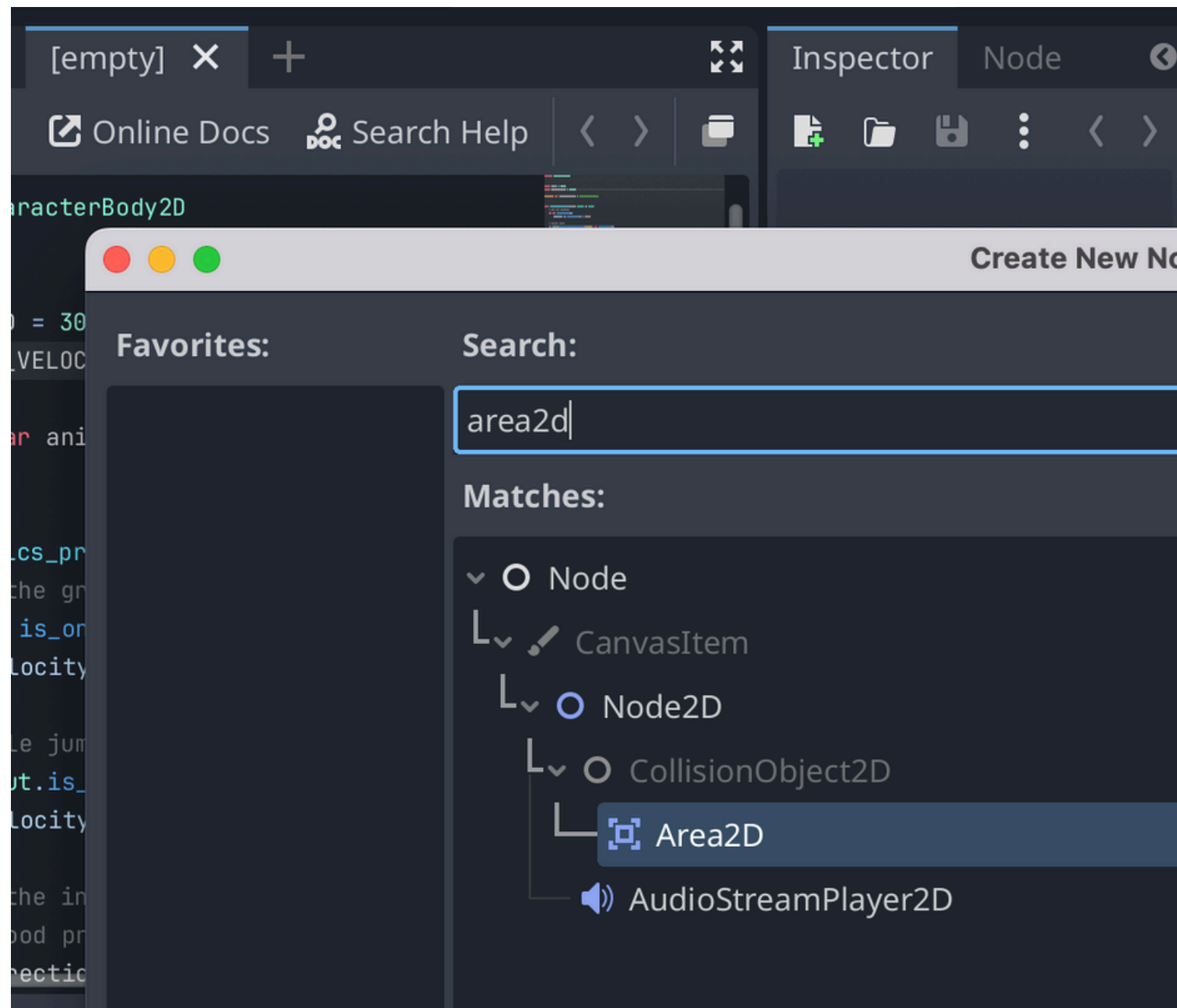
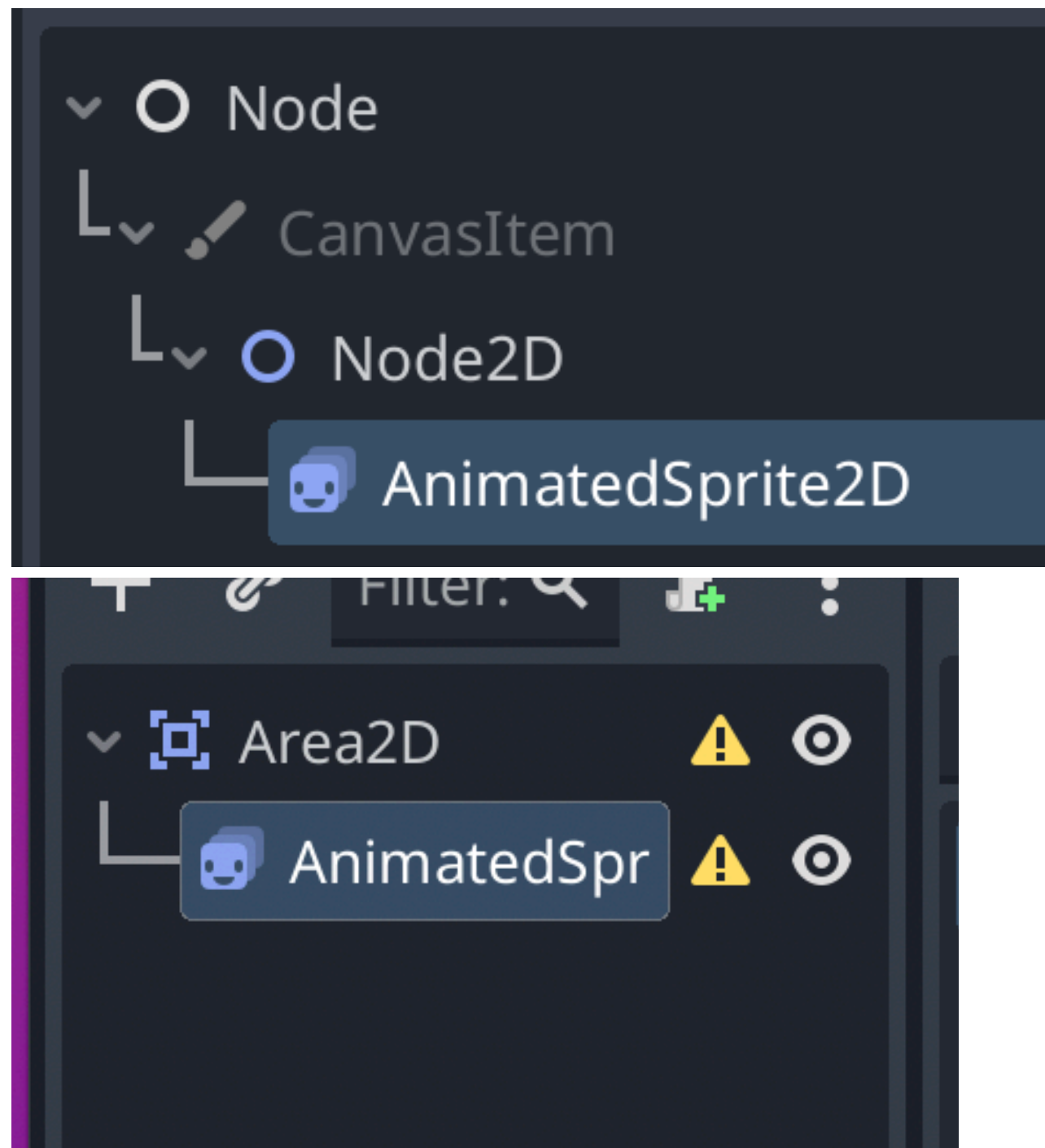


