

关于移动

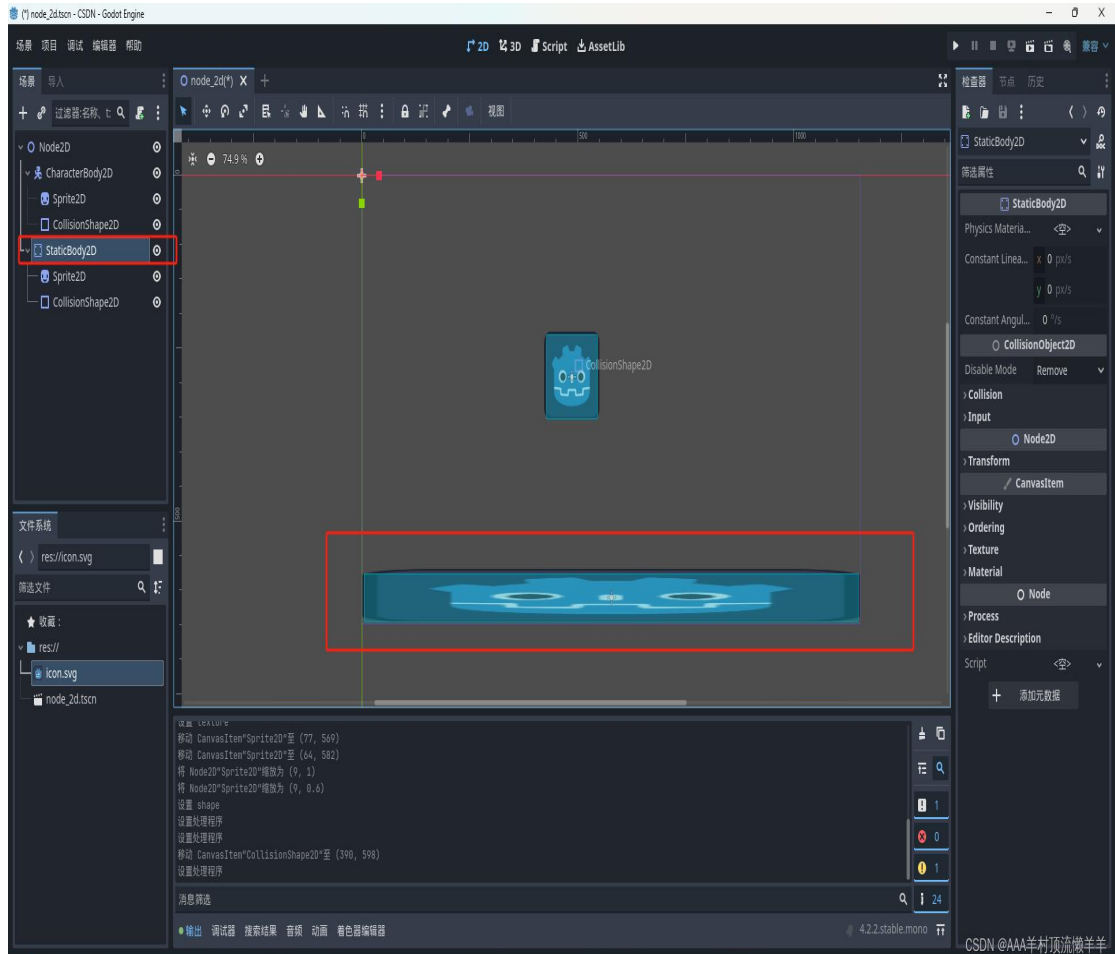
刚打了几把 DOTA2，这版本潮汐还挺好玩的，然后来更新一下笔记 hhhh，上一节内容讲了 2D 俯视角的移动，就是靠四个方向在屏幕中移动。这节课来讲一下 2D 横版移动，类似超级马里奥那种。

