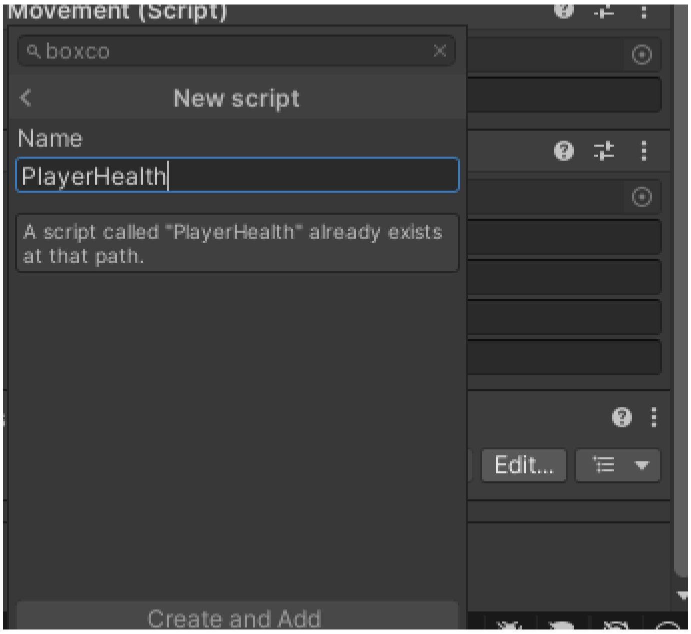
Unity Tutorial 3: Player health and damage system

This tutorial will show you how to create a simple health and damage system that makes objects damage players when they touch/collide with it.

1. Create a new C# script



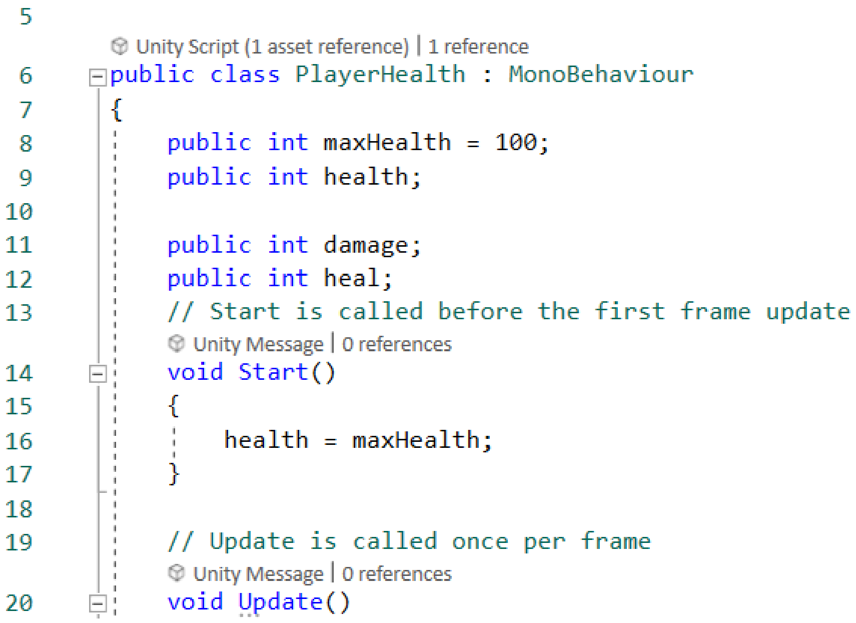
Name

PlayerHealth

A script called "PlayerHealth" already exists at that path.

Create a new script and name it “Player Health”. This is the script which you will use to assign a value to the player’s health and it will also display your player’s current health.

2. Declaring variables



```
public class PlayerHealth : MonoBehaviour
{
    public int maxHealth = 100;
    public int health;

    public int damage;
    public int heal;
    // Start is called before the first frame update
    void Start()
    {
        health = maxHealth;
    }

    // Update is called once per frame
    void Update()
```