Pickups



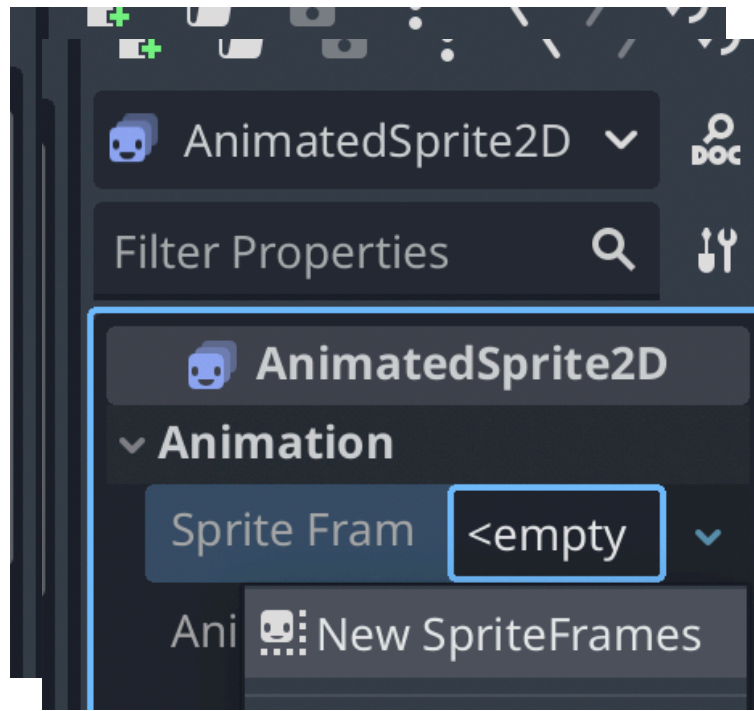
Create a NEW Scene
Add a new Node called
Area2D

This will allow us to have a
collidable area, but not a
collision reaction

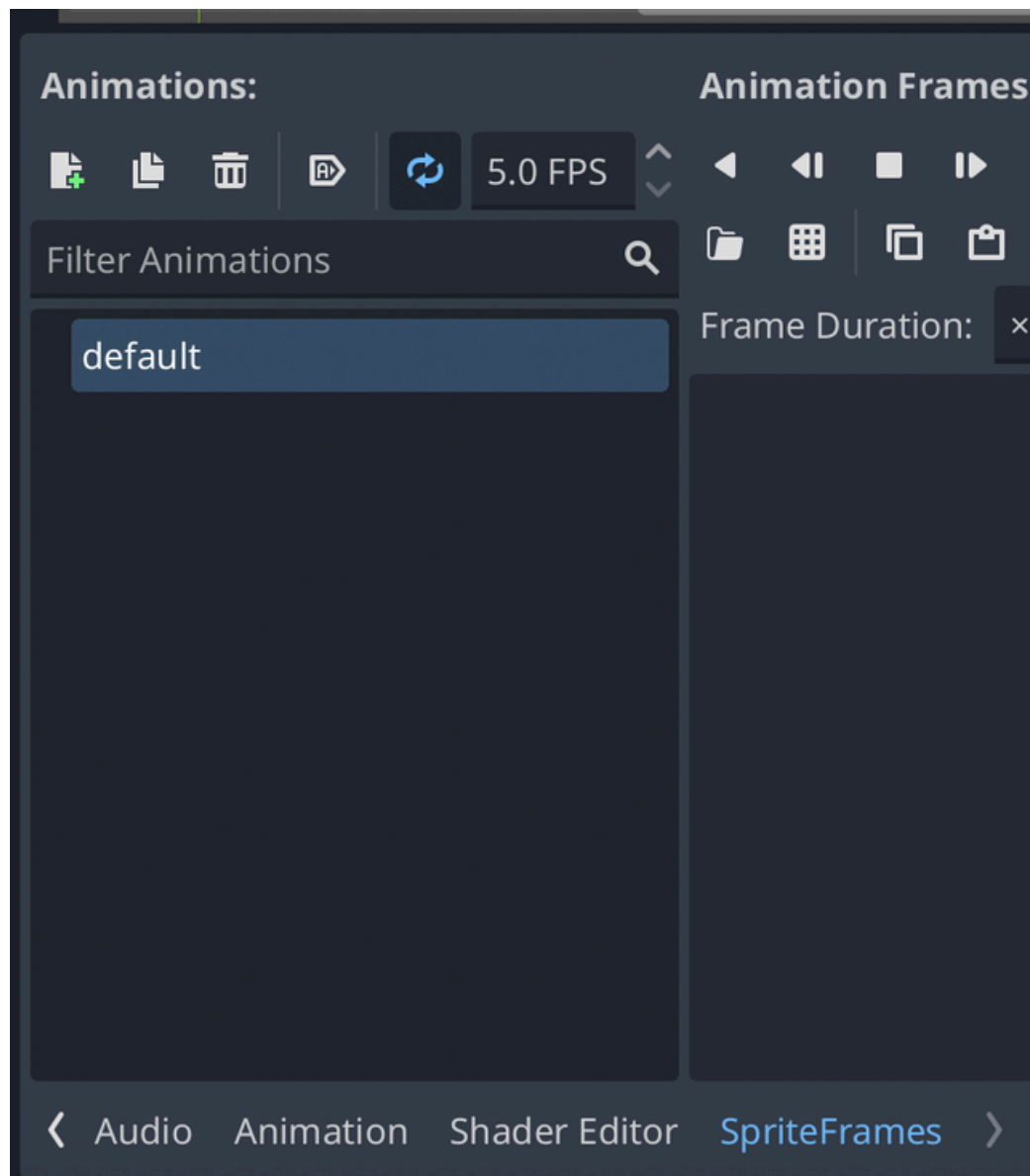


**Add a child node to the
Area 2D.**

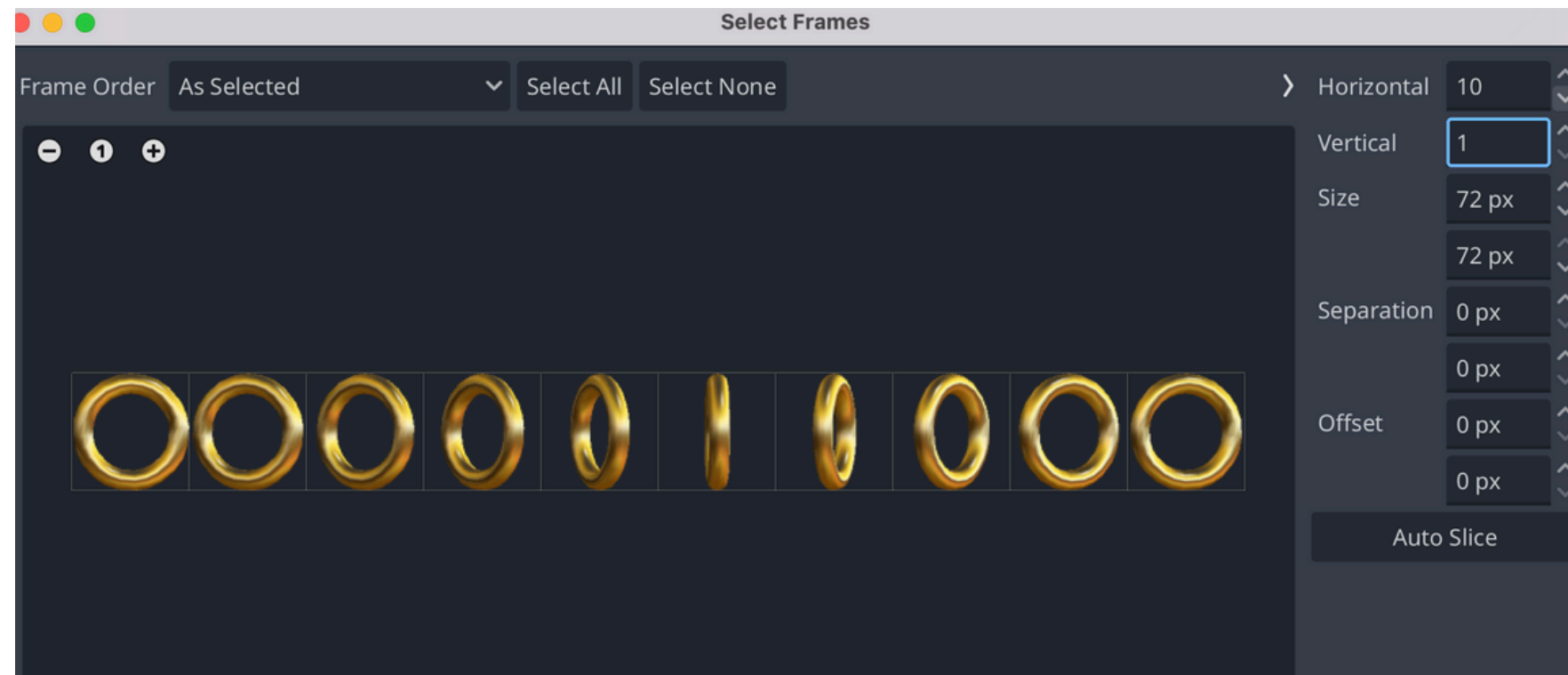
**The child node should be a
AnimatedSprite2D**



