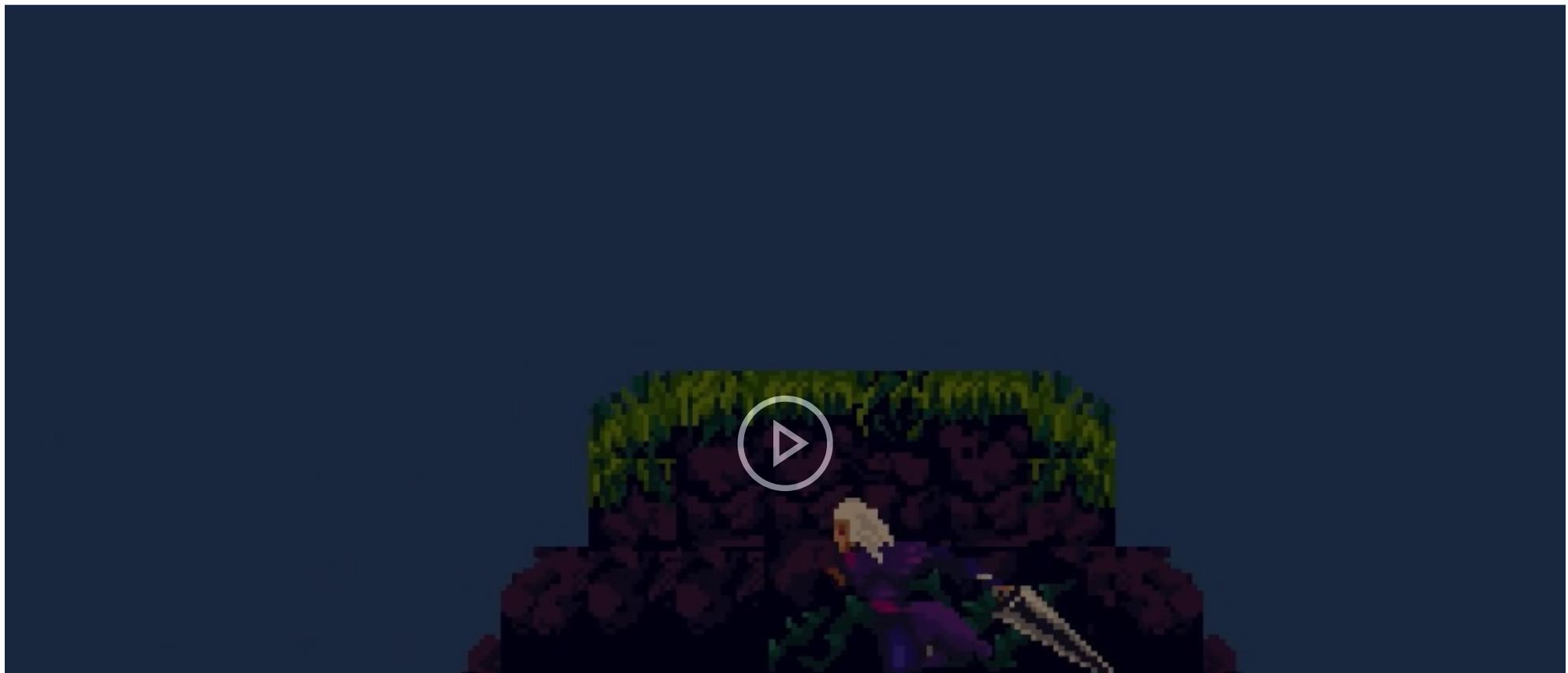
Effector 2D component angle, from 180 to 165 degrees, so that the Player can go through the sides of the platform. Another thing i did was add some spikes to see if they work against my Player, i also disabled my enemies so they wouldn't distract me when i'm testing my level.

After that i went to test the changes i made, the results were: the now way grass platform now lets the Player jump when on top of it, the Player can also move trough it's side but i now have a different issue, then the Player tries to go up the platform through the bottom, it can't go fully, it's like the Platform is stopping the Player as you can see from this video, i need to fix this since it feels and looks weird and people won't enjoy it. Another, somewhat weird, problem i encountered was when the Player touches the spikes the Player will be knocked to the right, i would have liked if the spikes pushed the Player back and not forward, other wise it looks weird, i tried fixing this by using a different collider like the box collider but nothing worked, i might be able to fix this if i can turn the spikes the opposite direction maybe that will make the Player be knocked back to the left. Another problem i found involves the ground tilemap, where i made a small inclination that would make the Player go up a bit but there was a problem where the Player couldn't go up, it would just stand there, the only way i could move a bit up was to jump but i wanted the Player to move up just by moving right, i need to fix these issues.

Then i wanted to add some 2D background objects to see how it will make my level look, if a few or many will make my level look pretty. What i did was make a new palette, so i can place the separate different sprites in my Tile Palette, placed it in the Palette folder i had and then i made a new Tile, for where i'll place the the sprites, after that drag the sprites to the Tile Palette inspector so i can place them in my scene easier. But when i did that i noticed some problems immediately, the sprites didn't seem to fit the boxes in the Tile Palette right, like it's too big or small, when i tried to use them in the scene the sprites were a bit off set, they wouldn't connect to the ground well, they were always a bit above the ground like it was floating or something, i was looking at the sprites and realized that the problem could be because the white box where it's the center is off, i believe i can fix this problem by going to the Objects Prefab palette, and make some adjustments or by making a new tile set or palette with these objects but with a change in the inspector window.

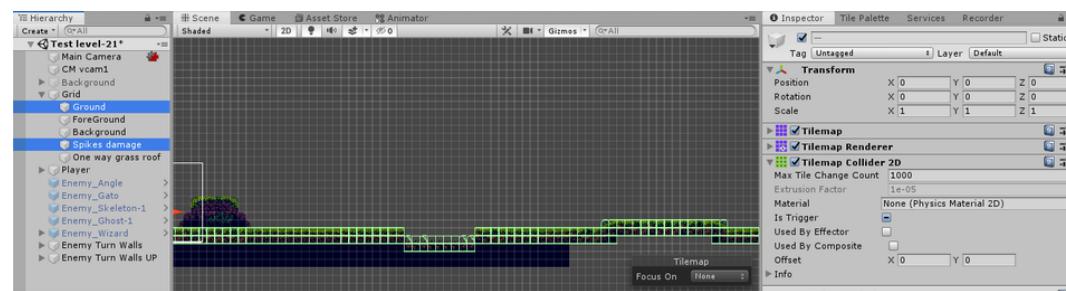


The two new tilemaps i created called "Spikes Damage" and "One way grass roof". I also deleted my old test level and started making a proper one.

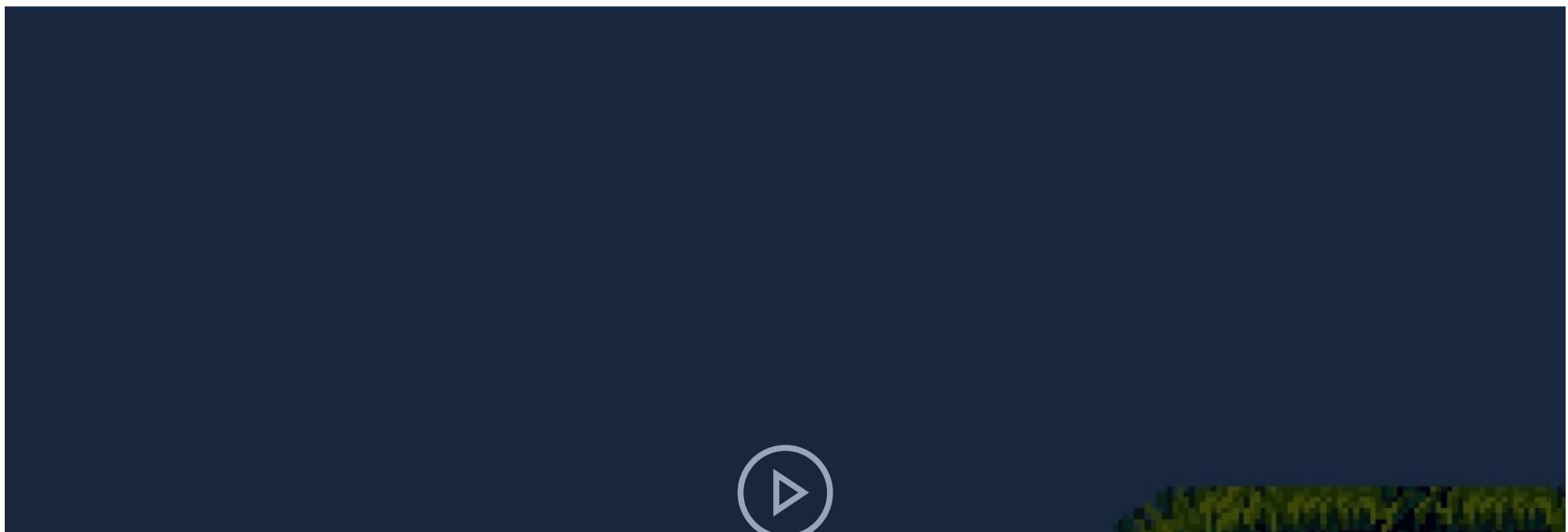




The test i did to see how my one way platform works and if there are any problems with it.

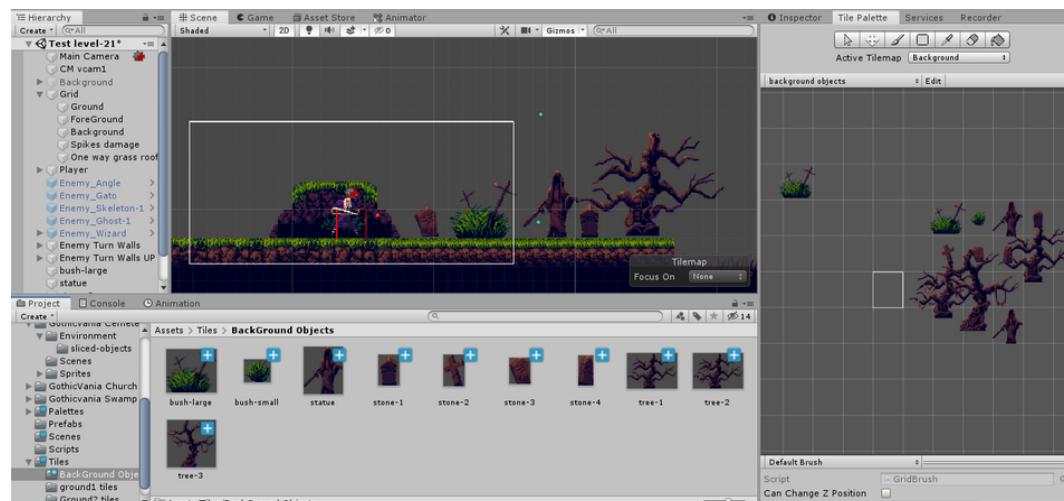


The additional platform and spikes i added to my game, in order to see how my Player reacts to them.