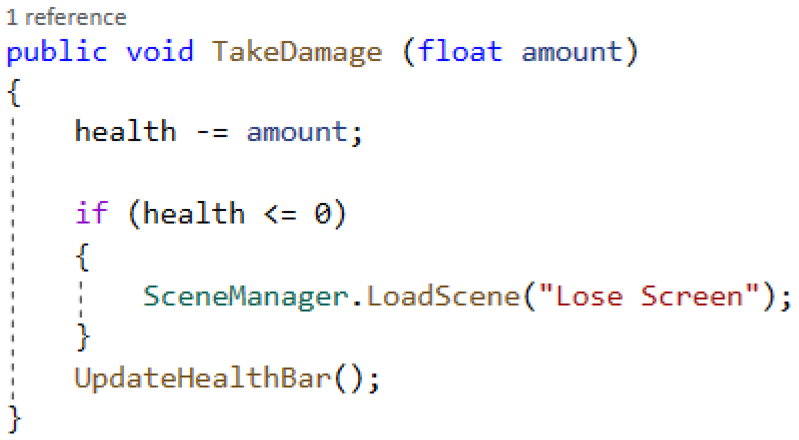
Unity Script (1 asset reference) | 1 reference
Unity Message | 0 references
Unity Message | 0 references

Now that you have created your C# script the next step is to declare the following variables:

- maxHealth: The maximum amount of health the player can obtain
- health: The player’s current health
- damage: Self-explanatory. The amount of damage the player can take when assigned to an object
- heal: The amount of healing the player can receive (optional if you feel evil and disabled healing for your players)

When you are assigning the max health to the player you must enter a value that will be your player’s max health. In the screenshot above I decided to set the value to 100.

When you start your game you must set your current health to the max health. If you are going to have items that increase your health cap then replace the max health variable to “defaultHealth” and have variables that assigns the player’s health cap to the amount you want.

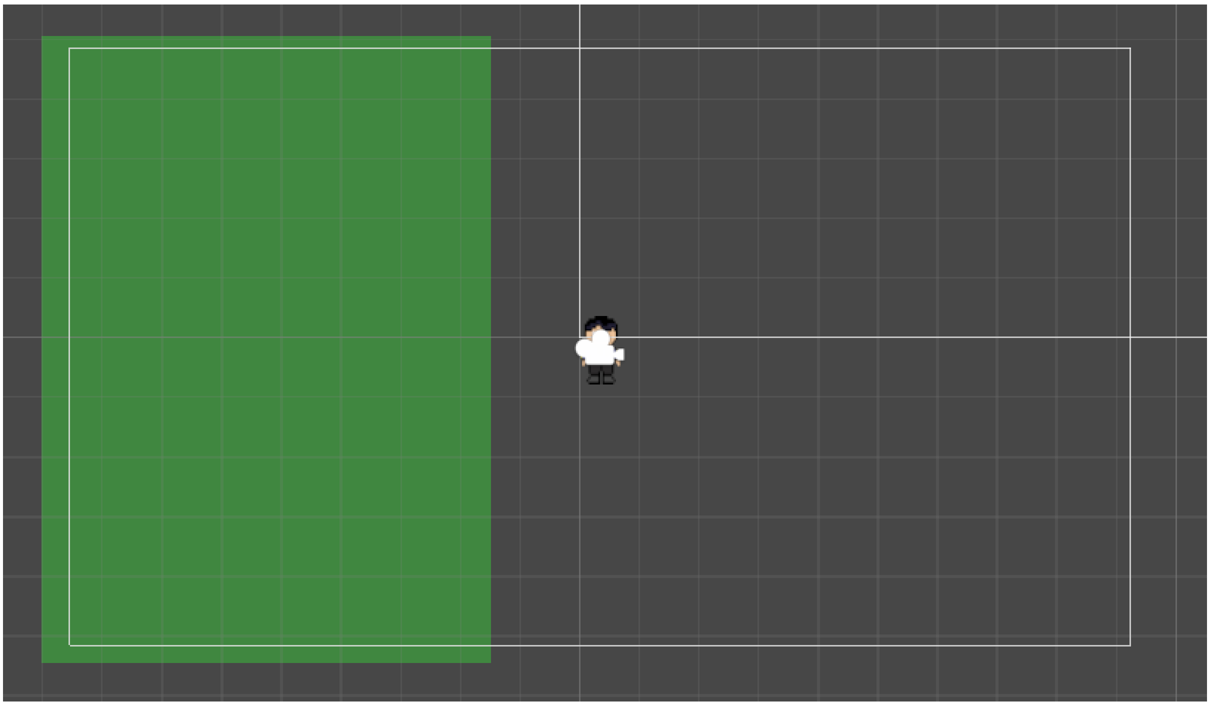


```
1 reference
public void TakeDamage (float amount)
{
    health -= amount;

    if (health <= 0)
    {
        SceneManager.LoadScene("Lose Screen");
    }
    UpdateHealthBar();
}
```