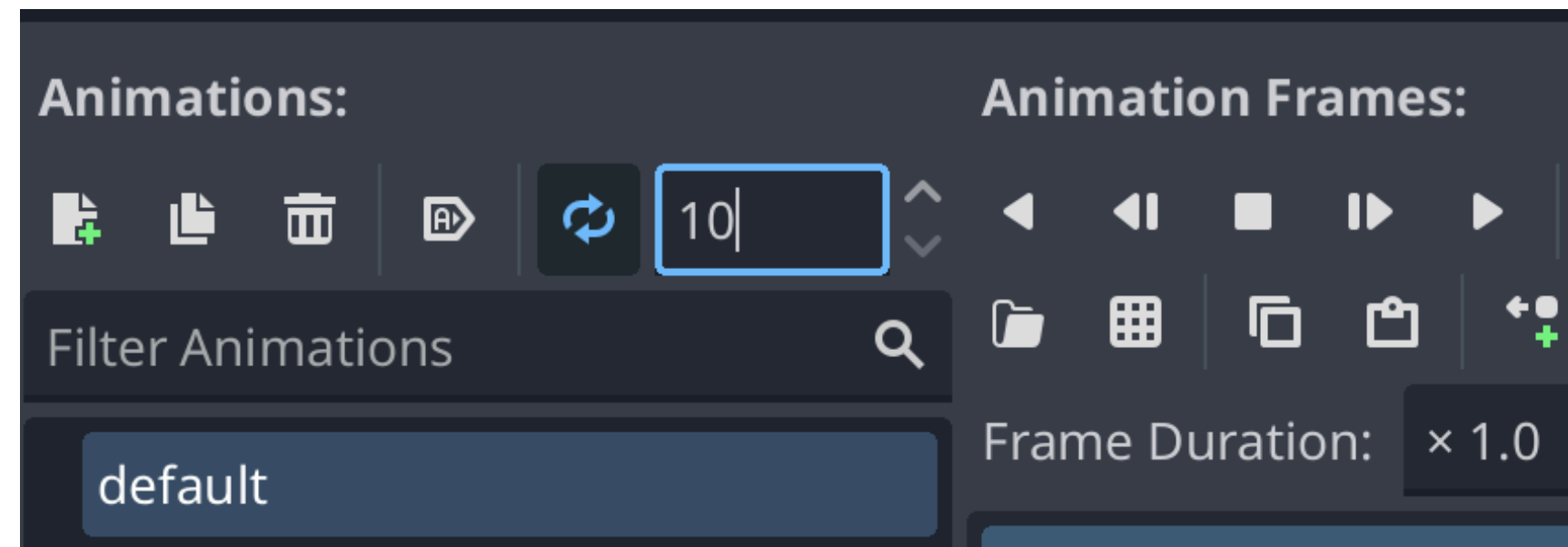
On the right side of the screen in the INSPECTOR go to Animation > Spite Frames and add NEW SPRITE FRAMES



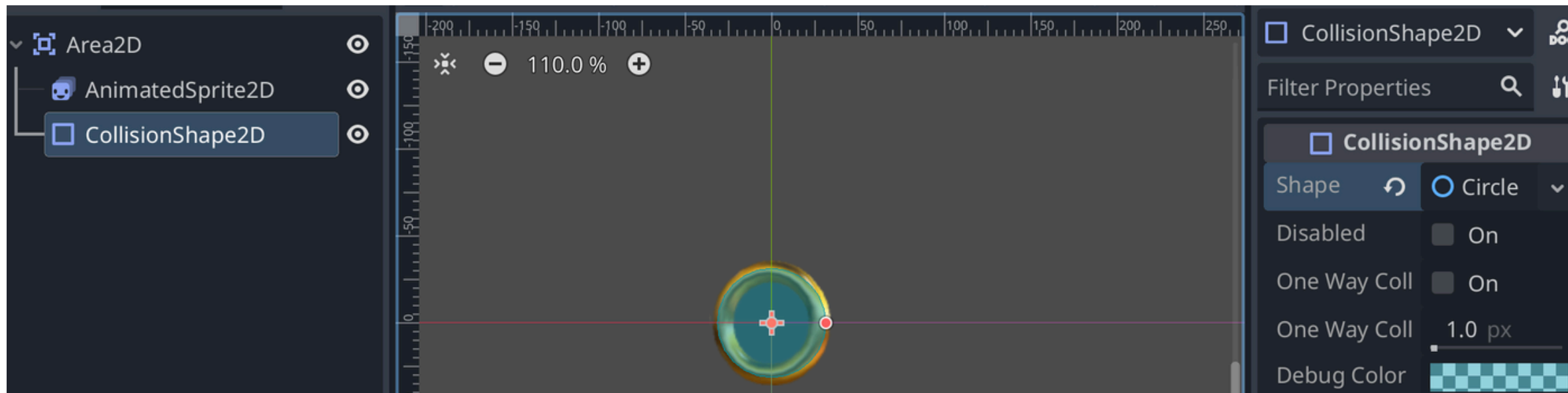
On the bottom of the screen in the Spriteframes window click on the Waffle Icon to add the Sprites



