Learning Journal

18/02/20

Today I made a VFX to use as the bullet for my scene: it took quite a while as I started not long ago using Unity shader graph, but in the end I managed to get the spiral effect on it that I was looking for.

25/02/20

Today I’ve been focusing on the player’s movement and on the camera controls. The Component project will be a third person character controller with the ability to shoot in the direction where the camera is aiming. With the help of Tony and Paul I have been able to make the player shoot at the collision point between a raycast (shot from the centre of the screen) and the ground. It was working when the player was shooting a sphere but not when it was shooting the VFX made last week. I didn’t figured out yet the reason why when the VFX is destroyed on-collision the fps massively drop but I think it’s a rendering issue.

27/02/20

Today I fixed the problems I had with the instantiation of the VFX game-object, which is fortunately not giving problems on the PCs upstairs in the game studio.

03/03/20

Today I’ve worked on the camera control. I was searching for a third person camera, which I already had, but it didn’t work properly because I linked the camera’s pointing direction to a target on the top right of my character with an offset. The problem was that if I was turning around it could point also the character itself, and that’s not what I want. I then decided to use the character as a pivot while the camera was rotating, but without pointing to the player. With Paul and Tony’s help I managed to do it, and in the end the most important part in the script was this one:

void LateUpdate()

{

objectToFollow.transform.Rotate(0, currentX\*speed, 0, Space.Self);

Vector3 dir = new Vector3(0,0, -distance);

camTransform.position = objectToFollow.transform.position;

camTransform.rotation = objectToFollow.transform.rotation;

Quaternion rotation = Quaternion.Euler(currentY, 0, 0);

camTransform.rotation \*= rotation;

camTransform.Translate(Vector3.back \* distance + offset, Space.Self);

}

10/03/20

I’m working again on the player’s movement and the way it behaves based on the camera pointing direction. The result I'm looking for is my character moving forwards wherever the camera is looking.

I asked Paul for help and just toggling a bit the already existing the void Update the result ended up being this:

*Old version:*

void Update()

{

rb.velocity = new Vector3(Input.GetAxis("Horizontal")\* speed, rb.velocity.y, Input.GetAxis("Vertical")\* speed);

if(Input.GetButtonDown("Jump"))

{

rb.velocity = new Vector3(rb.velocity.x, jumpForce, rb.velocity.z);

}

}

*New version:*

void Update()

{

rb.velocity =

Vector3.up \* rb.velocity.y +

rb.transform.right \* Input.GetAxis("Horizontal") \* speed +

rb.transform.forward \* Input.GetAxis("Vertical")\* speed;

if(Input.GetButtonDown("Jump"))

{

rb.velocity = new Vector3(rb.velocity.x, jumpForce, rb.velocity.z);

}

}

In the end I fixed a little problem with the camera: it was rotating in the opposite way on the Y Axis, so I had to go in Edit>Project Settings>Input>Axes>Mouse Y>Invert.

22/04/20

I had to change my Unity version as I was using the latest one on my pc and it was giving me problems. So doing, I had to pack all my assets as prefabs and then I had to import them in a new Unity project.  
Also I decided to not use anymore the vfx I made because it was pointless to have, so I wil just stick with my sphere prefab.

14/05/2020

Today I’ve been finishing my coursework project. I had to fix some issues I had with the prefabs and I had to set up the whole project in order to export it and in order to have to folders all in right place.  
I also had to finish my documentation which wasn’t too exact and I’ve not been precise enough in the packages’ manuals.  
Anyway I think I will need to use this project for my 3rd year’s AGP as my game will need a Third Person Camera Behaviour with a ranged combat system.