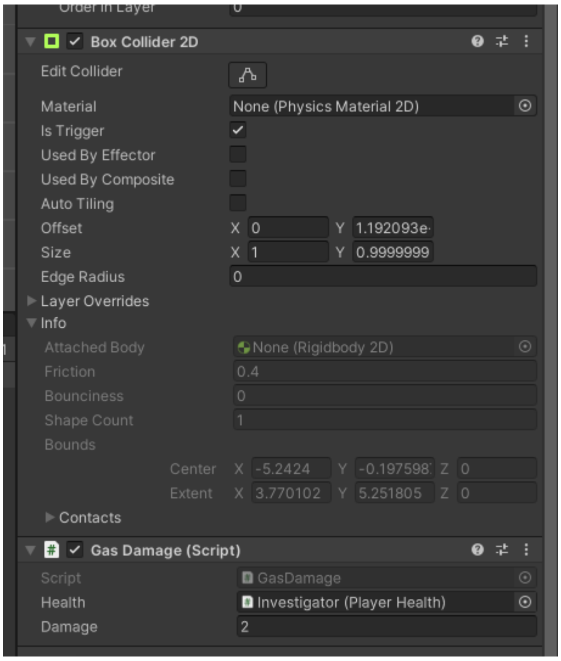
Add a take damage method which you will eventually use to damage the player when you are referencing this script. This will also cover the lose condition where you will be sent to the lose screen the moment your health reaches 0.

3. Add an object/entity that can damage the player

Now that your player has a health value assigned, you now require an object/entity to damage your player. In this screenshot below, I will be using a simple box as an example. Remember to give your object a name! Even naming him Gary works! (just kidding, don't do that)



I will name this green box "PoisonGas" for this tutorial as the example.



Order in Layer 0

Box Collider 2D ☑

Edit Collider

Material None (Physics Material 2D)

Is Trigger ☑

Used By Effector □

Used By Composite □

Auto Tiling □

Offset X 0 Y 1.192093e-

Size X 1 Y 0.9999999

Edge Radius 0

Layer Overrides

Info

Attached Body None (Rigidbody 2D)

Friction 0.4

Bounciness 0

Shape Count 1

Bounds

Center X -5.2424 Y -0.197598 Z 0

Extent X 3.770102 Y 5.251805 Z 0

Contacts

Gas Damage (Script) ☑

Script GasDamage

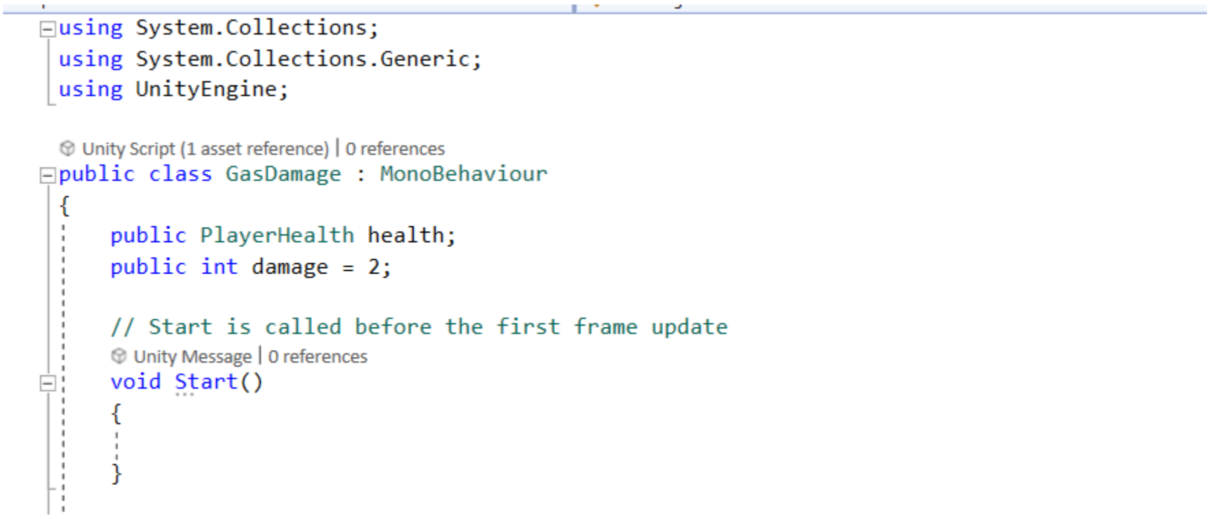
Health Investigator (Player Health)