**Divide the sprite sheet correctly.
Choose Select All and Add Frames**



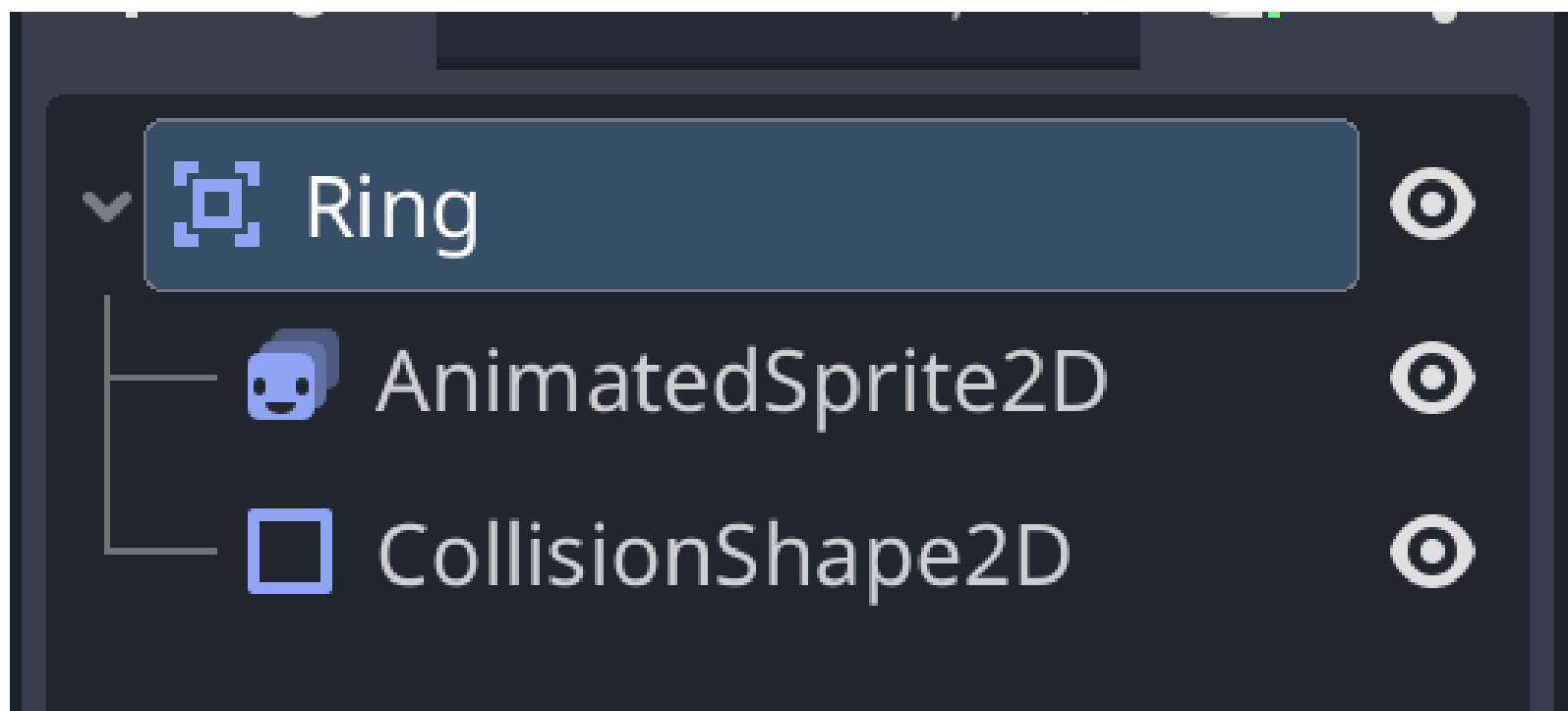
**Change the frame rate to 10 so that the
coin spins faster**