创建人物及地面

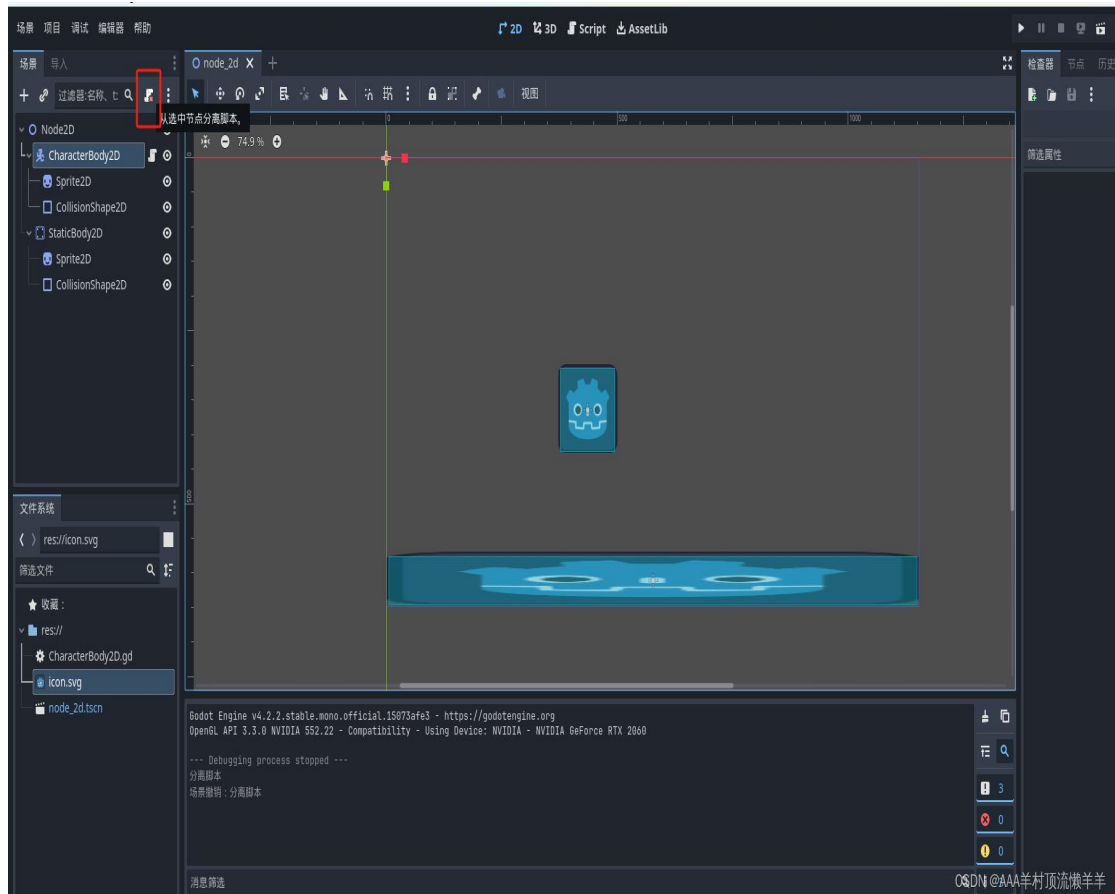
在上一节里讲过创建人物精灵节点，加入 **Sprite2D** 节点放进贴图，加入 **CollisionShape2D** 划出人物的碰撞区，这节课我们继续这样创建一个人物节点。

创建好后，我们在 **Node2D** 节点下创建一个 **StaticBody2D** 节点，这是一个刚体节点，用来作为我们的地面。这里可以看见这个刚体节点和人物节点都属于 **Node2D** 的子节点，所以他们是并列关系。

