

Sistema de Spawn

Por Fátima Sireth Vargas García

Introducción.

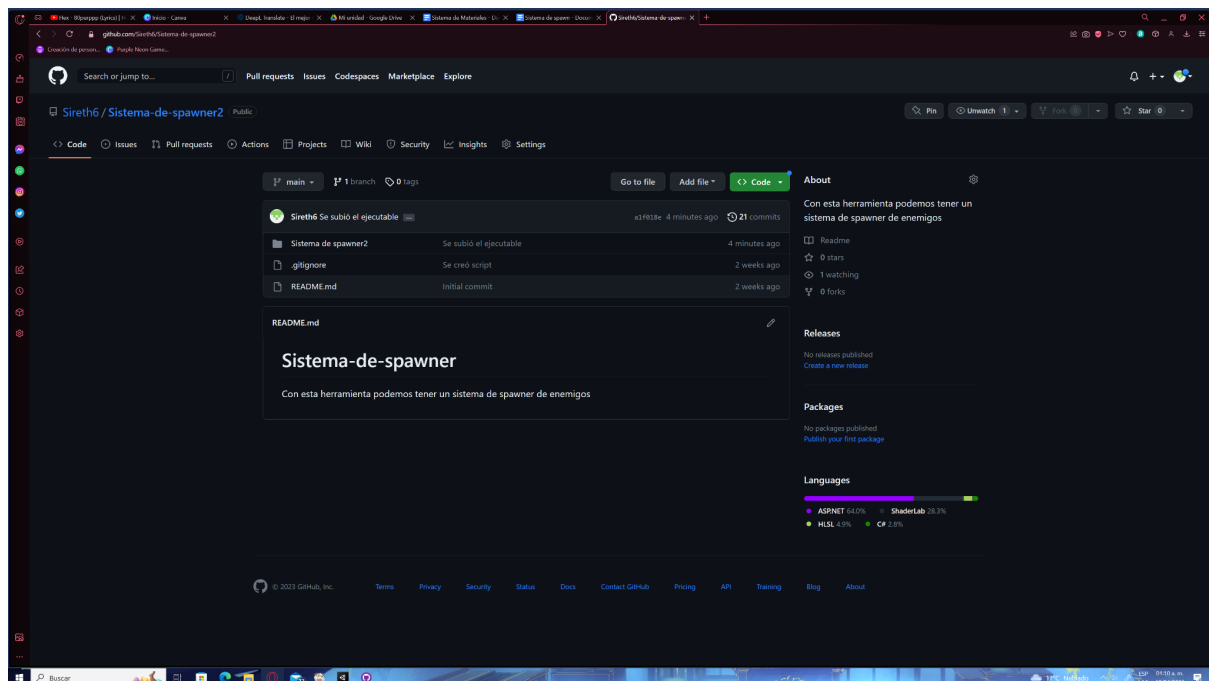
Este proyecto tiene contenido para hacer un sistema de spawn de enemigos (en este caso zombies) en primera persona, de forma más específica, es que, al inicio de la ronda saldrán dos enemigos, cuando tu player mate a esos enemigos se generarán el doble de enemigos, y así consecutivamente. Podrás encontrar script para movimiento de tu player, spawn, de estadísticas, etc.

Instalación.

Para la instalación solo hay que seguir unos sencillos pasos.

1. Para más comodidad, puedes entrar a este link <https://github.com/Sireth6/Sistema-de-spawn2> para entrar de una forma más sencilla y rápida.
2. En el caso de no ser posible entrar al link, puedes encontrar el repositorio buscando GitHub en Google, después, en el buscador de GitHub, introducirás el nombre “Sireth6/Sistema-de-spawn2”
3. Una vez hayas entrado a la página, dale clic al botón verde con la leyenda “code” y le darás clic en “Download Zip”

4. Se descargará un archivo .zip o .rar.
5. Darás clic derecho al archivo y después en “Extraer aquí”.
6. Te irás a la carpeta “Sistema de diálogos” y te aparecerán 4 carpetas más.
7. Finalmente arrastra esas 4 carpetas a tu proyecto de Unity.