The test i did after making a few adjustments to my One way grass roof and the spikes i added to the level to see how the Player reacts to them now.



The new palette and tile i created with the 2D objects in this image. And the problem that i got when trying to add this to my scene.

28/04/2020: I was still having trouble with the object sprites not aligning right with everything else, like the ground, so i mess around with the objects a bit and thought about solutions, and then thought about the research i did and how maybe i need to change the Pivot of the Objects in order to make all of this work. So i went to all of my object sprites, the stones, bushes and trees, and changed the Pivot from center to bottom, the reason i did this is because i thought that it would fix the problem by adjusting where the object is in contrast to the pivot point, essentially moving the objects to be more up, so the pivot is below, or so i thought.

After i made the changes that i thought would fix the problem i was having, but turns out it just changes where the objects are in the Tile Palettes inspector and where the pivot box is, with your mouse, compared to the object, the bottom pivot puts the pivot at the bottom of the objects.

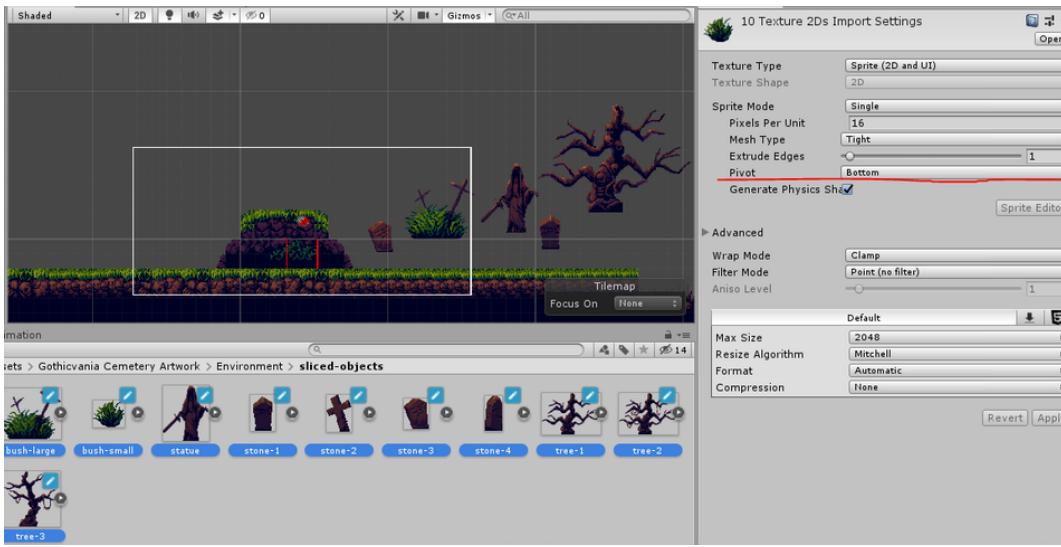
Later i still tried to fix the problem i was having, i even did quite some research about this case, i didn't find any solutions that were the same to my problems, only ones that have problems with how their objects or tiles look at the Tile Palette inspector, which i also had that problem but that doesn't affect anything in the game scene, while my objects are different in the scene compared to the other tile in my game. To fix the object sprites problem i was having i tried some thing i found in my research, i tried to make a new Palette but with the same objects just with some modifications and different values, and i even made a new Palette with one object to see if that solves any thing, but no, my 2D objects are still not aligning with the ground and everything else. I even tried to add more objects to the Tile Palette and deleting some but they all acted the same, i even noticed that if you delete a object that was from the Tile Palette in the inspector there will appear a sort of pink or skin colour square, which is interesting and weird but that won't help fix my problems with my game.

I then tried more solution to fix the problem i am having, using the research i did and what i can think of, but nothing was working. I thought about maybe if i can fix the issue i could just place the 2D object to the scene manually one by one, although that would take some time. I tried continuing to fix the issue i was having, i tried changing the Palette prefab of the objects sprites that i have in my Palette folder but nothing worked.

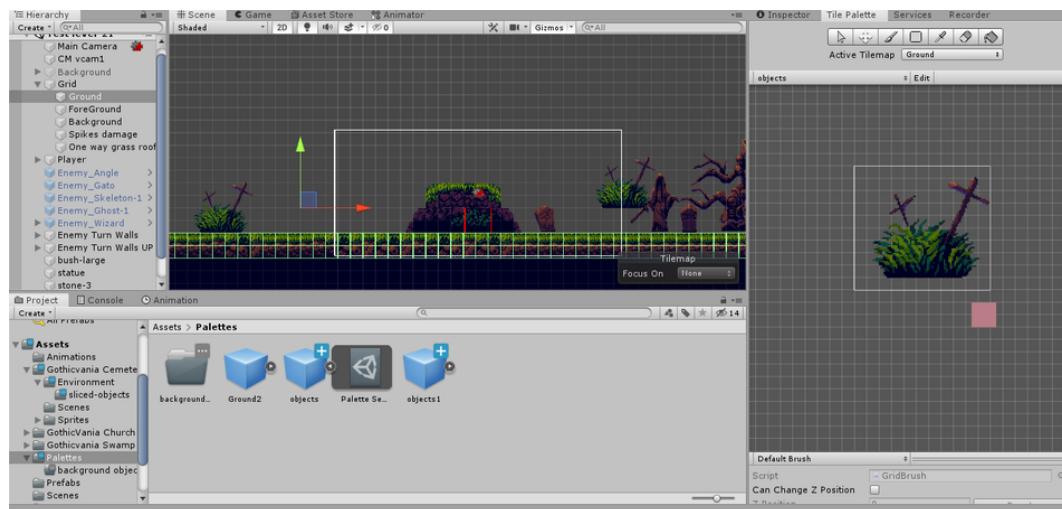
But then as i was messing around with the tile palette and trying to think of solutions, i thought about maybe i can move the grid itself instead of the objects itself, with that i could align the objects with the ground and other things without affecting the other tile, but i'll need a new tile where it will have a slightly different position from the others, in the Y axis. I decided to make a new tile called "objects BG", this will be used for when i want to place the object sprites in my level. After that i placed the object sprites in it's Tile Palette inspector, then i was placing a stone in the scene and it's finally fixed, the objects are no longer slightly above the ground or hovering, they look connected, which is perfect.

After that i, placed a whole bunch of object sprites in my level to see how they look, I entered play mode and moved the Player around and my level look much better now with object sprite in it, it make my level look less empty and more "alive", making the experience of playing my game more enjoyable. I also added more stuff to my level and i'm quite happy with the results.

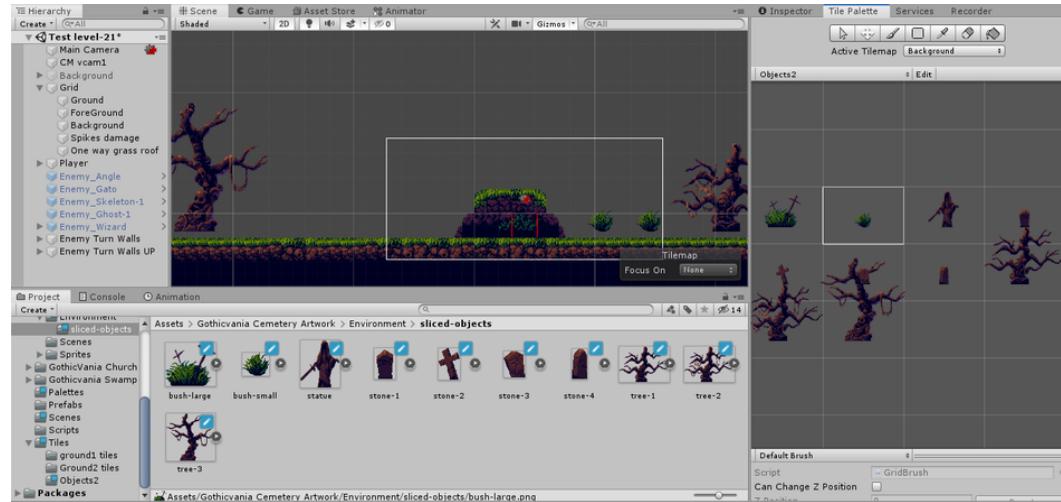
Turns out i just needed to change the tile's, that the 2D object sprites would be, Y axis form 0 to 0.5, now everything is aligned, making everything look better and natural, in terms of video games. Even though i fixed the objects problem i still had other problems i needed to fix such as the Platform Effector 2D not working as intended, some block being the foreground when it should be in the background of the Player and the Player getting stuck when in contact with a colliders side.



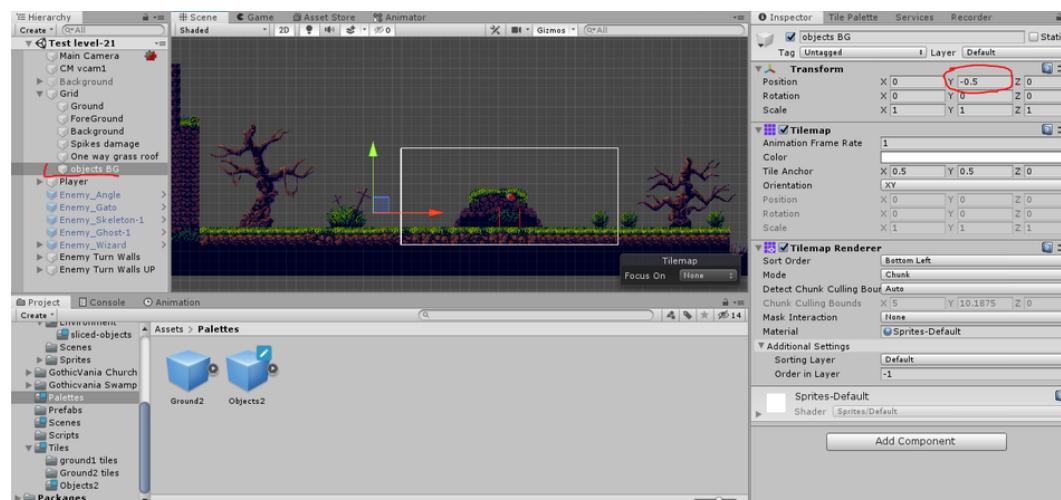
The change i made to all of my object sprites in order to fix the alignment issue.