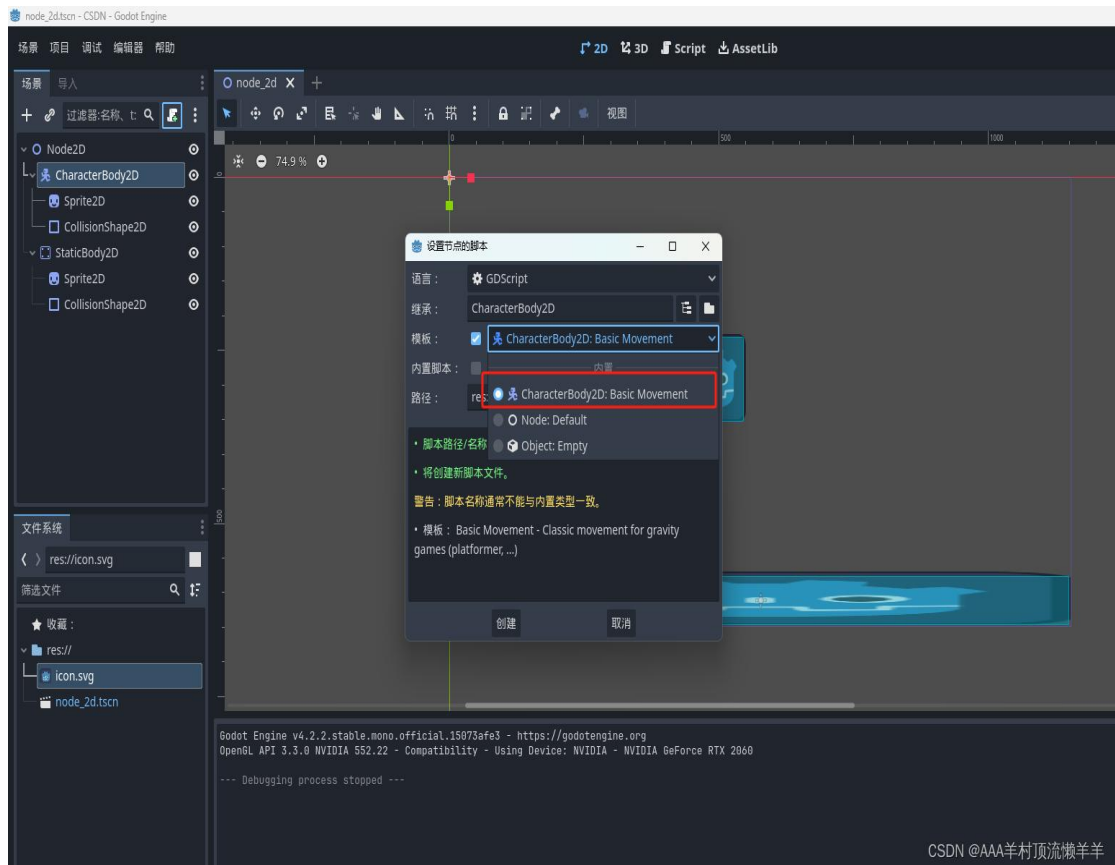
创建好刚体节点后我们用上一节和人物节点同样的方法把碰撞区划好，我们把刚体的图片拉长拖到下面作为地面，碰撞区也拉长。如果没有这个地面那么人物将会一直往下坠落掉出屏幕，所以需要地面挡住。