Damage 2

Using the image above as an example, add a box collider component so the object can detect the player's collision with the object. After that, set the "Is Trigger" to true so that the collision can trigger a command.

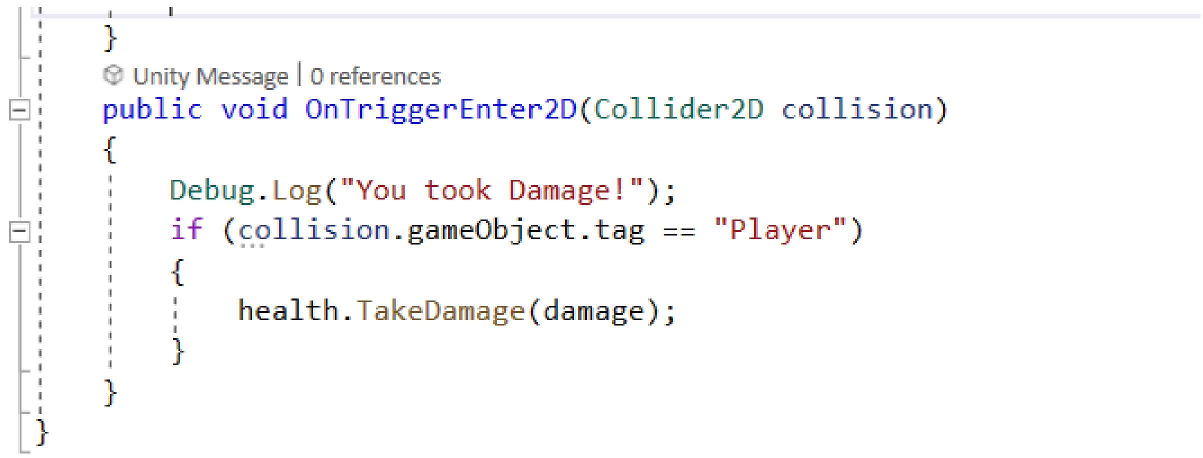
Now that you got everything you need, you just need to add a damage script for the object.

4. Programming the damage script for your object/enemy



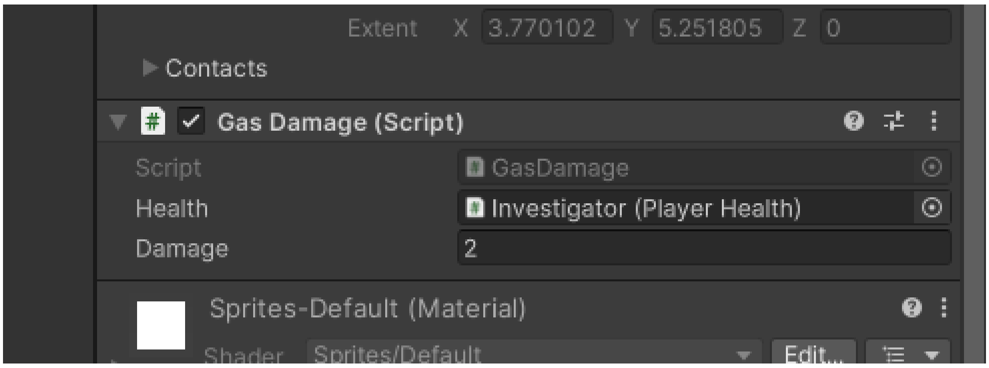
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

Unity Script (1 asset reference) | 0 references
public class GasDamage : MonoBehaviour
{
    public PlayerHealth health;
    public int damage = 2;

    // Start is called before the first frame update
    Unity Message | 0 references
    void Start()
    {

    }
```

Add a public variable referencing your player's health script, and a damage variable. Remember to assign the damage variable a value which will be the amount of damage your player will take.