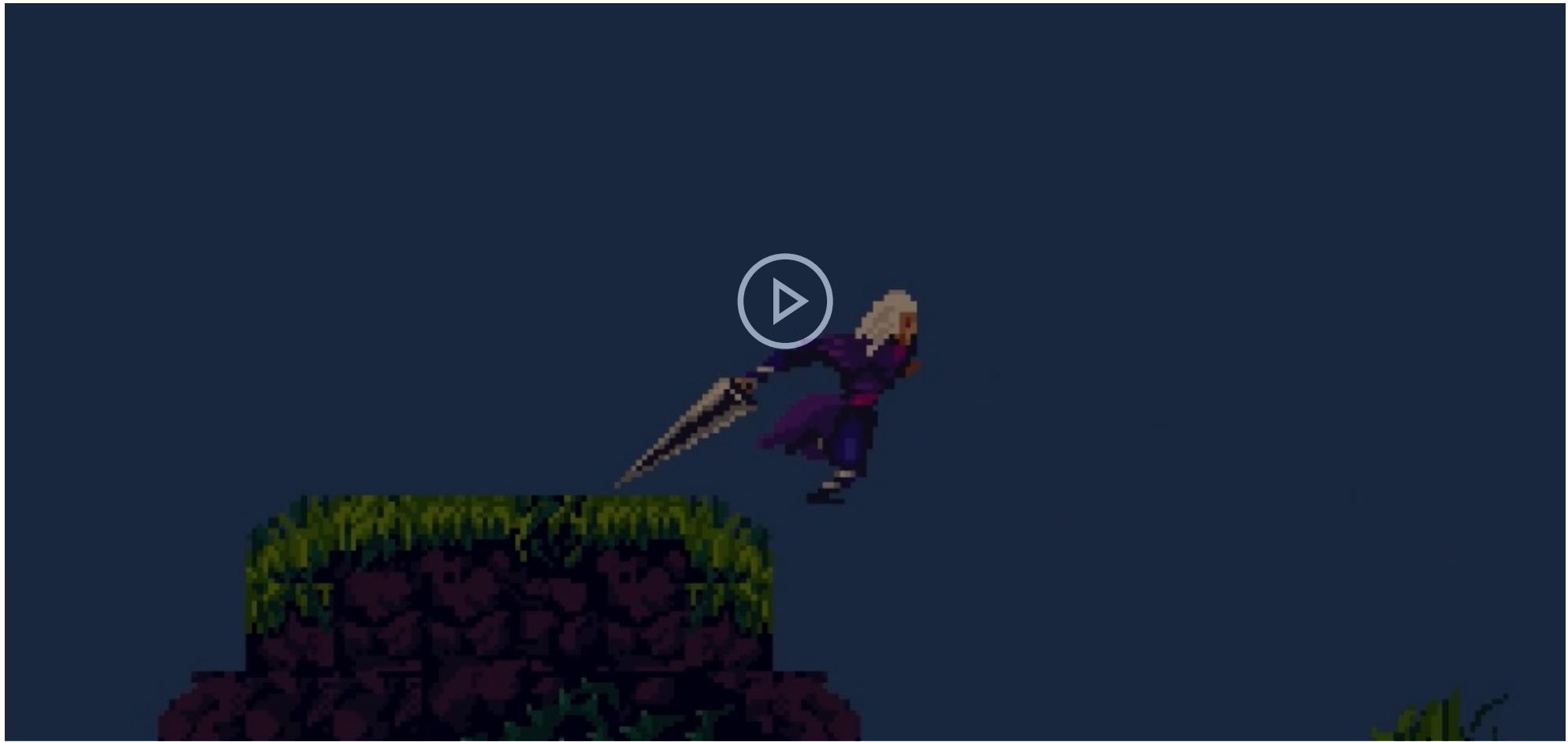
The additional Palette that i created with some differences in order to fix the problem i was having.



How my objects looked weird in the tile palette inspector, unlike the ground tile maps.



How i fixed the problem with my objects sprites and now they align very well with the ground, no longer hovering above the ground.



The test i did in my game to see how my objects look when the Player is running and if the object make my level overall pretty if people would enjoy w=seeing it.

29/04/2020: I wanted to fix the "One way grass roof" tile not working properly, for some reason the Player would get stuck when trying to go above the platform with the Platform Effector 2D component, and i wanted to fix that, so i tried some stuff, i added a box collider 2D and disabled the Tilemap collider 2D, to see if the reason for the problem was in the collider and if that would fix anything, but no, nothing was fixed, i did the same thing with the edge collider 2D, it had the same results. I did some research on Platform Effector 2D to see if i could find a solution but nothing that would help me directly, but i did find something that could help me in another area of my game and

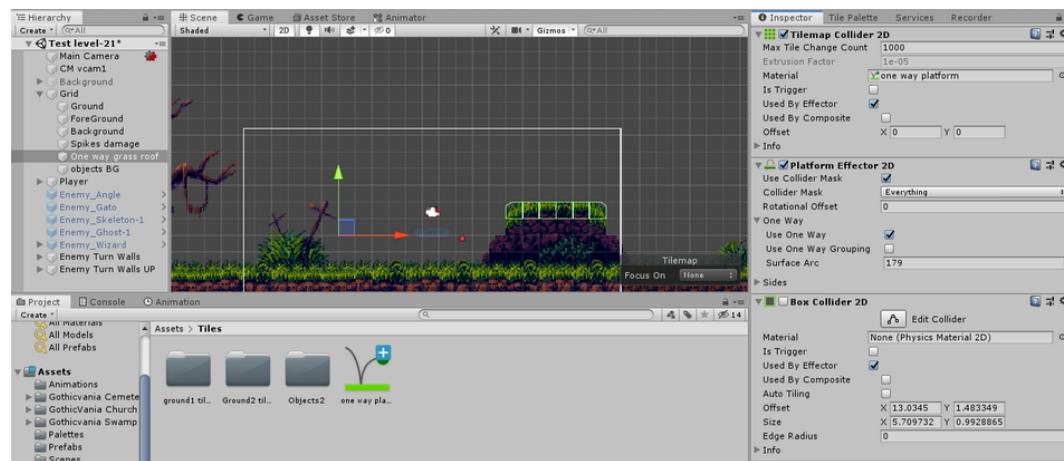
i found maybe a way to fix the platform effector problem but i doubt it will work. I tried making a 2D physic material with the friction set to zero, i testes my game and it didn't work, i tried to adjust the value of the friction and even bounciness and nothing worked.

I then continued to do research and trying to solve the platform effector 2D problem, i tried to do more changes to fix the problem, changing the colliders, what's on and off and modifying the platform effector 2D and more, but nothing worked, but i knew that in the past the Player could jump on top of the platform without resistance but now i can't, and then i realized something, the reason my Player is getting resisted when trying to go on top of the platform is because of the raycast, so i started messing with things that can impact the raycast in the One way grass roof inspector, and then i found out that if you turn the tile Layer to anything but the Ground Layer the Player will be able to jump through the platform from below it but the Player will not be able to jump when on top of the platform.

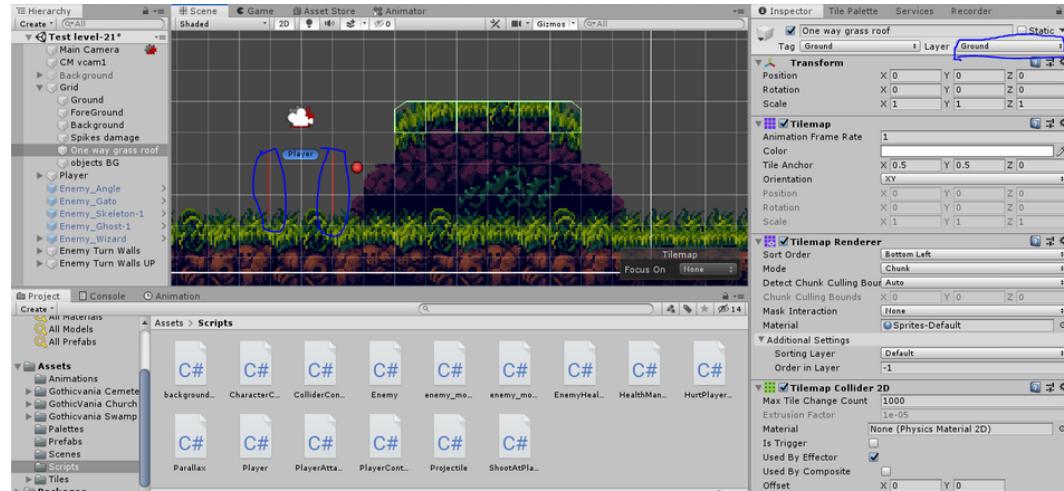
This has made me realize that maybe my game maybe doesn't need a platform that the Player can jump through below, with that i removed the One way grass roof tile map and all the things in the scene that would require that tile, since i no longer will be using it in my game. After that i began increasing my level again, with more objects, spike and other stuff to make my game better.

I then wanted to fix the problem with my spike, that make the Player get knocked forward when it should be backwards, i tried to change the spikes Y axis to 180, it didn't work, so i tried with different axis, nothing work, not even when i changed it's scale to -1 from 1. I wanted to see if this problem was with the spikes objects only, i made some top pillars have the effects of the spike and the results were different, instead of the Player would always get knockback backwards, as you can see in the video to the right, so i thought about adding theses as substitute for spikes, i'll would have just made the pillars invisible and place the spikes instead as foreground so it will look like the spikes are making the Player push back, but then i thought about it again and i decided to just make the spike do a lot of damage to the Player that they would instantly kill the Player, making people be actually careful when around spikes.

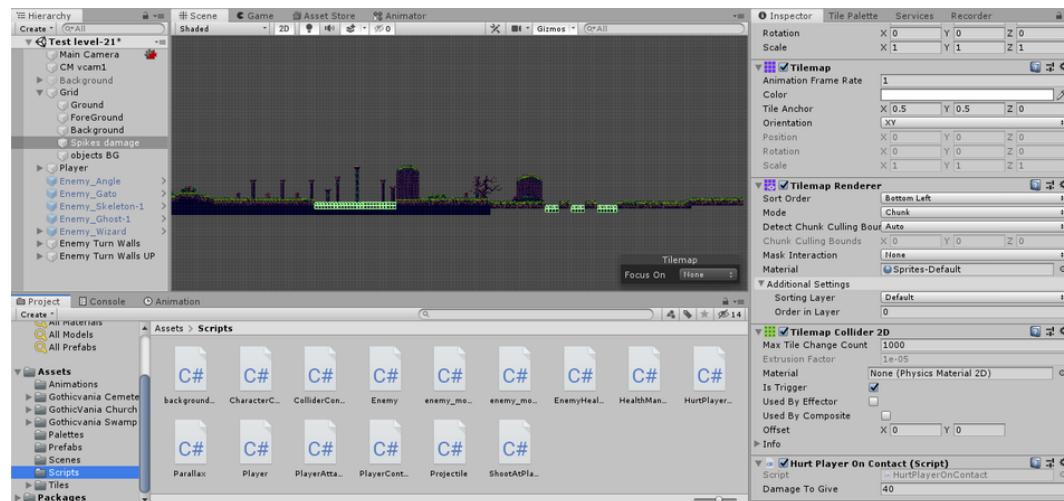
Then i wanted to test if my level looked good with the changes and additional things i added to it, i even enabled the background to see if it looked good on my level. I am glad to say that i'm happy that it at least looks good, in my opinion, but there are some thing i notice, my game looks a bit too big when i play my game in full screen and i don't quite know how to fix it, i tried doing some thing with the camera but nothing worked because my game is a 2D game, i believe i can fix this by increasing the Pixel per unit, but i'm afraid that i might ruin how everything looks, i'll need to think about it some more.