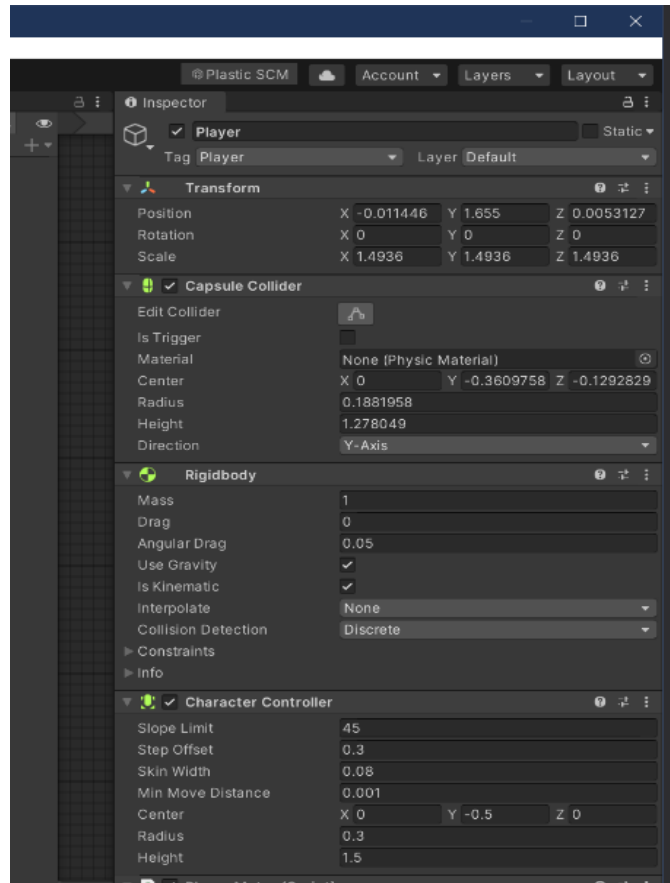


Contenido.

1. Scripts para tu player.

Antes de empezar con los scripts para que todo funcione forma adecuada, tendrás que añadirle a tu player los siguientes componentes:

- Capsule Collider
- Rigidbody
- Character controller



Player Motor: Este script sirve para el movimiento de tu personaje de izquierda a derecha y de adelante hacia atrás, también cuenta con una opción para que al apretar la tecla “shift” tu player corra más rápido y que con la barra de espacio pueda saltar. Aquí puedes modificar a tu gusto la gravedad, la velocidad al caminar, al sprintar y al saltar.

Camara Controller: Este script sirve para que al momento de mover el mouse, nuestra vista se dirija hacia este y que caminemos hacia la dirección que está viendo, para esto, la cámara tiene que ser “hijo” del player.

```

14
15 void Update()
16 {
17     Cursor.lockState = CursorLockMode.Locked;
18     RotateCamera();
19 }
20
21 void RotateCamera()
22 {
23     float mouseX = Input.GetAxis("Mouse X");
24     float mouseY = Input.GetAxis("Mouse Y");
25
26     float rotAmountX = mouseX * mouseSensitivity;
27     float rotAmountY = mouseY * mouseSensitivity;
28
29     xAxisClamp -= rotAmountY;
30
31     Vector3 rotPlayerArms = playerArms.transform.rotation.eulerAngles;
32     Vector3 rotPlayer = player.transform.rotation.eulerAngles;
33
34     rotPlayerArms.x -= rotAmountY;
35     rotPlayerArms.z = 0;
36     rotPlayer.y += rotAmountX;
37
38     if (xAxisClamp > 90)
39     {
40         xAxisClamp = 90;
41         rotPlayerArms.x = 90;
42     }
43     else if (xAxisClamp < -90)
44     {
45         xAxisClamp = -90;
46         rotPlayerArms.x = 270;
47     }
48
49     playerArms.rotation = Quaternion.Euler(rotPlayerArms);
50     player.rotation = Quaternion.Euler(rotPlayer);
51 }
52
53
54

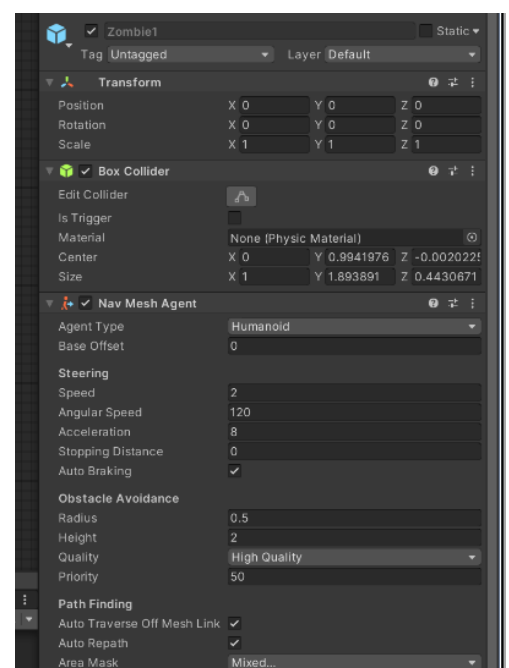
```

Character Stats y Player Stats: Estos dos scripts se utilizan para la vida de tu personaje y del enemigo, en este caso no se utilizaron pero sirven de ejemplo.