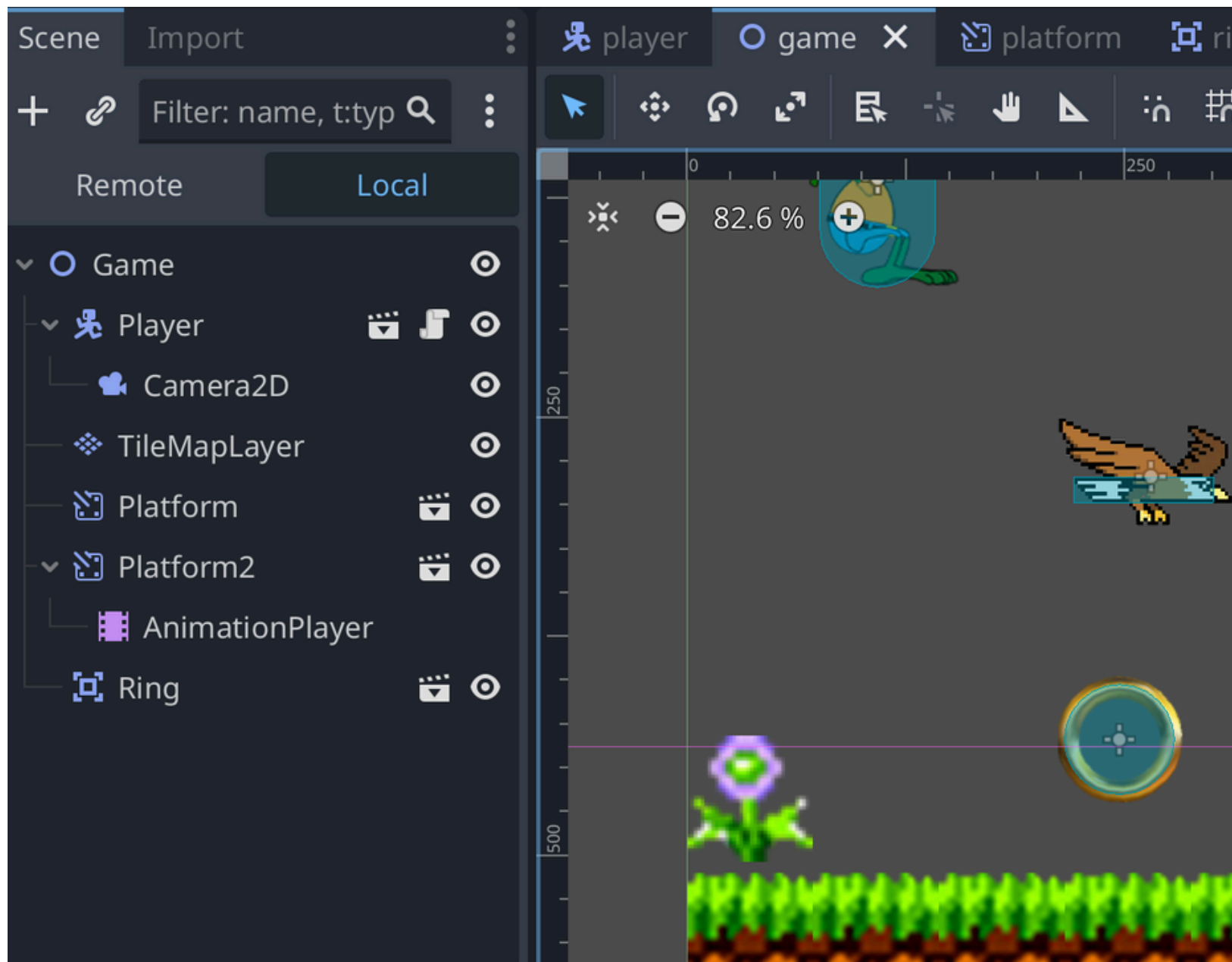
Add a CollisionShape2d

Add a Circle Shape to match the ring

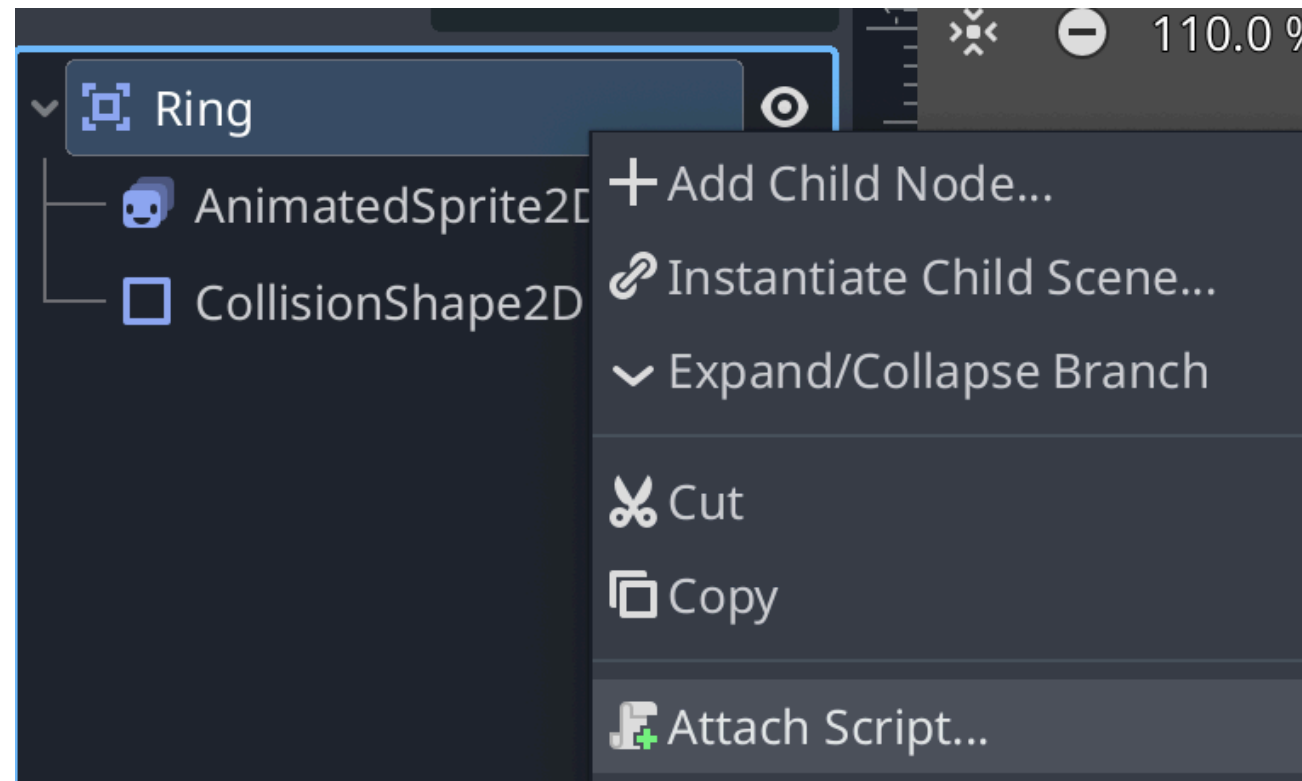
Rename the entire scene



**Rename it to Ring
(or whatever name you need)
and save the scene**

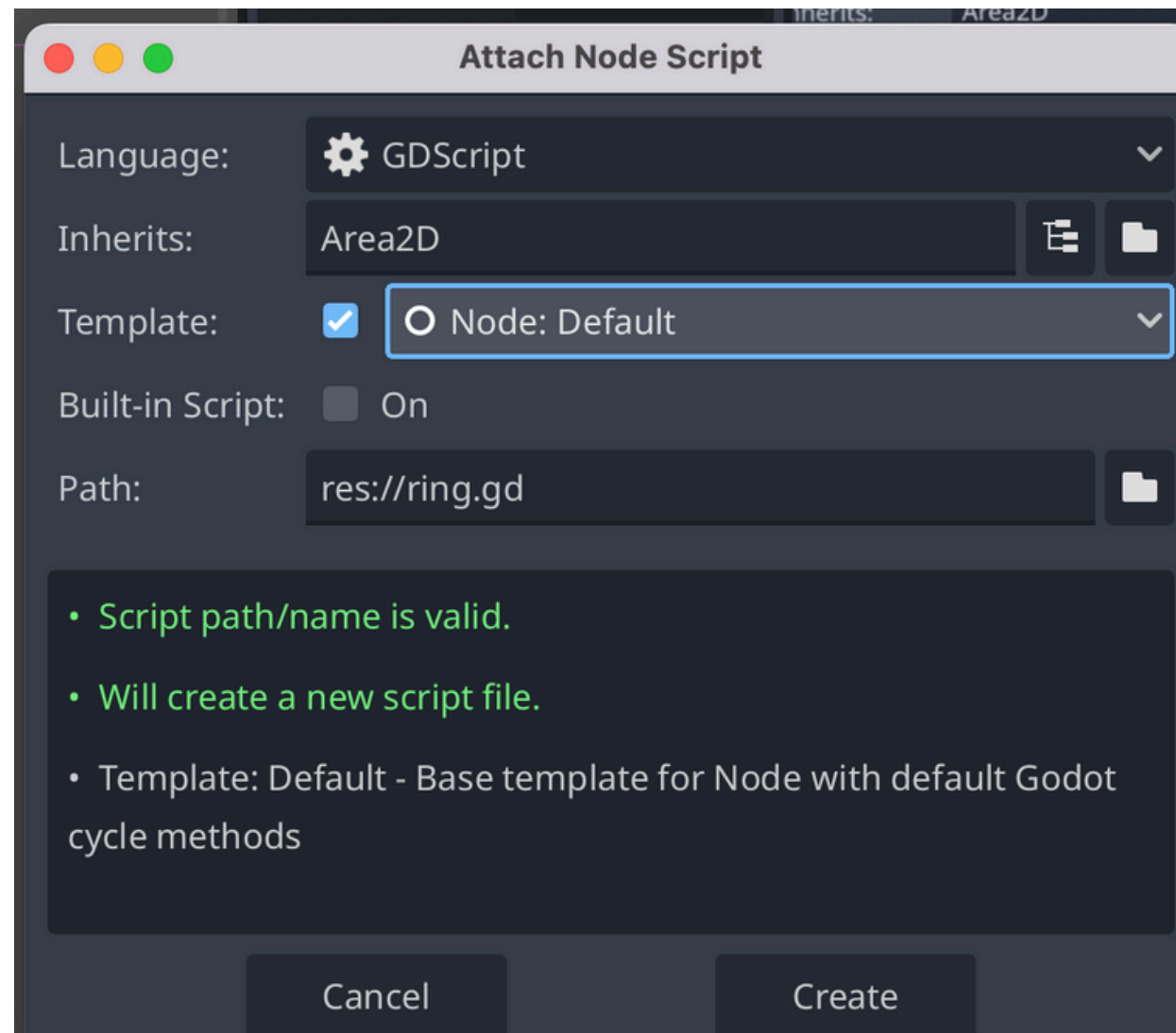


**Drag this new SCENE into
your game.
Test it.
It should animate.
You will notice it doesn't
collide, which is ok! We need
to add script.**



Go back to the Ring Scene
Add SCRIPT to the top node

Choose the Template ON
And choose the NODE:
DEFAULT
option

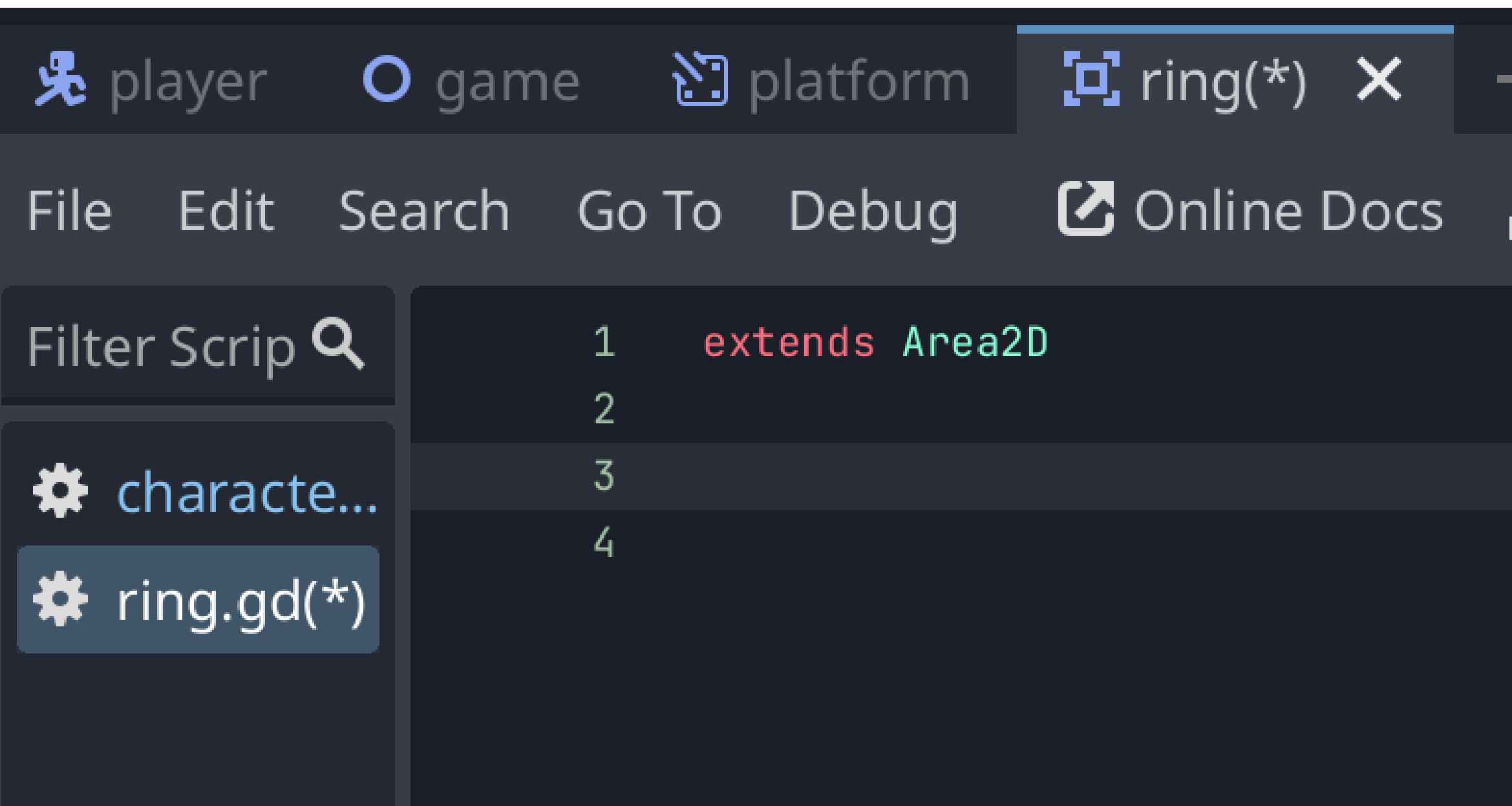


```
1 extends Area2D
2
3
4 # Called when the node enters the scene tree for the first time.
5 func _ready() -> void:
6     pass # Replace with function body.
7
8
9 # Called every frame. 'delta' is the elapsed time since the previous frame.
10 func _process(delta: float) -> void:
11     pass
12
```