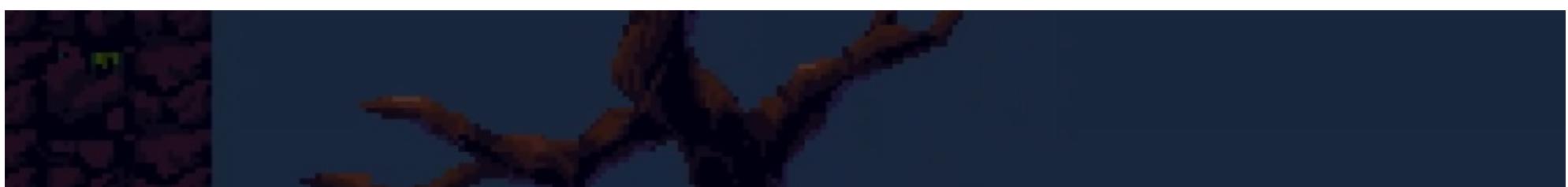
The changes i made to my One way grass roof tilemap in order to fix the problem with the Platform Effector 2D

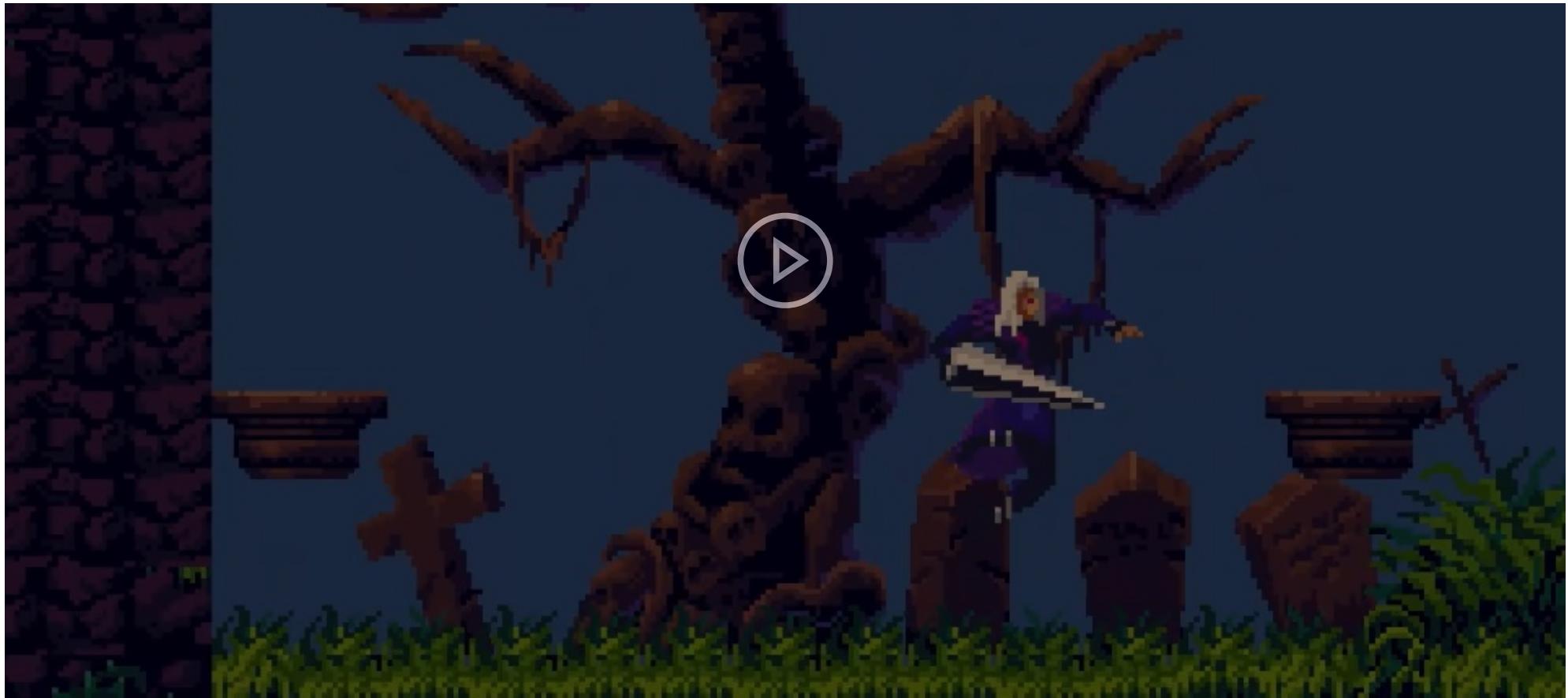


The cause in why the platform effect 2D won't work, in my opinion.



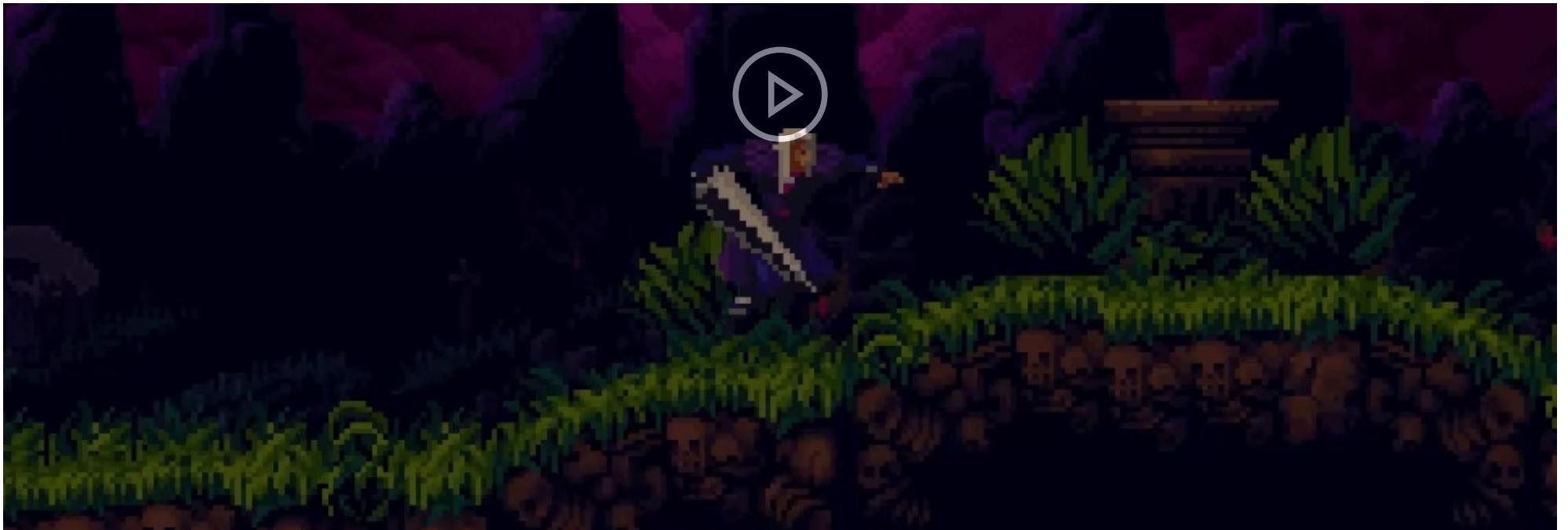
The additional ground, spikes, objects and more, i added to my level.





How the Spikes Damage tilemap does to some objects and how that effects the Player.





The test i did to see how my level is so far and how my Player reacts to it, with the background to see if it looks nice.

01/05/2020: What i did next was add more Ground and other things to my level, this will be the last time i will add something that will make my level longer, this will also be the final part of my level, the end of my level or game if you could say that. The red line shows you what i added to my scene.

Later i wanted to finally add enemies and new enemies with a small difference to every other enemy. I positioned them around my level to where i wanted them to be and to where i thought would make my level more challenging and interesting. The new enemies i created where the Ghost with a halo, a blue aura around it, and the skeleton with red cloth on it. These enemies will be the same as their other versions but will be slightly faster and will have more health. This will made enemies more diverse and will make my game more challenging and interesting.

I then added the death animation to the new enemies Animator tab, and i also changed the wizard enemy's Animator a bit so that the explosion death was more consistent and would happen independent if the enemy was attacking or idle. Everything had the right condition, transition and everything in between, i even corrected a mistake i made with the Gato enemy transition with the exit time.

Then i went testing and everything seemed fixed and doing well.

After that i wanted to fix a issue my game had for a long time and that was the issue, with the Player getting stuck when the Player jumps onto a wall or the sides of a platform when moving forward. I used a physic material 2D, thanks to the research i did about fixing issues with the Player getting stuck, with frictions set to 0, so that the Player would be slippery and not get stuck in wall and platforms. I tested this out and it does work, i even showed the difference between the Player with and with out the physic material 2D on it's box collider 2D. But even though it works there are still some issues i don't like about it, like for example, the Player now having a slippery box and the Player can still get stuck on the sides of the platforms, although it more harder to do it now, so even though it fixed the problem i want to use a different way.

I went to my Ground tilemaps inspector and added it a Platform Effector 2D and checked the Used by Effector on the Tilemap Colider 2D, i also removed the physic material from the Player, i later removed it from my Unity.

The video "Terraria Like Platformer Controller In Unity - Part 3 - Stuck Midair Bug And One Sided Colliders." was helpful in knowing about this method of solving this specific problem, i wanted to use this way because then i don't have to change the Player, ruining the code and components i have in it, this way it only changes the Ground tiles.

Then i went to testing if what i did actually works, i happy because it does work fine, the Player no longer gets stuck when moving on to a wall or the sides of a platform, but the Player can still get stuck on the sides of a platform, although very hard to do now, if you move into it in a certain height and position, i thought that maybe this was because of the box collider being larger then the circle collider, so i made the box thinner, then tested again but the Player could still get stuck. I decided to just leave it since i don't have time to waste nor do i know how to fix the problem, also it's quite hard to even get the Player stuck on the sides of the platforms now so it's not that important to fix the issue.

After that i wanted to add music and sound effects to my game so i downloaded and imported those from the asset store, i downloaded classic explosion sound effects and 8-bit music. All of this will make my game more alive and better overall.

Then i wanted to add some extra code to my HealthManager script so that the Player make a sound when hit by an enemy, but this doesn't work because there was a error with the `HurtSound.Play();` line of code, i couldn't even play my game with this code on my script. This didn't work, not even with the video "Play sound upon Collision in Unity" as a example, turns out it's not as simple to make the Player do sound effects when something happens, even though to make the camera play a background music was quite easy, you just have to add a AudioSource component and drag your music to the empty slot and that's it, very easy.