```
    }
    Unity Message | 0 references
    public void OnTriggerEnter2D(Collider2D collision)
    {
        Debug.Log("You took Damage!");
        if (collision.gameObject.tag == "Player")
        {
            health.TakeDamage(damage);
        }
    }
}
```

Now add a conditional using "OnTriggerEnter2D" which triggers when the player collides with the object. The conditional will make it so the player's health will be deducted.



Extent X 3.770102 Y 5.251805 Z 0

▶ Contacts

☑ Gas Damage (Script)

Script	GasDamage
Health	Investigator (Player Health)
Damage	2

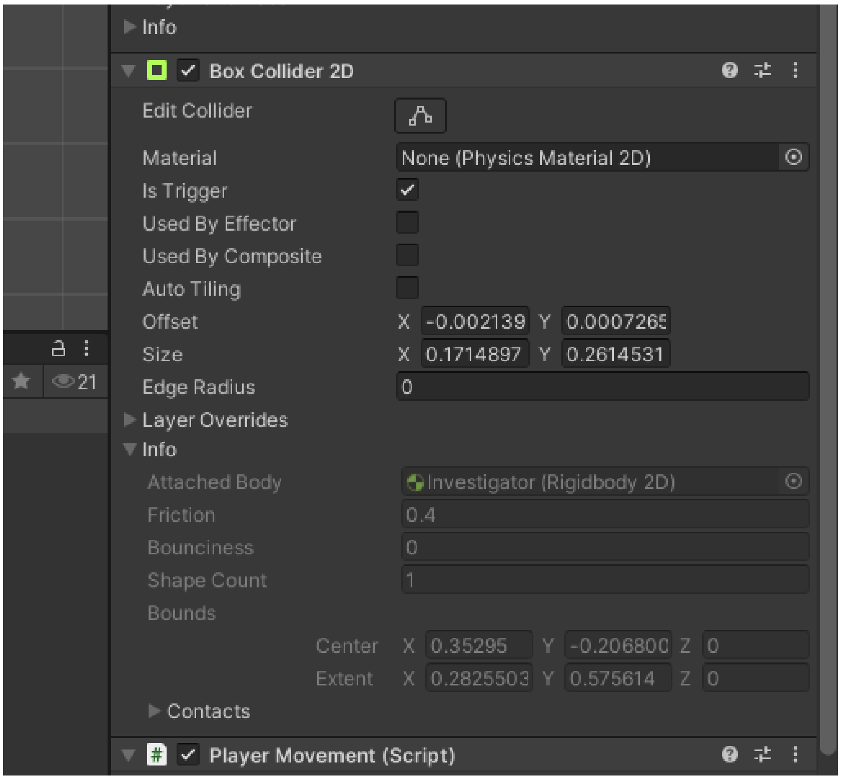
Sprites-Default (Material)

Shader Sprites/Default Edit...

Remember to connect the object damage script to the player health.

5. Player collision and hitbox

Now that your collision script is ready add a box collider to your player. This will be your player's hitbox when they are moving around. After adding a box collider, set the "IsTrigger" to true in order to enable the player to interact with objects and items when colliding with an object.



Info

☑ Box Collider 2D

Edit Collider

Material: None (Physics Material 2D)

Is Trigger: ☑