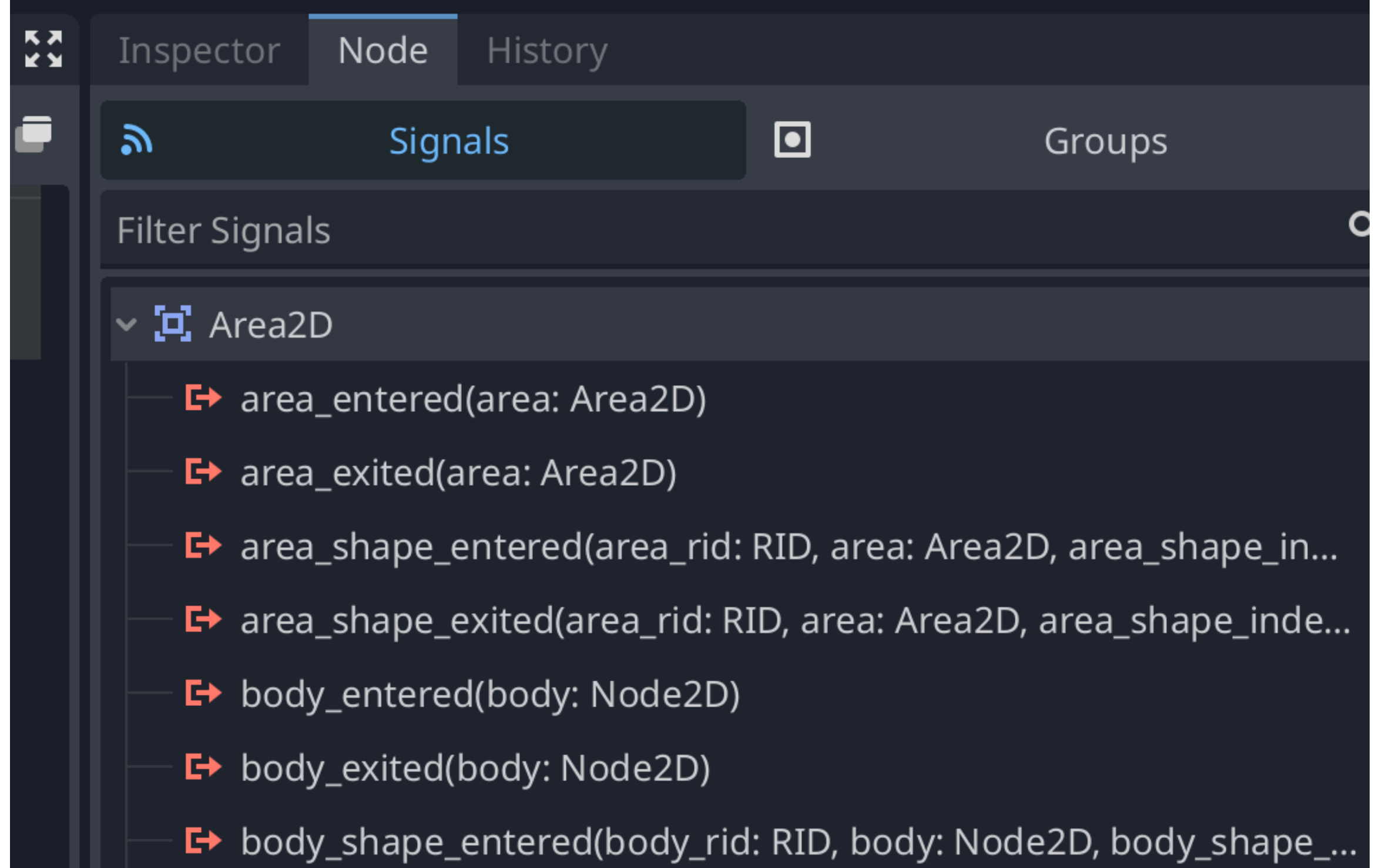
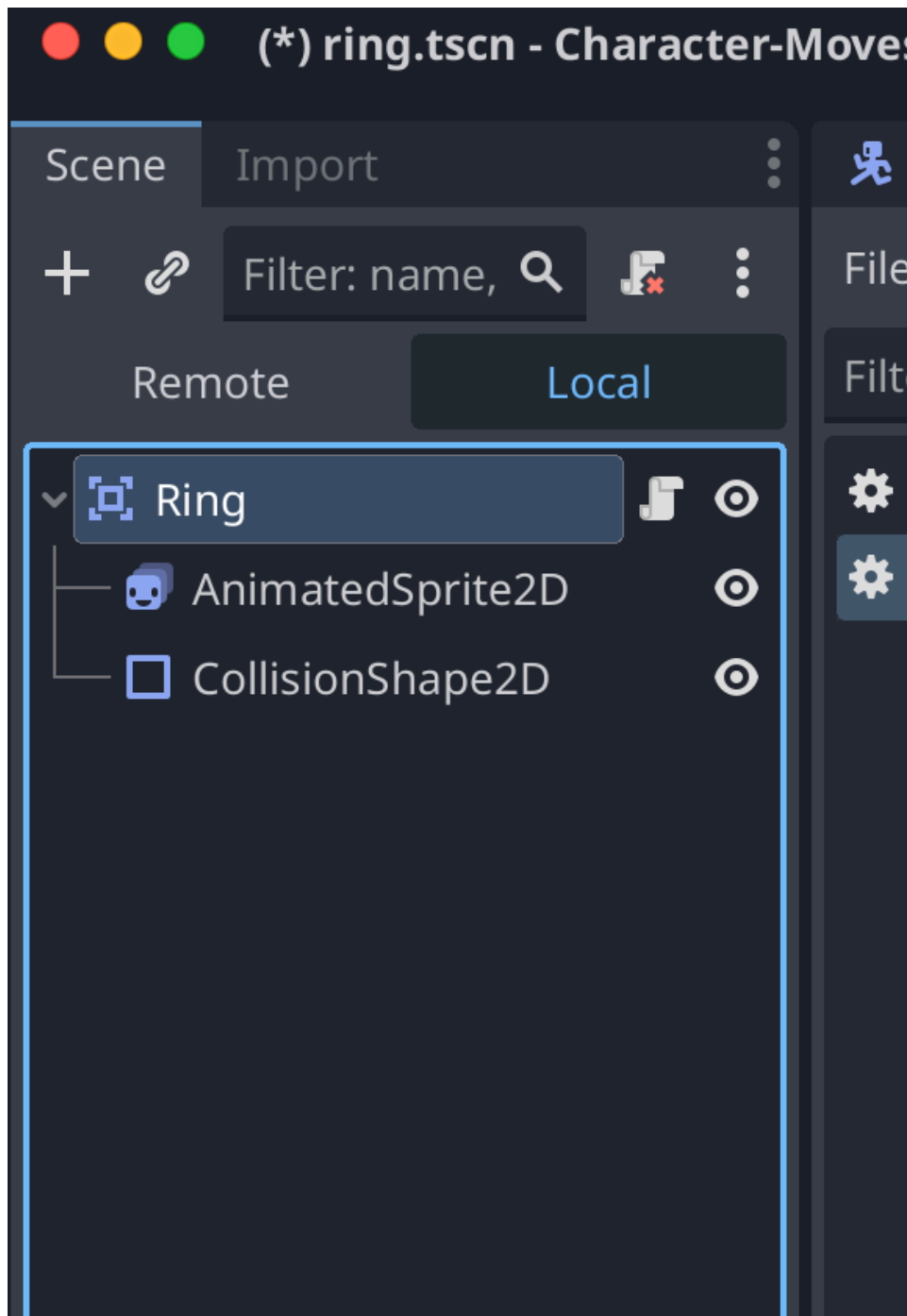
This will create a script that is connected to our Area2D node. Keep this code.