I decided to remove the previous code i had on my HealthManager script and create SoundManager, i'm following a video called "How To Add Sound Or Audio Effects SFX To Unity 2D Arcade Game | Very Simple Unity 2D Tutorial." This C# script will be used to make me be able to get the sounds with the same name as the ones in quotation marks in red in the script, i'll need to rename a few sounds. I'll also need to

make a folder called "Resources" so i can use the sounds in it, i will also need to make a Sound Manager empty object that will have this script.

This image has the empty game object i created called "Sound Manager" that will carry the script and Audio Source to play music, this also has the Resources with the sounds effects i want in my game. I also downloaded another thing from the asset store and that was old bit sound effects of hit and attack.

After i made the SoundManager script i went to my Enemy, Player and HealthManager scripts and added a single line of code that would make a specific sound play with the designated name, when something happens, for example, in this image it shows you the line of code that will make a sound when a enemy, with the Enemy C# script, gets hit, it also shows you the one where if the health is zero or below a sound will play.

Later just about when i was going to test my game, i thought about add a sound to when the player jumps as a lot of video games do this, so i thought about add it as well to my game, i made a extra variable in my SoundManager script called playerJump, i then added the codes in this script accordingly, after that i choose a sound and renamed it to playerJump and placed it in my Resources folder.

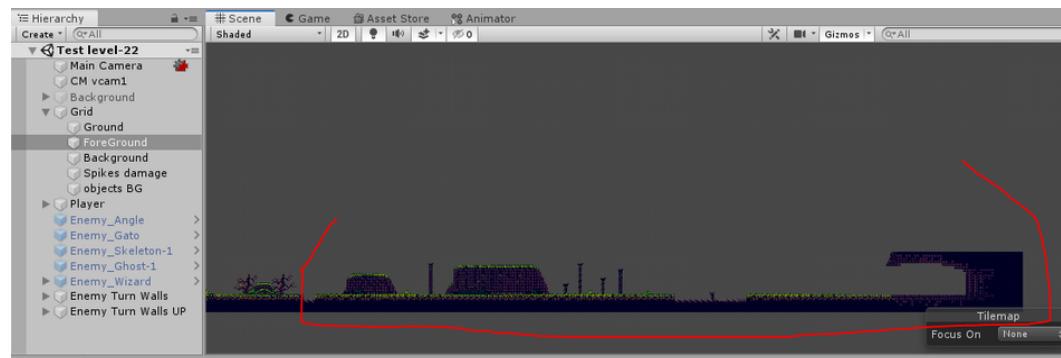
After i did all that i went to testing and i'm glad to say that my game sound good in my opinion, but there could be a few adjustments here and there i could play around with, mostly the volume and quality of the sound. But there is a problem one of the sounds have, when a enemy dies the enemyDead sound will play but it will have a weird repeating sound, like it's trying to do the sound multiple times very fast of something.

I tried to fix the enemyDead sound effect but i could fix it, and i didn't want to wast too much time in this, so i decided to delete the code involving that.

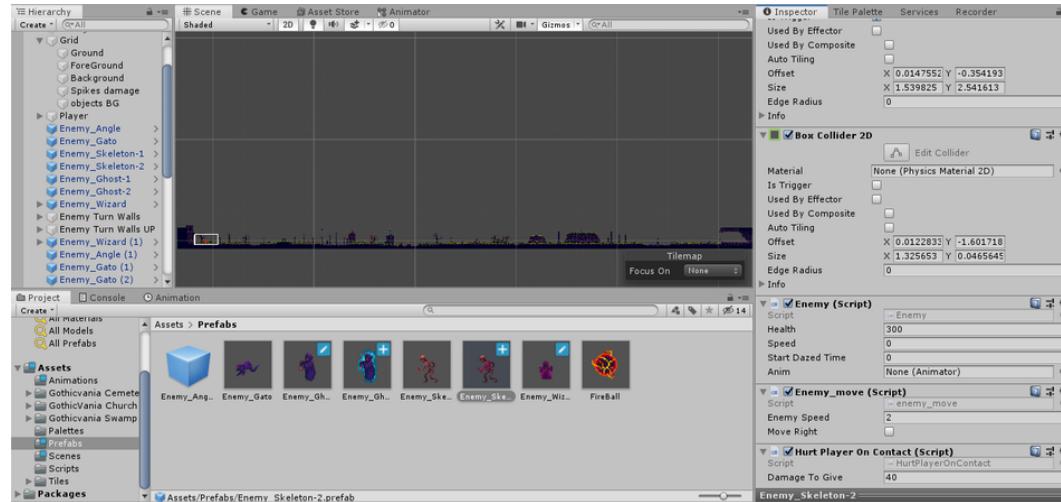
After that i decided to properly test my entire game, from the beginning to the end, from the fat left to the far right, and see how i feel about my level. After i did my level i knew i needed to do a few adjustments, i also feel like there are a bit too many enemies and near the

end of the level i need to do a few small changes to the level as not to be too hard.

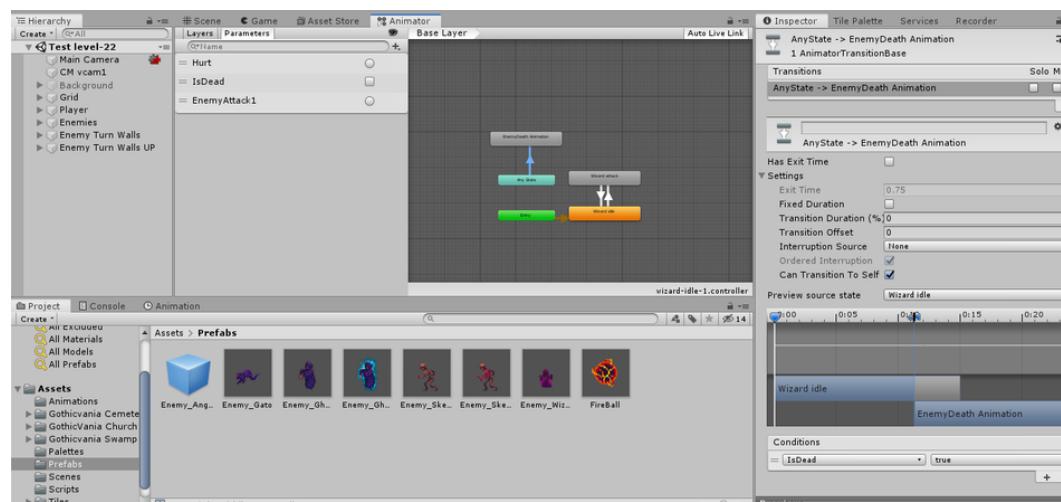
What i'm going to do next is ask one of my family to play my game and ask them to give me their honest opinion about my game, so i can better my game for me and for everyone and so i can learn from this experience. After that i might make a few adjustments to my game but then i will start improving and polishing my SharePoint and do the Evaluation of my project and then add my game to the Final Product page in my SharePoint.



This is the additional stuff i added to my scene, the final part of my level, the end.



The enemies i added to my scene and the two new enemies i added that are lightly different then the... other versions.

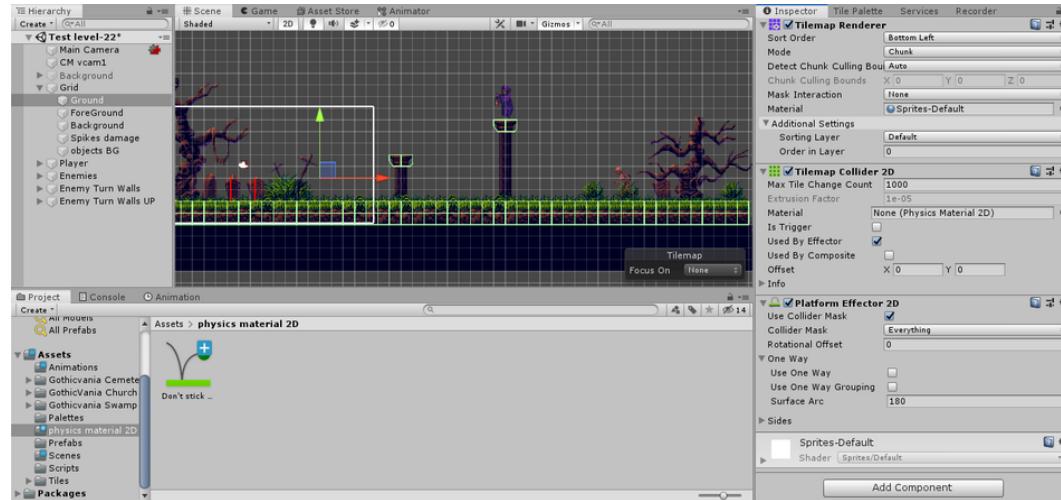


The death animation i added to the new enemies, in the Animator, and the change i did to my Wizard enemy so that the death animation works all the time, more consistent.

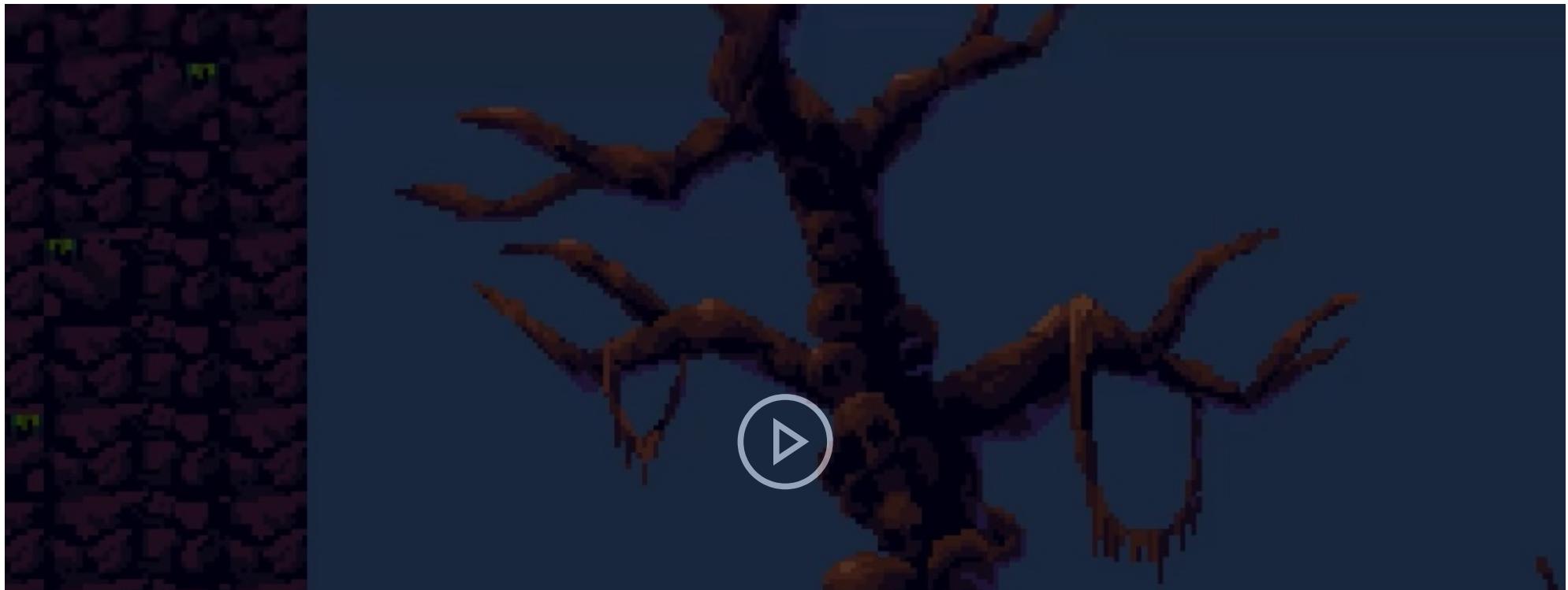




The change i made so that i can fix the problem of he Player sticking to the walls and platforms, and me testing the difference between before i added the physic material 2D to the Player's box collider and after i added.