Used By Effector: □

Used By Composite: □

Auto Tiling: □

Offset: X -0.002139 Y 0.0007265

Size: X 0.1714897 Y 0.2614531

Edge Radius: 0

Layer Overrides

Info

Attached Body: Investigator (Rigidbody 2D)

Friction: 0.4

Bounciness: 0

Shape Count: 1

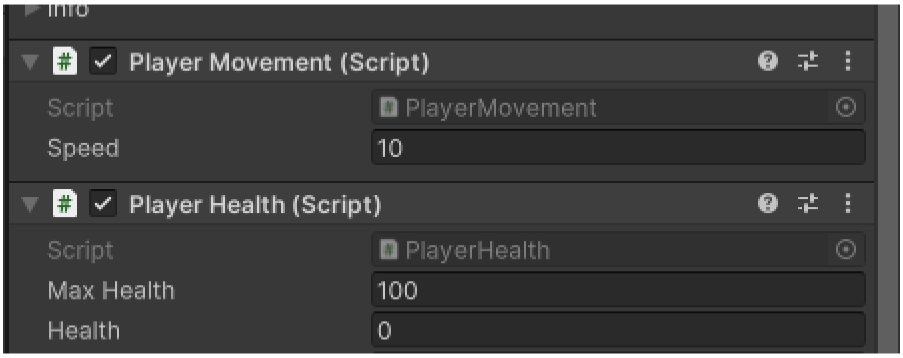
Bounds

Center: X 0.35295 Y -0.206800 Z 0

Extent: X 0.2825503 Y 0.575614 Z 0

Contacts

☑ Player Movement (Script)



Info

☑ Player Movement (Script)

Script: PlayerMovement

Speed: 10

☑ Player Health (Script)

Script: PlayerHealth

Max Health: 100

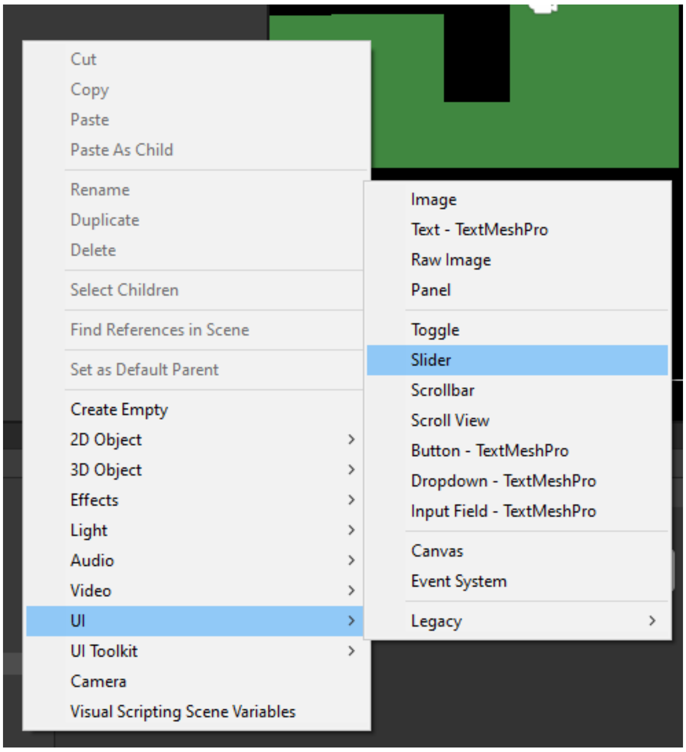
Health: 0

When you start your game, the player's health will be set to the same as the max health! The moment they take damage, their health should go down by 2.

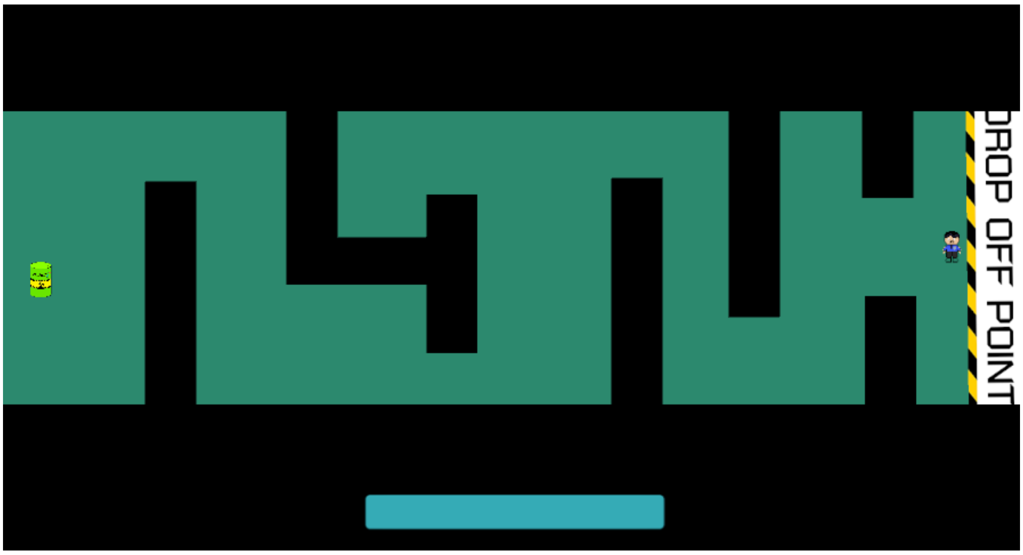
6. Adding a health bar UI

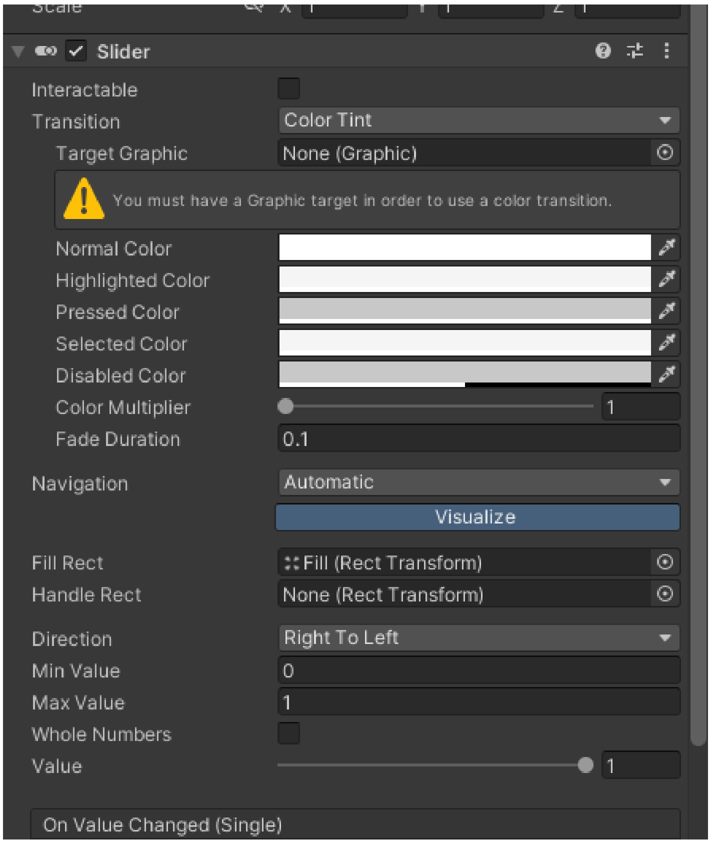
Health bars are one of the most important parts of a health system because it is used to notify the player how much health they have.

First things first add a slider by right clicking hierarchy then go to UI - > Slider



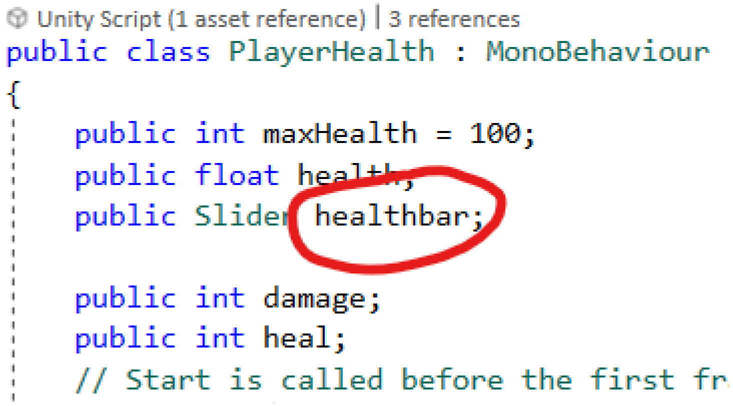