2. Scripts para tus enemigos.

Al igual que el paso anterior, antes de ver los scripts vamos a ver algunas recomendaciones para que todo funcione de forma adecuada, a nuestro objeto “enemigo” lo haremos un prefab, a ese prefab le añadiremos los siguientes componentes.

- Box Collider
- Nav Mesh Agent (éste se usa para que nuestros enemigos sigan a nuestro player, se puede encontrar en la ventana “component” y en “navigation”)



Script Enemy

El contenido de este script sirve para que nuestros enemigos sigan al objeto con el tag “player” con ayuda del Nav Mesh Agent, también sirve para que los enemigos se detengan a cierta distancia de nuestro player y no nos arrastren, tu puedes modificar la distancia en la que quieras que se detengan.

```
6
7 [SerializeField] float stoppingDistance;
8
9 NavMeshAgent agent;
10
11 GameObject target;
12
13 private void Start ()
14 {
15     agent = GetComponent<NavMeshAgent>();
16     target = GameObject.FindGameObjectWithTag("Player");
17 }
18
19 private void Update ()
20 {
21     float dist = Vector3.Distance(transform.position, target.transform.position);
22     if(dist < stoppingDistance)
23     {
24         StopEnemy();
25     }
26     else
27     {
28         GoToTarget();
29     }
30 }
31
32 private void GoToTarget()
33 {
34     agent.isStopped = false;
35     agent.SetDestination(target.transform.position);
36 }
37
38 private void StopEnemy()
39 {
40     agent.isStopped = true;
41 }
42
43
44
```

Game Controller

Sirve para llevar un score de los enemigos asesinados.

Enemy Stats

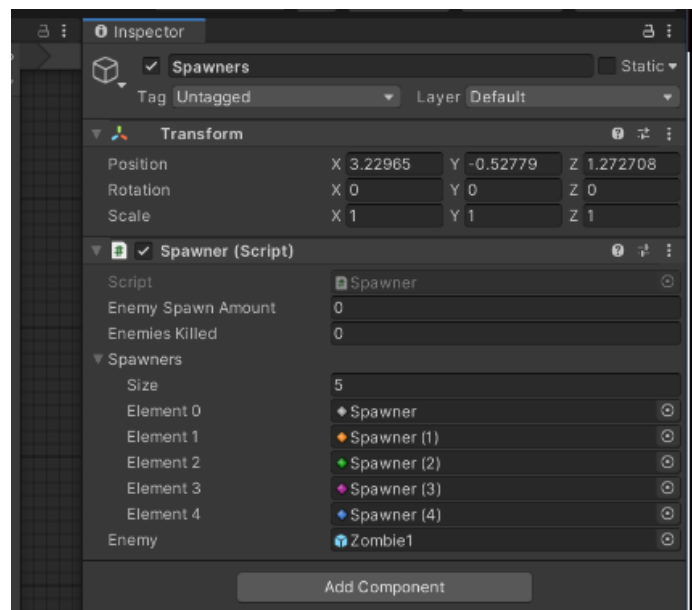
Lleva las estadísticas de los enemigos, en este caso se agregó una función para destruir a los enemigos con una sola tecla en este caso es la “L” pero puedes usar cualquier otra.

3. Script Spawner

Ahora vamos a lo principal, antes de usar el script de spawner, tenemos que crear un objeto vacío el cual llamaremos “spawners” dentro de ese objeto crearemos otros 5 objetos vacíos que llevarán el nombre de spawner1,

spawner2, etc. De estos objetos vacíos es de donde saldrán nuestros enemigos de forma aleatoria. A nuestro objeto vacío “Spawners” le agregaremos este script, después en el apartado “size”

pondremos el número de spawners que hayamos creado y en los elementos arrastraremos cada uno de ellos y en el apartado de “enemy” arrastraremos nuestro prefab del enemigo.



Este script funciona para que los enemigos spawnen en nuestros objetos vacíos de forma aleatoria, al darle play, empezará con dos enemigos, cuando presionemos la tecla “L” los enemigos se destruirán pero ahora se spawnearán 4, este script está hecho para que una vez mates a todos los enemigos se sumarán otros dos.

Recomendaciones

Recuerden agregar todos los componentes necesarios para que la herramienta funcione correctamente, también ten en cuenta que puedes modificar cualquier valor de los scripts para que no tengas limitaciones.