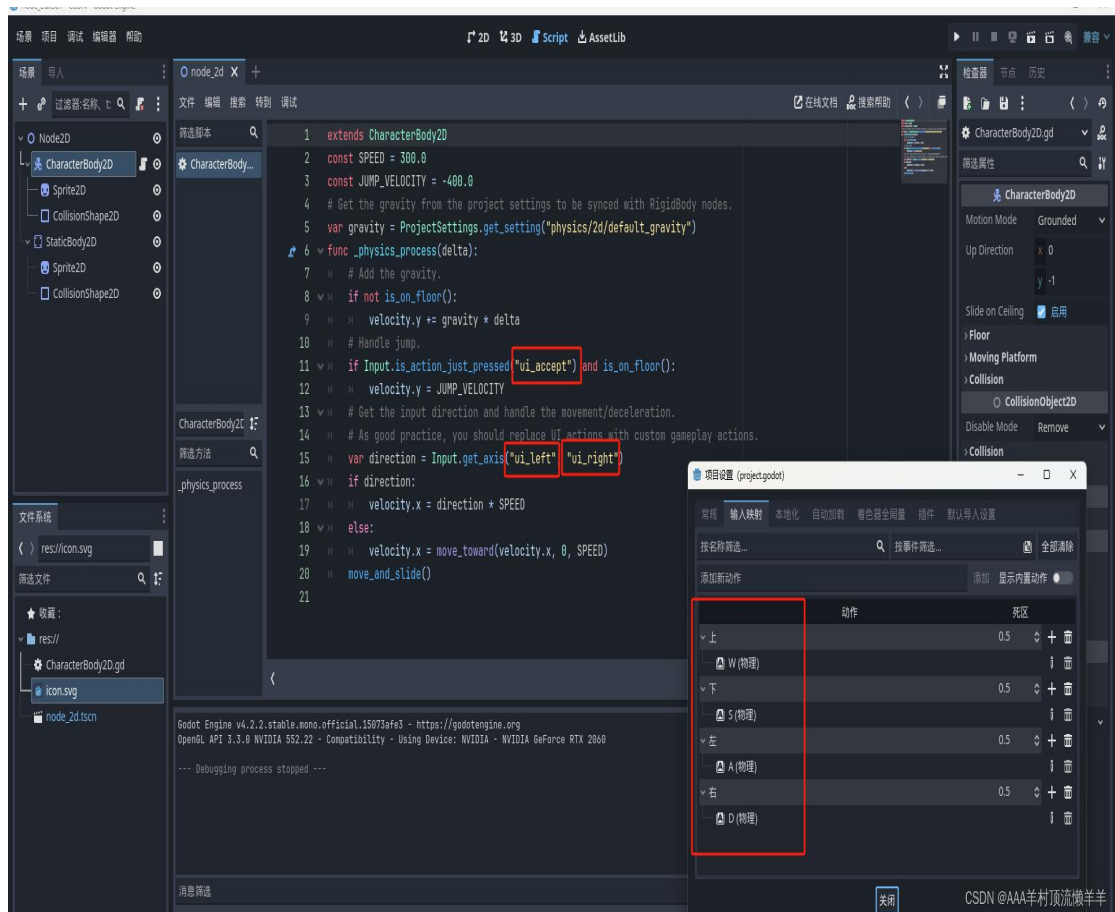
这次我们依旧是在 **CharacterBody2D** 节点中添加脚本，不过 **CharacterBody2D** 已经有上一节留下来的一个脚本，我们可以选中 **CharacterBody2D** 节点，上方有一个分离脚本的按钮（在没有脚本时就是之前添加脚本的按钮），使用这个按钮就可以对脚本进行清除



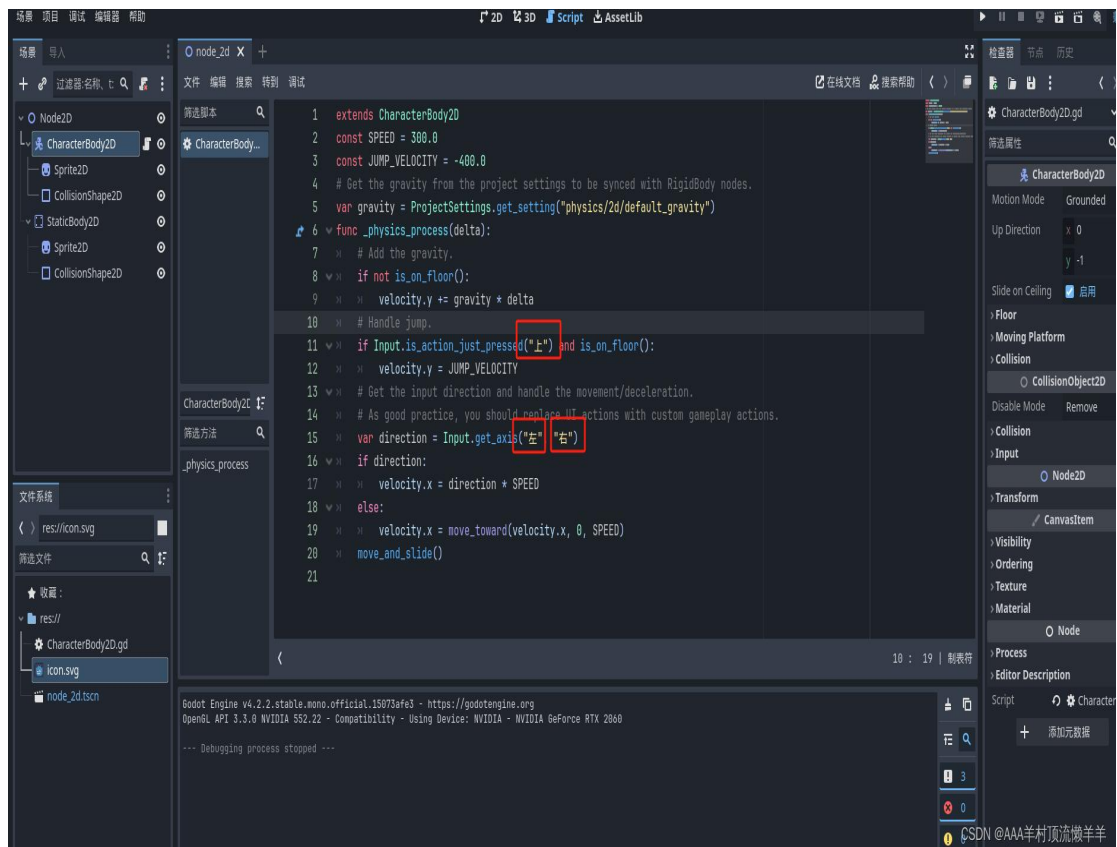
然后我们再对脚本进行添加，这次我们注意可以使用他默认的 **Basic Movement** 这个脚本，这里已经有一部分较为完善的代码，到时候根据自己微调就行了。



