

Tutorial 4: Death by falling, losing a life and respawning

This tutorial is going to be 100% made by me. It's not just one big code but 3 different ones that complement each other. The idea is to part from having already a moving character, it doesn't matter if it is in first or third person, it should work the same.

The player will be able to fall while platforming and die. While the player has enough lives left, they should keep respawning at the cost of losing one life. If the player runs out of lives, it's game over.

Just in case you haven't read my other 3 tutorials, here I leave a screenshot of the script for the Player Controller:

```
//Ground collision check variables
public Transform groundCheck;
public float groundDistance = 0.4f;
public LayerMask groundMask;
private bool isGrounded;

//Movement variables
public CharacterController controller;
public float speed = 12f;

//Jump variables
public float jumpHeight = 3f;

//Gravity variables
public float gravity = -9.81f;
private Vector3 velocity;
```

```
// Update is called once per frame
void Update()
{
    //Ground collision check
    isGrounded = Physics.CheckSphere(groundCheck.position, groundDistance, groundMask);
    if (isGrounded && velocity.y < 0)
    {
        velocity.y = -2f;
    }

    //Movement
    float x = Input.GetAxis("Horizontal");
    float z = Input.GetAxis("Vertical");
    Vector3 movement = transform.right * x + transform.forward * z;
    controller.Move(movement * speed * Time.deltaTime);

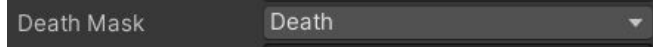
    //Jump
    if (Input.GetButtonDown("Jump") && isGrounded)
    {
        velocity.y += Mathf.Sqrt(jumpHeight * -2f * gravity);
    }

    //Gravity
    velocity.y += gravity * Time.deltaTime;
    controller.Move(velocity * Time.deltaTime);
}
```

First, you need to situate an object at the end of the fall, and change it's layer to a different one from everything else. In my case I have added a plane at the end of the fall and created a new layer named "Death".

I'm going to use the same technique as when I wanted to check if the player was grounded, so let's create a few new variables.

```
public LayerMask deathMask;  
//change the layer of the variable in the inspector.
```



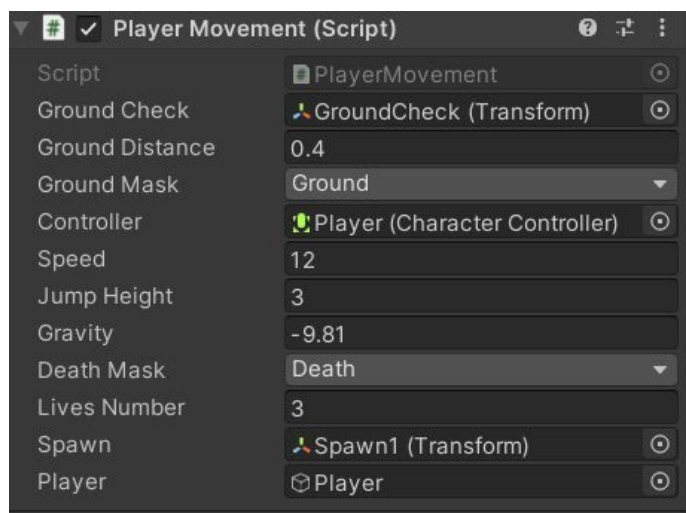
```
private bool isDead;  
//to check if the player has indeed touched the "death plane".
```

```
public float livesNumber = 3f;  
//we add a number of lives so that the player can only respawn a limited  
number of times.
```

```
public Transform spawn;  
//You'll need to create a GameObject to act as a spawn point.
```

```
public GameObject player;  
//To be able to teleport the player to the spawn point.
```

This is how the script in the inspector should look like:



After that, we'll start to write inside the Update void. Imitating the "isGrounded checker", we will do an "isDead checker".

```
isDead = Physics.CheckSphere(groundCheck.position, groundDistance, deathMask);  
//The only thing we need to change is the boolean and the layermask
```

Then, if the boolean turns out “true”, we’ll need to check if the player has lives left or not. If they have some lives left, we will activate the respawn, if not, it will be game over for the player. To activate the respawn, we’ll need to enable the character controller for a brief amount of time, since apparently it hardly works with manual changes of position. Then, we only have to teleport the player to the spawn point and enable the character controller again.

```
controller.enabled = false;
player.transform.position = spawn.position;
controller.enabled = true;
livesNumber -= 1;
//Since we have touched the “death plane”, the player should lose a life.
```

If the player has no more lives, it would be game over. In that case it’s a free choice for you to choose how to represent a game over. Closing the game, changing the scene to a game over menu, a reset of the game or even just a message to the console. That last one is the one I’m going to use today. So, inside the “else” after the “if”, you can just Debug.Log a message:

```
else
{
    Debug.Log("Game Over");
}
```

The new pieces of code would look like this:

```
//Death and spawning variables
public LayerMask deathMask;
public float livesNumber = 3f;
private bool isDead;
public Transform spawn;
public GameObject player;
```

```
//Death collision check
isDead = Physics.CheckSphere(groundCheck.position, groundDistance, deathMask);
if (isDead)
{
    if (livesNumber > 0)
    {
        controller.enabled = false;
        player.transform.position = spawn.position;
        controller.enabled = true;
        livesNumber -= 1;
    }
    else
    {
        Debug.Log("Game Over");
    }
}
```