the script below adds a place for functions.

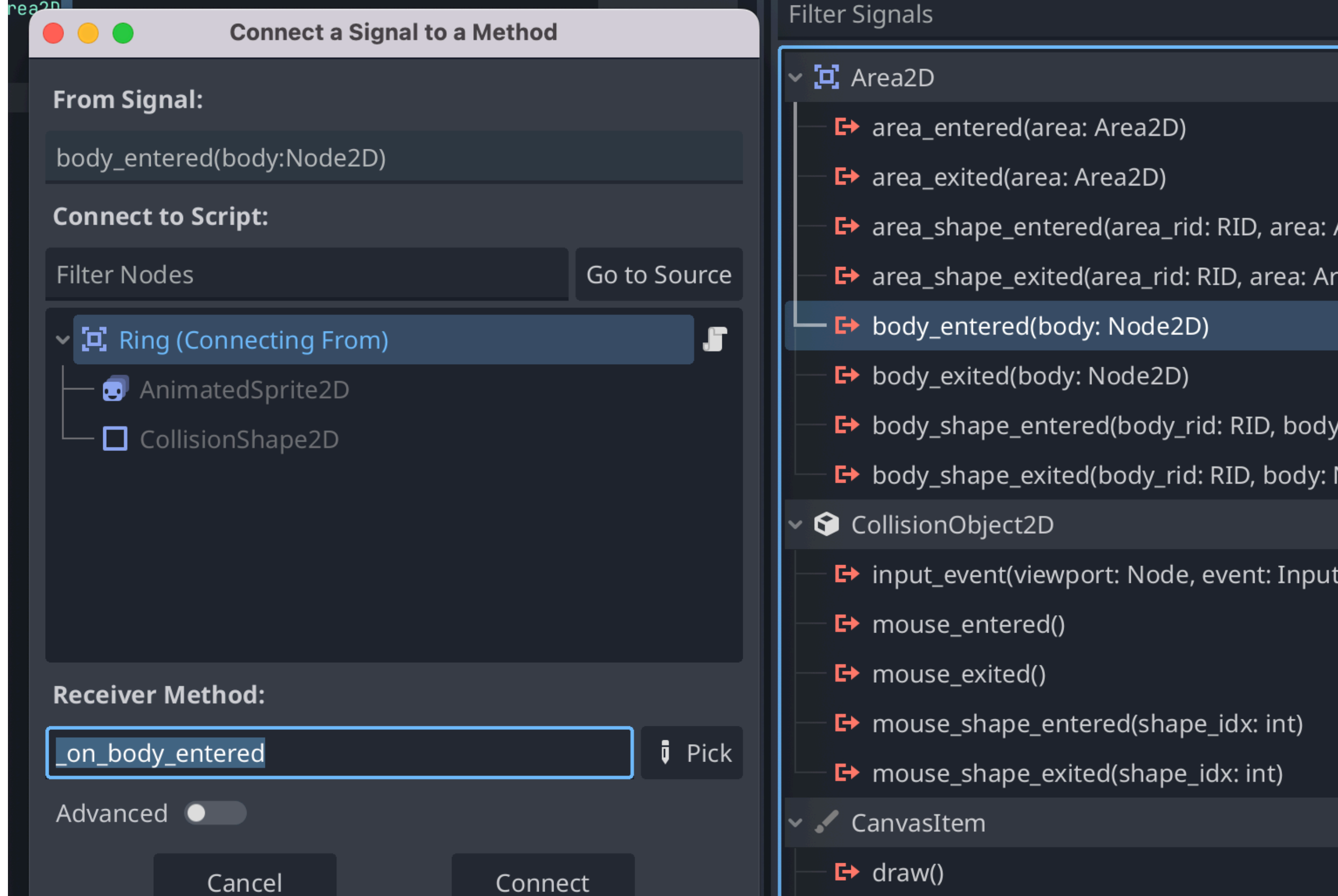
We will remove that and use a SIGNAL instead.



**Keep only the
extends code line**



Click on the NODE and then on the right side of the screen look for the NODE tab and the SIGNALS



Double click on the body_entered signal.
And then in the next window choose CONNECT

```
Go to Debug Online Docs Doc Search
```

```
1  extends Area2D
2
→ 3  func _on_body_entered(body: Node2D) -> void:
4      >|  pass # Replace with function body.
5
```

The Signal adds. new code that creates a function, that will be called when a (body) enters our Ring area.

```
1  extends Area2D
2
→ 3  func _on_body_entered(body):
4      >|  print ("ring collected!")
5
```

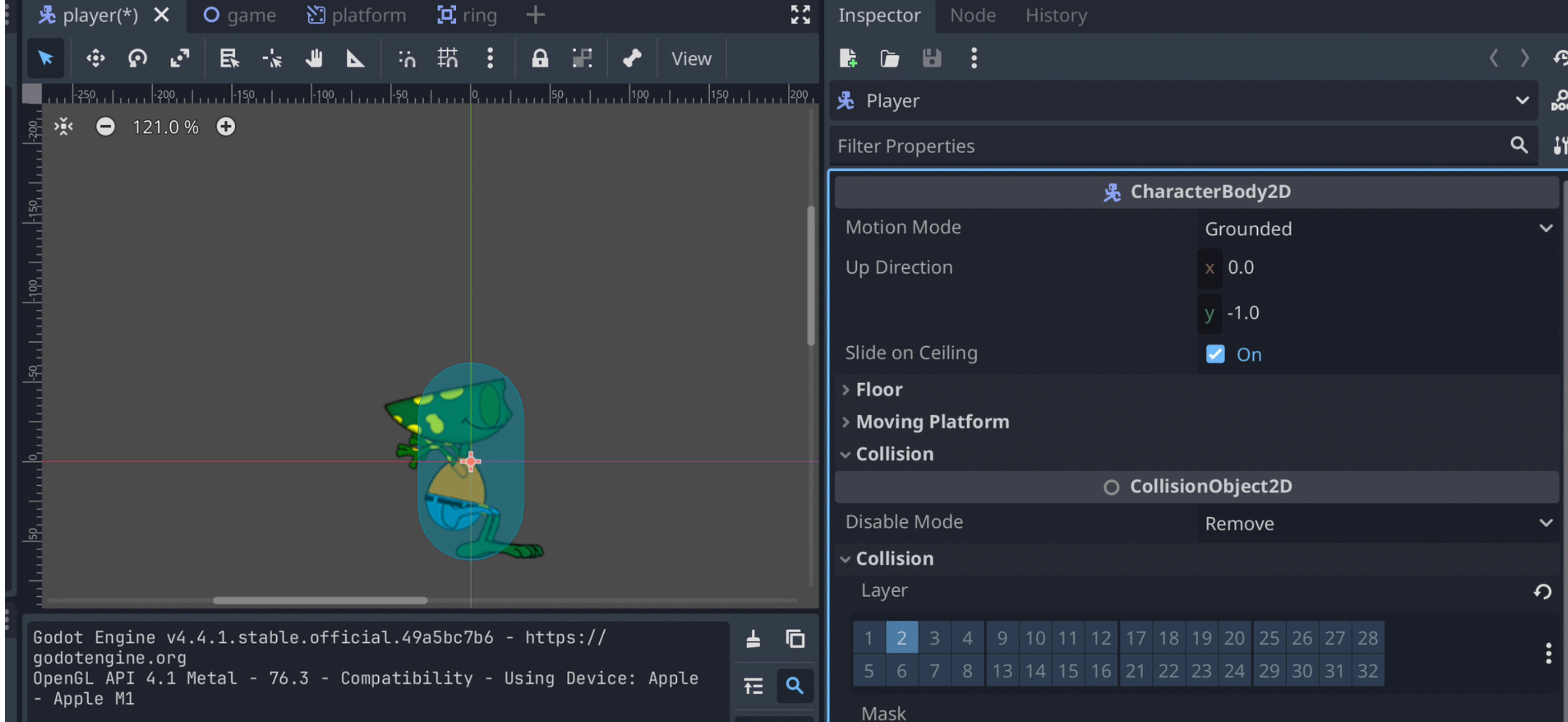
Simply the code to just (body):