Now that you got a slider UI, you can move it to an appropriate spot. This image can be used as an example of a suitable spot. You can colour it with anything you want.





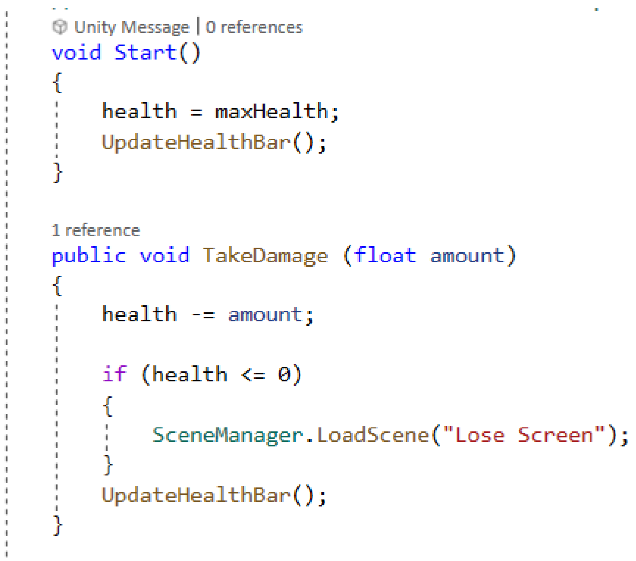
```
Scale                X 1        Y 1        Z 1
Slider
Interactable         ☐
Transition           Color Tint
  Target Graphic     None (Graphic)
  ⚠ You must have a Graphic target in order to use a color transition.
  Normal Color
  Highlighted Color
  Pressed Color
  Selected Color
  Disabled Color
  Color Multiplier   1
  Fade Duration      0.1
Navigation           Automatic
                     Visualize
Fill Rect            Fill (Rect Transform)
Handle Rect          None (Rect Transform)
Direction            Right To Left
Min Value            0
Max Value            1
Whole Numbers        ☐
Value                1
On Value Changed (Single)
```

On the inspector you can modify the min and max value. It is recommended to set the range to 0 (min) –> 1 (max).



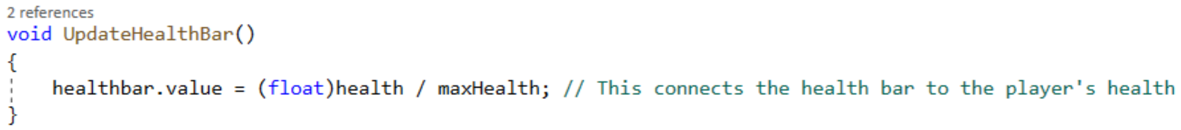
```
Unity Script (1 asset reference) | 3 references
public class PlayerHealth : MonoBehaviour
{
    public int maxHealth = 100;
    public float health;
    public Slider healthbar;