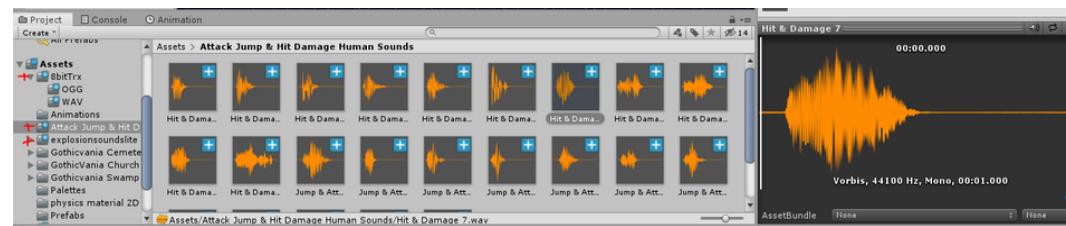


The Physic material 2D i decided to not use and the Platform Effector 2D i added, with the Tilemap Collider 2D, to my ground tile so that my Player doesn't get stuck when touching the walls or sides of platforms.





Testing if i fixed the problem of the Player getting stuck on wall and platforms after doing the changes with the Ground tilemap.



The new music and sound effects i added to my game thanks to the asset store.

The screenshot shows the Unity code editor with the tab bar at the top containing "HealthManager.cs", "Player.cs", "Enemy.cs", and "PlayerAttack.cs". The active tab is "HealthManager.cs". Below the tabs is a dropdown menu showing "Assembly-CSharp" and "HealthManager". The main area displays the C# code for the HealthManager script:

```
4
5     public class HealthManager : MonoBehaviour
6     {
7         public int maxPlayerHealth;
8         public static int playerHealth;
9
10        public AudioSource HurtSound;
11
12        // Start is called before the first frame update
13        void Start()
14        {
15            HurtSound = GetComponent<AudioSource>();
16            playerHealth = maxPlayerHealth;
17        }
18
19        // Update is called once per frame
20        void Update()
21        {
22            if (playerHealth <= 0)
23            {
24                Destroy(gameObject);
25            }
26        }
27
28        public static void HurtPlayer(int damageToGive)
29        {
30            HurtSound.Play();
31            playerHealth -= damageToGive;
32        }
33
34        public void FullHealth()
35        {
36            playerHealth = maxPlayerHealth;
37        }
38
```

The additional code I added to my HealthManager script in order for my Player to play a sound when it gets hit, unfortunately it doesn't work.

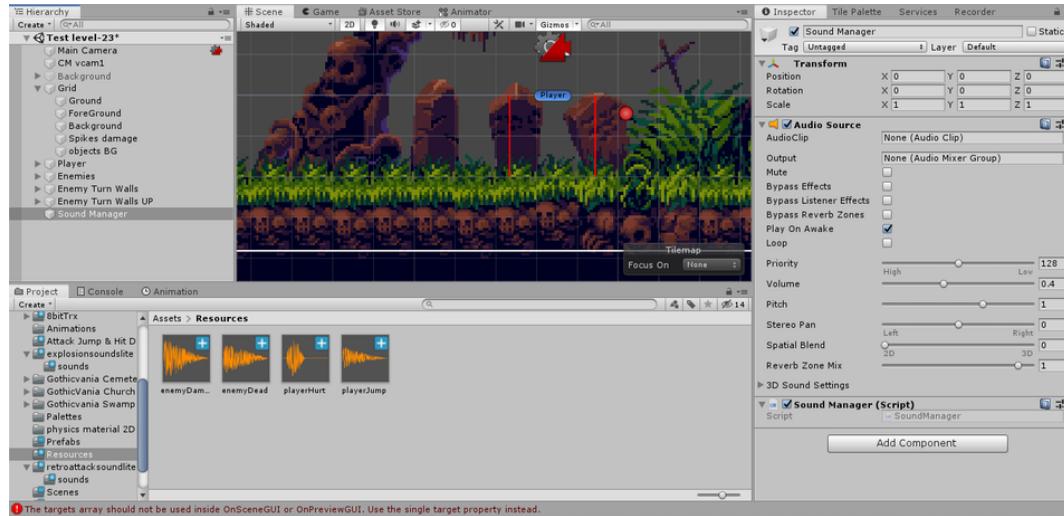
The screenshot shows a Unity code editor window with the tab bar at the top containing "SoundManager.cs", "HealthManager.cs", "Player.cs", "Enemy.cs", and "PlayerAttack.cs". The active tab is "SoundManager.cs". The code editor displays the following C# script:

```
public static AudioClip playerHurt, enemyDamaged, enemyDead;
static AudioSource audioSrc;

// Start is called before the first frame update
void Start()
{
    playerHurt = Resources.Load<AudioClip>("playerHurt");
    enemyDamaged = Resources.Load<AudioClip>("enemyDamaged");
    enemyDead = Resources.Load<AudioClip>("enemyDead");

    audioSrc = GetComponent<
```

The new C# script i created called "SoundManager" that will make it possible to for sound to play when the Player does something or when something happens in my game.

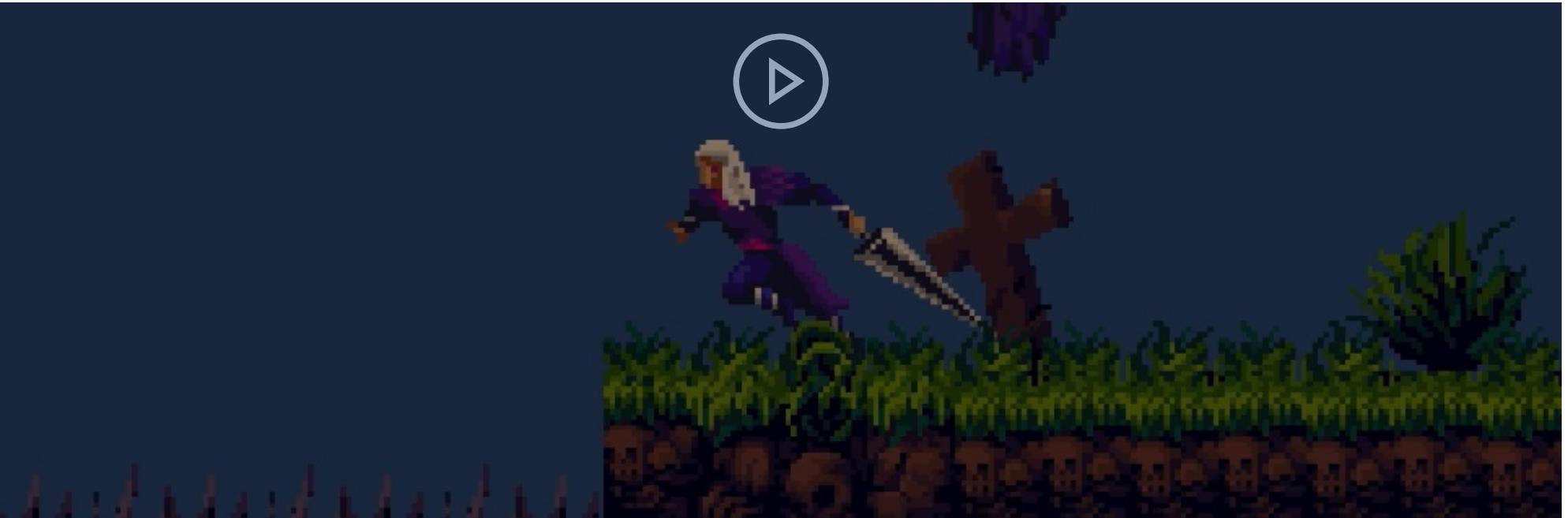


The Sound Manager empty object, with the SoundManager script, that i created and the Resources folder i created with the sound effects i want in my game.

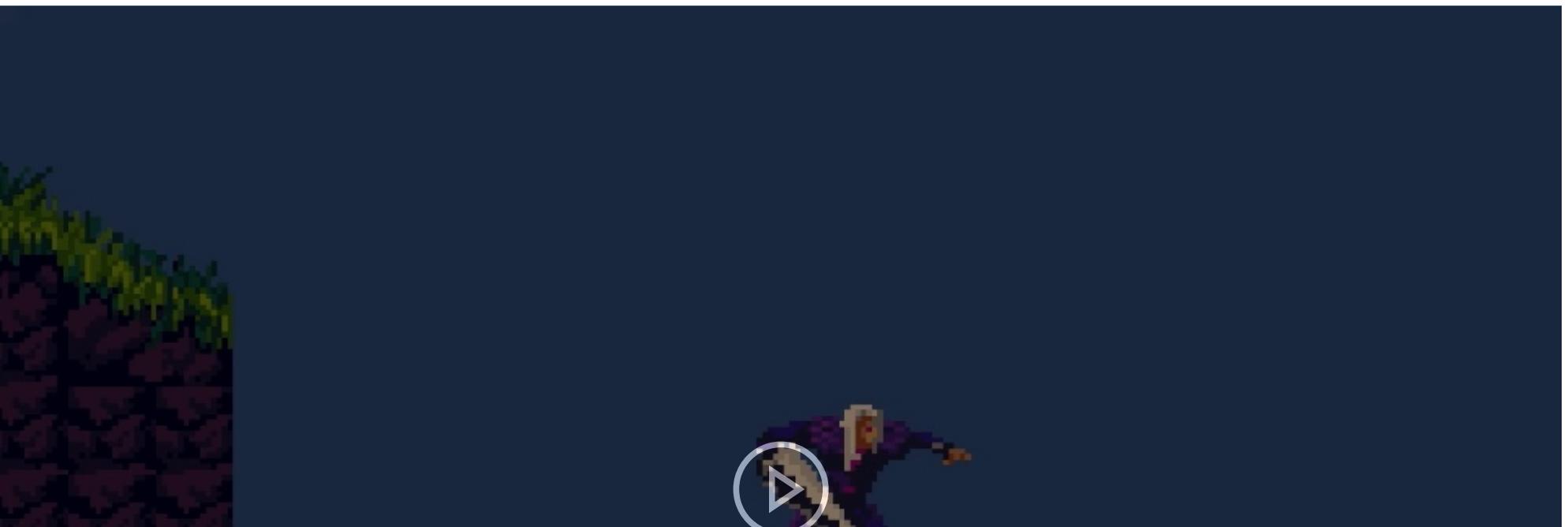
```
SoundManager.cs      HealthManager.cs      Player.cs      Enemy.cs      X      PlayerAttack.cs
Assembly-CSharp      Enemy
32
33      if (health <= 0)
34      {
35          Destroy(GetComponent<enemy_move>());
36          Destroy(GetComponent<enemy_move_up>());
37          Destroy(GetComponent<ShootAtPlayerInRange>());
38
39          GetComponent<Collider2D>().enabled = false;
40
41          SoundManager.PlaySound("enemyDead");
42          anim.SetBool("IsDead", true);
43
44          Destroy(gameObject, 0.45f);
45      }
46
47
48      public void TakeDamage(int damage)
49      {
50          SoundManager.PlaySound("enemyDamaged");
51
52          dazedTime = startDazedTime;
53
54          health -= damage;
55          Debug.Log("damage TAKEN!");
56      }
57
58  }
```