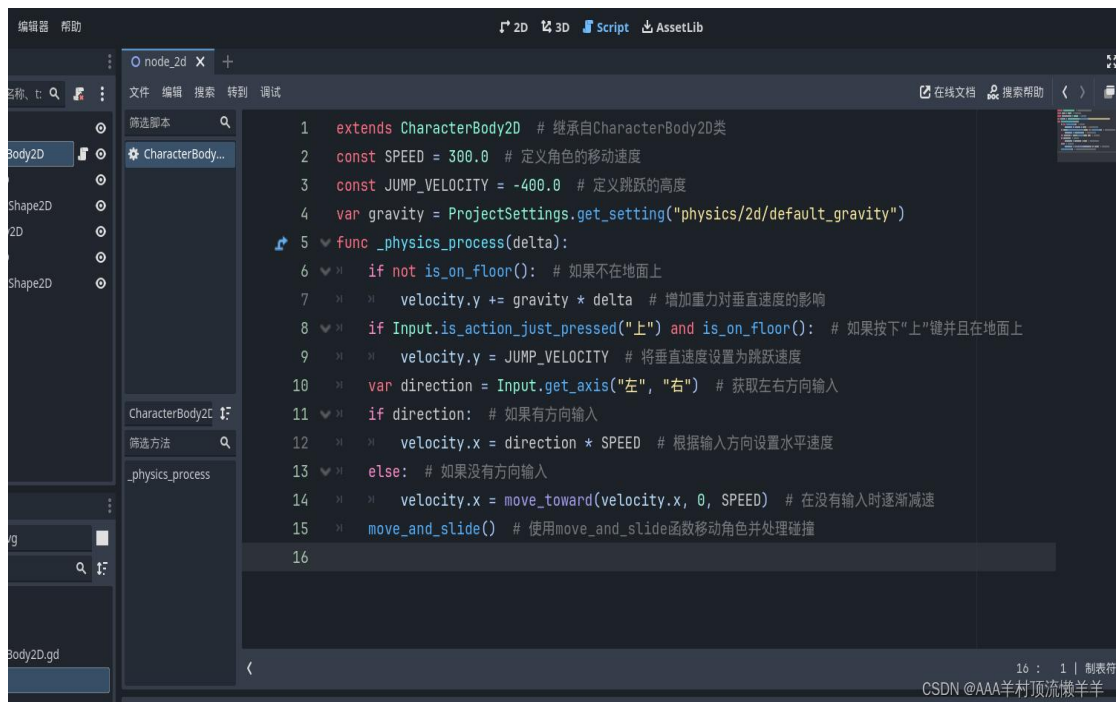
添加好脚本之后我们可以看代码他已经给我们把跳跃，重力，向左向右的代码都写好了，其实是可以直接运行了，但是运行之后默认是小键盘方向的左右移动，空格作为跳跃，我们已经把相关按键的代码框了起来，我们只需要对着我们上一节的输入映射改就好了



