

Controlador de Personaje

Uriel Alejandro Guillén Ayala

El programa fue creado como un controlador básico de personaje para que el usuario pueda moverlo usando inputs del teclado.

Ahora explicare para que sirve cada sección del código

En esta sección se declaran todas las variables que van a ser usadas dentro de unity

```
5 public class PlayerController : MonoBehaviour
6 {
7     public float HorizontalMove; //---Esta variable almacena el valor de movimiento horizontal
8     public float VerticalMove; //---Esta variable almacena el valor de movimiento vertical
9     public float JumpMove; //---Esta variable controla el salto
10    public CharacterController Player; //---Esto define a quien se le va a efectuar el script (Character controller afecta a Player)
11    public float speed; //---Esta variable va a almacenar el valor de la velocidad a la cual se mueve el objeto
12
13
14    private Vector3 playerInput;
15    public Camera mainCamera; //---Esta es la variable que referencia a la camara
16    private Vector3 camForward;
17    private Vector3 camRight;
18    private Vector3 movePlayer; //---Esta variable almacena hacia donde se mueve el jugador
19
20    public float gravity = 9.8f;
21}
```

Esta sección dentro de Start obtiene la información recibida del componente Character Controller del objeto player en el juego.

```
21
22 // Start is called before the first frame update
23 void Start()
24 {
25     //Esto es para especificar que estamos obteniendo los datos del character controller del objeto (capsula)
26     Player = GetComponent<CharacterController>();
27 }
28
```

Estas líneas de código son las encargadas de declarar las teclas de las que se va a recibir el input del teclado.

La siguiente se encarga de mantener los parámetros de movimiento limitados a una escala de 0 a 1 impidiendo que vayan más allá

La sección de Cam Direction se encarga de hacer que el personaje mire a diferentes direcciones en posición relativa a la cámara.

```
30 void Update()
31 {
32     HorizontalMove = Input.GetAxis("Horizontal");//<---Este codigo indi
33     VerticalMove = Input.GetAxis("Vertical");
34     JumpMove = Input.GetAxis("Jump");
35
36
37     playerInput = new Vector3(HorizontalMove, JumpMove, VerticalMove);
38     playerInput = Vector3.ClampMagnitude(playerInput, 1);//<---Lo que ha
39
40     CamDirection();
41
42     movePlayer = playerInput.x * camRight + playerInput.z * camForward;
43
44     movePlayer = movePlayer * speed;//<---La variable speed fue reinsert
45
46     Player.transform.LookAt(Player.transform.position + movePlayer);//<
47
```

En la primera línea de código se declara la gravedad, se encarga de que el personaje pueda caer al suelo.

La segunda línea de código Player.Move se encarga de declarar los valores de movimiento del objeto player que van a manipular las variables horizontal move, vertical move y JumpMove.

```
SetGravity();//<--- esto va a dar el valor de
Player.Move(movePlayer * Time.deltaTime);//<---
```

Estos 2 void se encargan de almacenar todo el código referente a la gravedad y la posición relativa de la cámara que previamente en el código se manda a llamar.

```
1 referencia
void CamDirection()
{
    camForward = mainCamera.transform.forward; //---Estas líneas de código declaran las variables camforward y right para que den la vuelta a esa dirección
    camRight = mainCamera.transform.right; //---Para usar transform se debe primero llamar a la variable afectada.transform.direccion a la cual se va a transformar

    camForward.y = 0; //---Estas líneas solo es para que la cámara no se mueva en el eje y
    camRight.y = 0;

    camForward = camForward.normalized; //---Esta línea hace que los valores de movimiento sean exactos y no se vayan a decimales innecesarios
    camRight = camRight.normalized;
}

1 referencia
void SetGravity() //---Aquí se declara la gravedad
{
    movePlayer.y = -gravity; //---Esto le indica al objeto que se desplace en el eje y usando el valor de la gravedad declarado
    //Es importante poner el signo de menos en gravity para que se mueva hacia abajo y no hacia arriba
}
```

Por último este código se encarga de hacer que la cámara siga a todos lados al personaje

```
public class CameraController : MonoBehaviour
{
    // Start is called before the first frame update
    public GameObject Player;
    private Vector3 PosiciónCamara;
    ☞ Mensaje de Unity | 0 referencias
    void Start()
    {
        PosiciónCamara = transform.position - Player.transform.position;
    }

    // Update is called once per frame
    ☞ Mensaje de Unity | 0 referencias
    void LateUpdate()
    {
        transform.position = Player.transform.position + PosiciónCamara;
    }
}
```