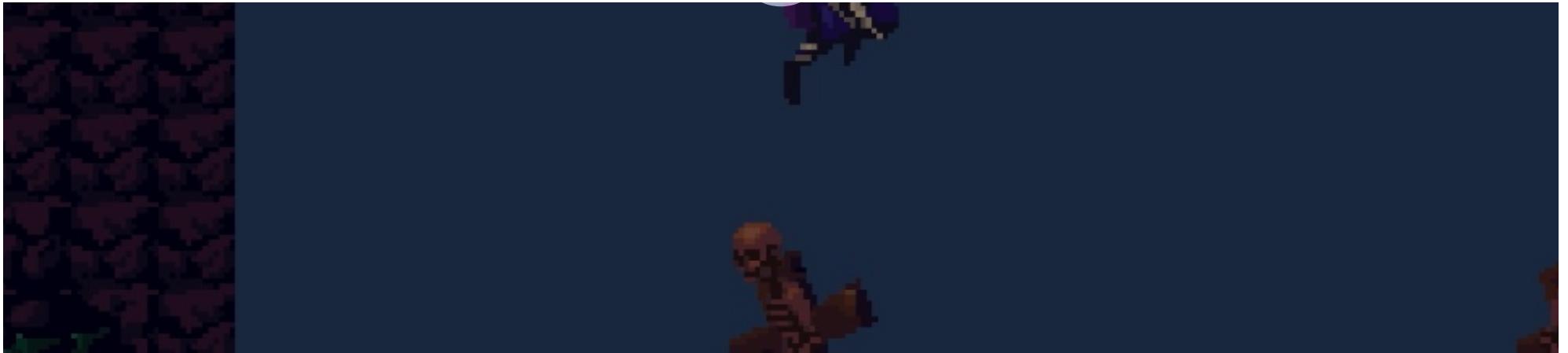
The changes i made to my Enemy, Player and HealthManager C# scripts, so that sound effects plays when these scripts do somthing.





A video with the music i added to my game, sound effects and everything, although there is one problem involving the sound effect with the death of a enemy.





The test i did after removing the death of a enemy explosion sound effect, and then i just played my game from where you start to the end, from the left to the right.

04/05/2020: Since the deadline is near the end i just want to make the final adjustments to my game, by asking help from my sisters and make changes based on that feedback.

I asked one of my sisters to play my game, i have build it and placed it in a folder and i double clicked my games icon and the game started. I told my sister what she can do to the Player, i told her she can use the arrow keys or the WASD keys to make the Player move, crouch and jump, i also told here she can jump with the space button and attack with the K key or with the left mouse button, i also told here that she has unlimited health since i believe that i made the level quite hard. After that i kept quite most of the time she played my game, only talking once in a while to help her a bit, she first started moving and and went back to the left to see if there was anything there but she couldn't go through, then she tried to go on top of a platform pillar but she was trying to go from below the pillar to the top of it, trying to go through it even when it wasn't working she thought that the jump mechanic wasn't working when trying to do that, but then she later realized that she could just jump around the pillar. After that she tried to attack the Enemy_Ghost on top of the second pillar next to the first pillar she's on, but she had difficulty in trying to do that, she kept falling off the pillar she was on and kept getting hit by the ghost, but then she eventually hit the ghost and killed it. Anther thing i noticed was that my sister kept using the mouse for attacking instead of the K button, i guess she just prefers that way. When my sister kept moving to the right with the Player she kept having a hard time attacking enemies and was sometimes stuck in certain situations, at one point she asked me what was her objective and if she needed

to kill all of the enemies, with this knowledge i came to believe that some people prefer to have objectives or be forced to do something rather than figure something out with no clue or directions to what to do.

Another thing i kept noticing was that she was always going head into the enemy while attack but then would get hit and be knocked back, she would do this multiple times, she did this maybe because my sister isn't experienced in playing video games or maybe because playing a game with only one melee attack is hard with rapid movement from the Player i'm not quite sure, all i know is that she had difficulty killing enemies and not getting hit. One thing she kept noticing was the high places that you can go to but you need to go a specific way to go there, she initially wanted to go to those places but end up giving up and just continued moving forward. One thing i noticed when she was play and that was that the Wizard enemy wasn't shooting the fireball projectiles, when my sister encounter her first enemy wizard it shoot a fireball once but the rest of the enemy wizards didn't shoot a fireball at all, i believe this is because of the error when i keep getting when the destroy delay of the fireballs is starting or even when my game is trying to destroy the fireballs. She kept having trouble with enemies and moving on platforms but eventually overcame all of the problems that came her way, Then she reached the end of the level, the cave where you go and all you will see is black. I then clicked on the windows key of my computer and exited my game that way, since i didn't add a mechanic that if you click the esc button or reach a certain point you exit the game, i did want to add that mechanic to my game but i was running out of time and i needed to do this, and the teachers are going o play my game in a small windowed mode in my Final Product in SharePoint, which won't be the best quality, i now regret having my sprites this big but it is what it is, i just have to learn from my mistakes.

After my sister played my game i asked her for her feedback, i asked her what she liked, disliked and what could improve in my game. She liked how my game looked and the animations, what she didn't liked was how hard the enemies were and how the player moves and how there is not objective. One thing she said she would like my game to have was a objective like collecting coins or something.

The feedback that my sister gave me has helped me tremendously, it helped me gain perspective in what people might like and what they notice, especially since this sister of mine isn't quite what people would call "a gamer", she has played games before but not as serious or as often as i have. But still this knowledge i gained was useful, without it i wouldn't have known some problems and error that my game has and the fixes it needs.

I first wanted to fix the problem of my Player not doing the animations properly, i went to the Player's Animator and added a new transition from the Jumping2 to the Idle animation, and then i added a few more conditions to other transitions, i mostly added the isGround is true conditions so that my Player does the animations it need to do when it on the ground or in the air.

Later i modify my level a bit, i took out some enemies because i believed it was overcrowding my game and making it a bit too hard, then i adjusted them so they are more balanced. I then changed the level a bit, i moved some ghost enemies so that it easier to kill them, i also made their movement wider, it takes them longer to go to the top and then down, giving the Player a wider gap of safety and movement. Later i made some adjustments to my Player, i made it's attack longer, made the fall multiplier weaker and it's movement slower. I also made some parts of my level easier to get to and some areas easier to traverse, especially the pillars.

After that i tested my game and i believed that it's better then it was before, it's much simpler and easier to play my game now, although it could use a few adjustments here and there.

I deleted my old build of my game and made a new build with these adjustments, then one of my other sisters wanted to play my game, this one is younger then the first one that player my game. I opened my build of my game with the recent updates and let her play it, before she could play i told here her what here controls were, i told her she could move with the WASD or arrow keys and she could attack with the left mouse button or the K button, i also told her she can jump with the space or W button, and she could duck with the S or down arrow buttons.

She first started to get a feeling of the controls and which button did what, in the end she moved with the WASD keys but jumped with the space bar and she attack with the K key, which i find interesting since the other sister attack with the mouse and jumped with the W key. After my sister started moving with the WASD keys and jumping with space and attacking with the K button, she started moving and she saw a ghost moving up and down and wanted to attack it, instead of doing to the pillars she waited for the ghost to move low enough for her to jump and attack it, and she accomplished that, she then moved on she killed the skeleton next and then the angle, but she asked me if the angle was to be attack also but later attacked it nonetheless and killed it. It would seem that the angle didn't seem threatening so she questioned if she should attack it. Later in the multiple pillars she went down and was surprised by the Gato enemy, i believed that's because of how big the Player and everything is, and how little the Player can see around it leaving little time for reaction if something appears. But my sister overcame it, but she was having difficulty on where to go now, she didn't know if she should go back and jump on top of the pillars or keep moving below it and move to the right, she decided to move to the right and then go back while on top of the pillars to kill the ghosts. Then she continued moving to the right and killing enemies on the way, then she encounter a wizard it shoot a fireball and then she killed it, but then she encounter a second wizard and this one didn't shoot any fireballs, which i found weird because

when i test my game in the play mode in Unity it works fine i don't know why in the build game it doesn't work, possibly because of the error involved in the fireballs but i'm not sure, another thing that happened was that the death animation for the wizard enemy didn't happen which should happen, i'll need to fix this in unity.

The next thing my sister did was that she realized the ghost in the halo, on top of the second enemy wizard, and tried to go up where it is but couldn't because she missed her opportunity, so she moved on killing enemies and such. Then she encountered the skeletons on the top of the hill and she wanted to get up there with the skeleton and kill it but the height was a bit too high for here so she eventually gave up and moved on, but then she found the multiple pillars next to it to the right and destroyed the wizard and angle that were on top of the pillars and then was able to get to the angels and skeletons on top of the hills, this made here happy as she wanted to get there and she did after finding about the pillars, then she moved to the final part of my level, kill the rest of the enemies and and went to the cave where i told her she completed my game. The only enemy she missed was the first ghost with the blue halo that she missed her opportunity to go and attack it.

After all that i asked her what she like, didn't like and what she would like i add to my game if i could. My sister said that she liked how my game looked and how the Animations moved, what she didn't like was how some times the death animation, the explosion that enemies do when they are destroyed, doesn't appear, it ruined her immersion a bit. One thing she would like my game to have is a option in the start of the game to choose different characters, different Player she can play from with different looks and maybe different mechanic or abilities.