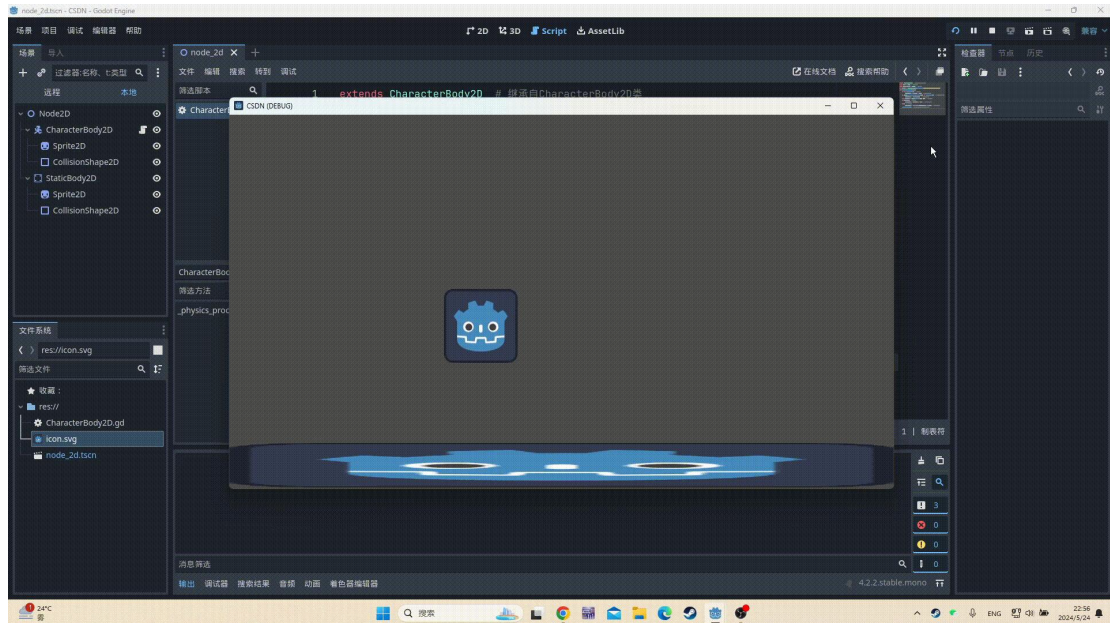
把对应双引号里的内容改成我们设定好的输入映射名称就可以了



如图是已经加上注释的代码，大家可以根据注释自己改移动速度和跳跃高度



预览效果如下



代码部分：

```
extends CharacterBody2D # 继承自 CharacterBody2D 类
const SPEED = 300.0 # 定义角色的移动速度
const JUMP_VELOCITY = -400.0 # 定义跳跃的高度
var gravity =
ProjectSettings.get_setting("physics/2d/default_gravity")
func _physics_process(delta):
    if not is_on_floor(): # 如果不在地面上
        velocity.y += gravity * delta # 增加重力对垂直速度的影响
    if Input.is_action_just_pressed("上") and is_on_floor(): # 如果按下“上”键并且在地面上
        velocity.y = JUMP_VELOCITY # 将垂直速度设置为跳跃速度
    var direction = Input.get_axis("左", "右") # 获取左右方向输入
    if direction: # 如果有方向输入
        velocity.x = direction * SPEED # 根据输入方向设置水平速度
    else: # 如果没有方向输入
        velocity.x = move_toward(velocity.x, 0, SPEED) # 在没有输入时逐渐减速
    move_and_slide() # 使用 move_and_slide 函数移动角色并处理碰撞
```

小结

本节在上一节的基础上，创建了一个刚体作为地面，删除了原先的脚本，重新创建人物精灵节点的脚本文件。对默认脚本进行适当修改，使其可以实现左右移动和跳跃的功能。打完几把 DOTA2 已经有点晚了，所以这节更新的内容不多，很好理解。如有问题欢迎大家指出