**Add a new line underneath to test it.
print (“ring collected!”)**

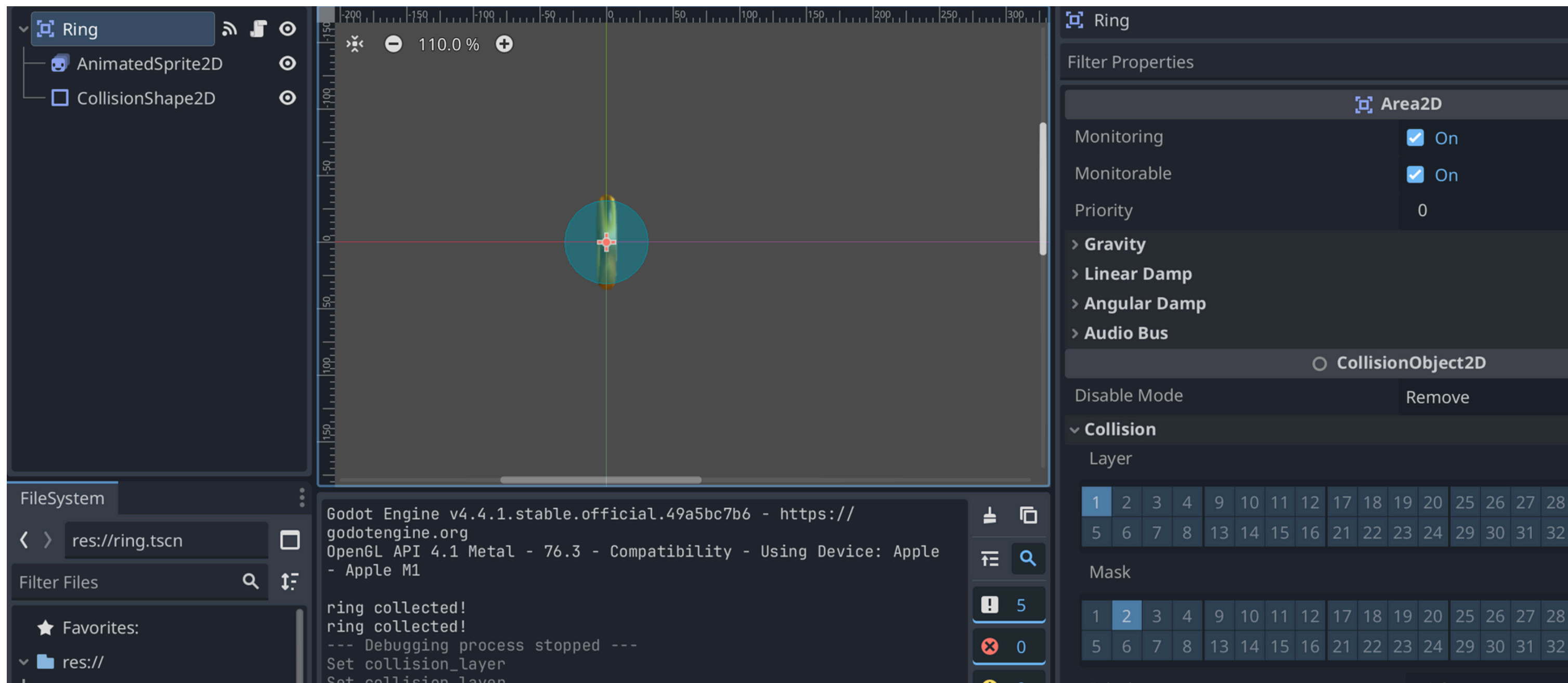


Now, when we test the game, everytime a body enters the ring, it will print our message.

Test it!

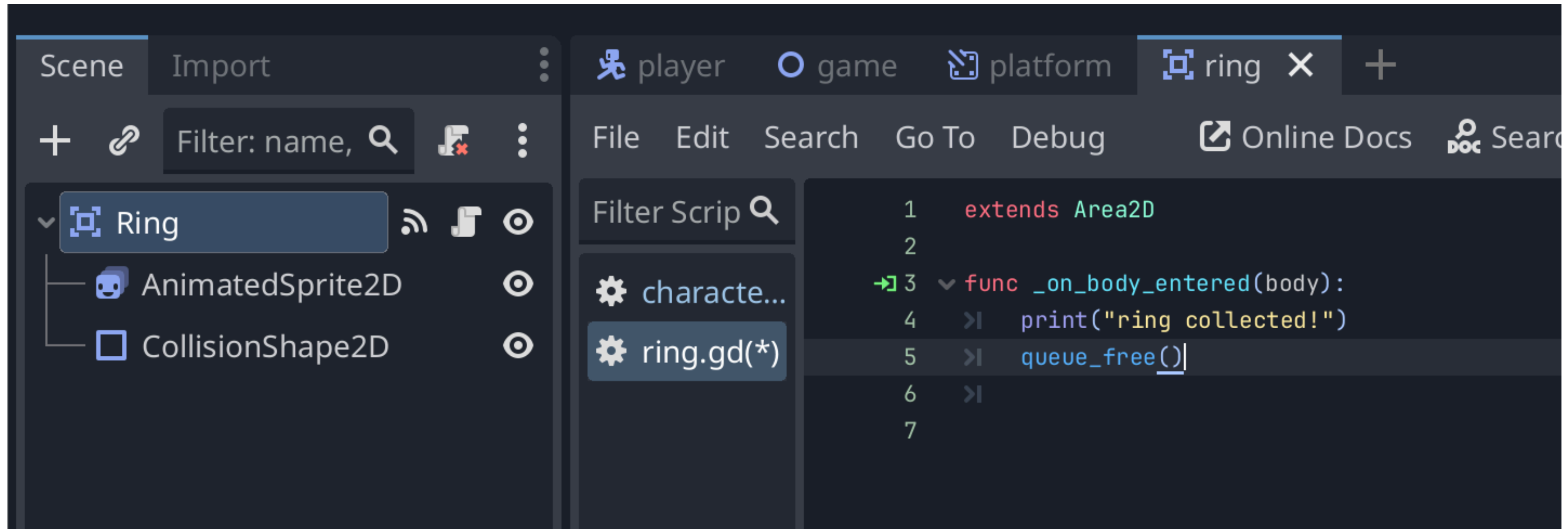


We only want our Player body to activate the ring body, so we will change the Physics layer of our Player to 2. Do this in collision. Save!



We want the ring to listen for this by changing the MASK to also have only layer 2 active.

Save!



**To make the rings disappear as we touch them
add the `queue_free()` code underneath our
current code in the ring**