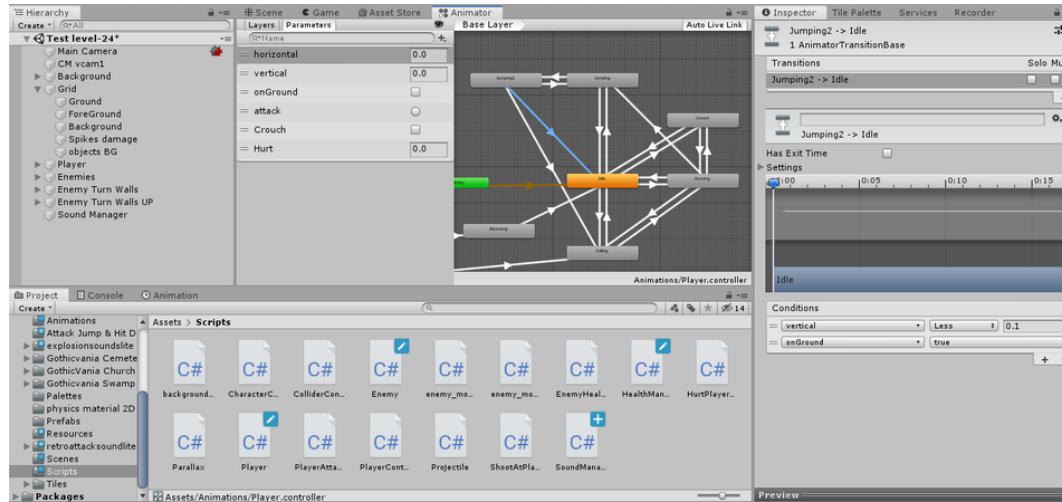
From the start when this sister was player my game she knew how to play a game a bit better then my other sister, she had a much easier time playing my game, but it wasn't all because she was good at playing games, it was also because i made my game easier after my first sister played my game. Even though my second sister still gave me very useful information and perspective on her opinion on my game and how i can improve it for the better, she saw flaws i couldn't see, which is good to know, if i had more time in this project i would have more people play my game and get more feedback so that my game can become the best it can be, but i'll have to deal with the short time i have and do my best work.

After my sister tested my game build i went to my Unity and started improving my game with small adjustments, i improved my ground, especially the small inclinations and declines where the Player could get bumped and be stopped, i added some Edge Collider 2D to my Ground tiles and adjusted them so they would substitute the corners so the movement to them from the Player would be smooth and not weird.

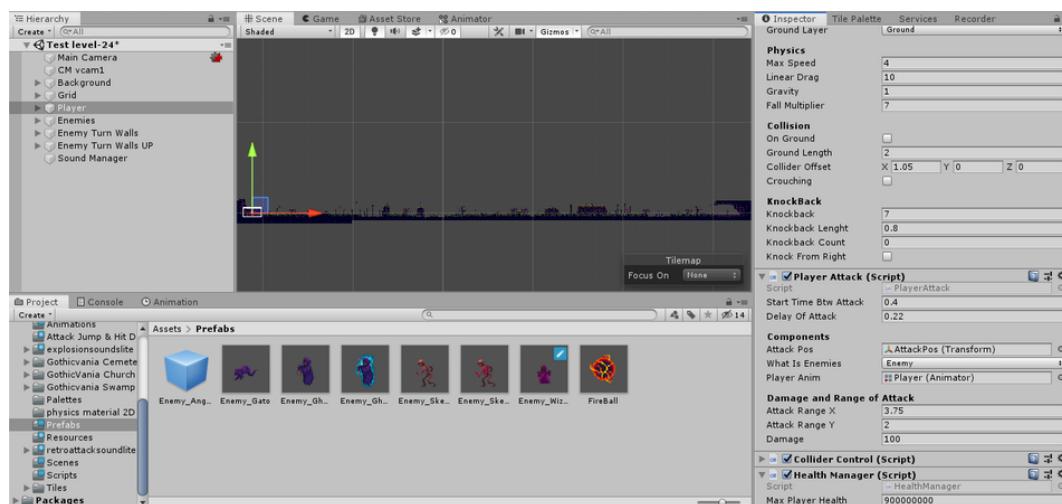
I then went to the ShootAtPlayerInRange C# script and deleted the Destroy(fireball, 0.3f); code so that i don't have that error anymore in the console. I thought about adding a different code that would disable the projectile script but this was a much simpler solution and the fireball would eventually be destroyed by any collider it would touch so all of my problems are solved now, hopefully my wizard enemies now shoot fireballs in my build game.

After that i made a few small modifications to my game so that it's slightly better then before. I am quite happy with the results and the changes i made to my game, because it's all for the betterment of it and for me as a future game programmer. I then tested my game in Unity as well as the built version, and i'm glad to say there are no problems with my game right now, there is a few very minor issues that i could try to fix but i don't have time, for example the jump sound can be spammed if you press the jump button multiple times, but everything else is ok and there are no major problem.

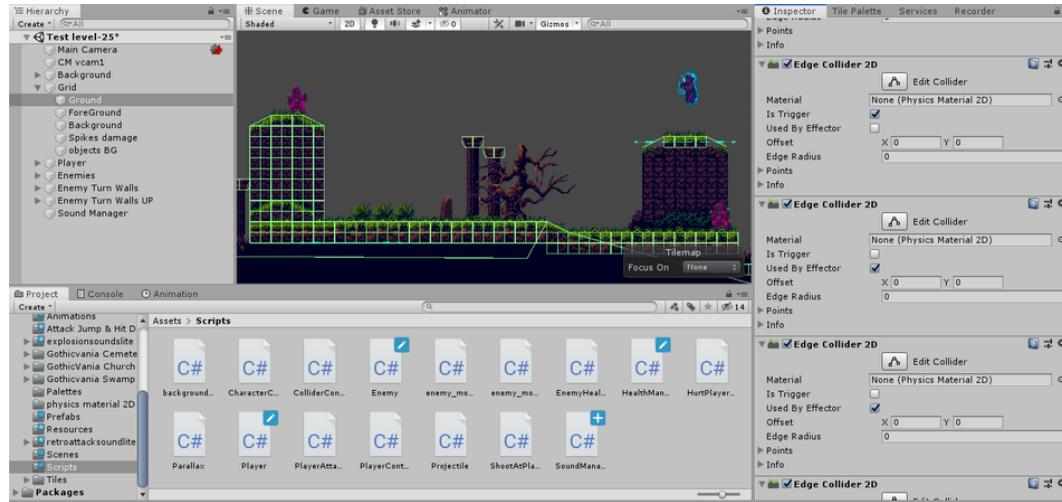
I did the choices i did mostly because of my mind, the feedback i got and the lack of time i have now. What i'm going to do in the future now is clean up and organize my sharepoint, and then do a final ongoing reflection and then do the Evaluation and place my game in the Final Product tab in sharepoint and hopefully that is all.



The changes i did to my Player's Animator so that the Player won't look like it's doing the jump animation when in fact it's just running normally in the ground.



The additional changes i did to my game, it now has less enemies, minor changes to the level and the game is less harder to play now,



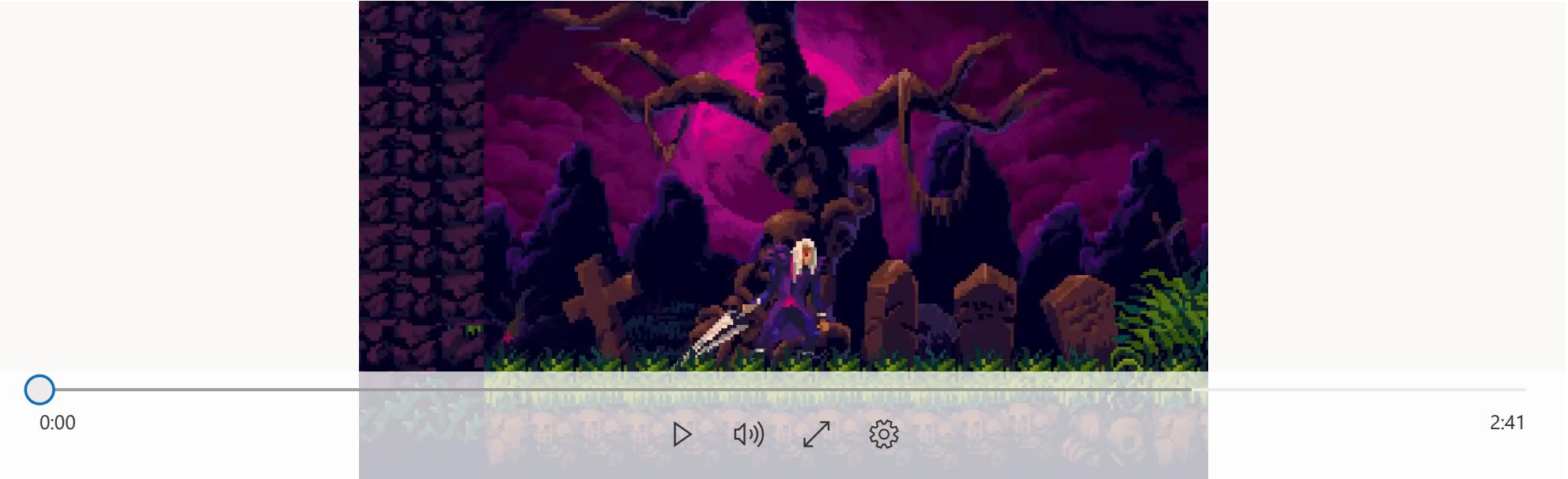
The modifications i made to my game, mostly add some Edge Colliders 2D so that the inclinations, the ups and downs, of my game are easier to move on.

```

25 // Update is called once per frame
26 void Update()
27 {
28     Debug.DrawLine(new Vector3(transform.position.x - playerRange, transform.position.y, transform.position.z), new Vector3(transform.position.x + playerRange, transform.position.y, transform.position.z));
29     shotCounter -= Time.deltaTime;
30
31     if (transform.localScale.x < 0 && player.transform.position.x > transform.position.x && player.transform.position.x < transform.position.x + playerRange)
32     {
33         enemyAnim.SetTrigger("EnemyAttack1");
34         StartCoroutine(EnemyDelay());
35
36         shotCounter = waitBetweenShots;
37     }
38
39     if (transform.localScale.x > 0 && player.transform.position.x < transform.position.x && player.transform.position.x > transform.position.x - playerRange)
40     {
41         enemyAnim.SetTrigger("EnemyAttack1");
42         StartCoroutine(EnemyDelay());
43
44         shotCounter = waitBetweenShots;
45     }
46
47 }
48
49 IEnumerator EnemyDelay()
50 {
51     yield return new WaitForSeconds(delayOfEnemyAttack);
52     Instantiate(fireball, launchPoint.position, launchPoint.rotation);
53 }
54

```

The changes i made to my ShootAtPlayerInRange C# script so that my wizard enemies could fire fireballs at the player in the build version of my game.



How my game is after all the test, changes, adjustments and everything i did to my game, which has been a improvement from previous versions.