    public int damage;
    public int heal;
    // Start is called before the first fr
```

Now on the player health script add a public slider and name it “healthbar”. This will be used to reference the health bar you have for your game.



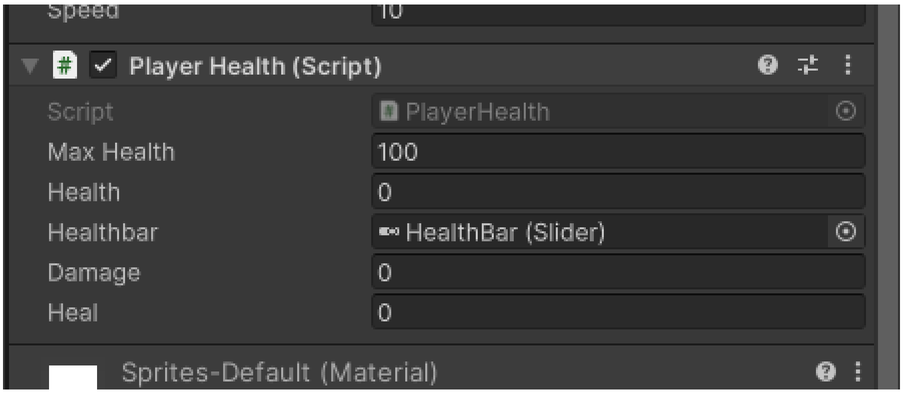
```
Unity Message | 0 references
void Start()
{
    health = maxHealth;
    UpdateHealthBar();
}

1 reference
public void TakeDamage (float amount)
{
    health -= amount;

    if (health <= 0)
    {
        SceneManager.LoadScene("Lose Screen");
    }
    UpdateHealthBar();
}
```

Add an "UpdateHealthBar()" method which will constantly update your health bar every time you take damage.



```
2 references
void UpdateHealthBar()
{
    healthbar.value = (float)health / maxHealth; // This connects the health bar to the player's health
}
```

This method connects your health bar to the player's health.



Speed 10

Player Health (Script)

Script	PlayerHealth
Max Health	100
Health	0
Healthbar	HealthBar (Slider)
Damage	0
Heal	0

Sprites-Default (Material)

After that drag your health bar slider to the health bar in the player's inspector.

If you have followed the steps properly your health bar should be able to go down every time you take damage in game!