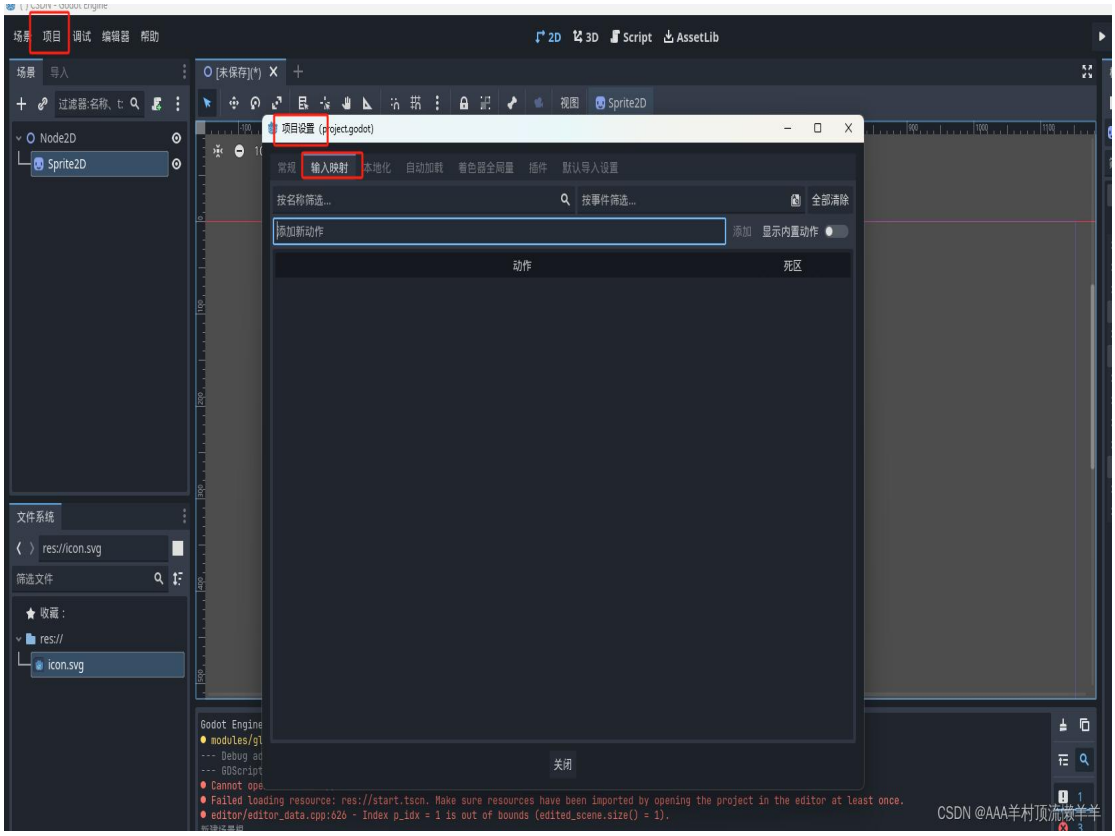
输入及移动

本节教大家如何利用键盘上的按键使图片进行移动

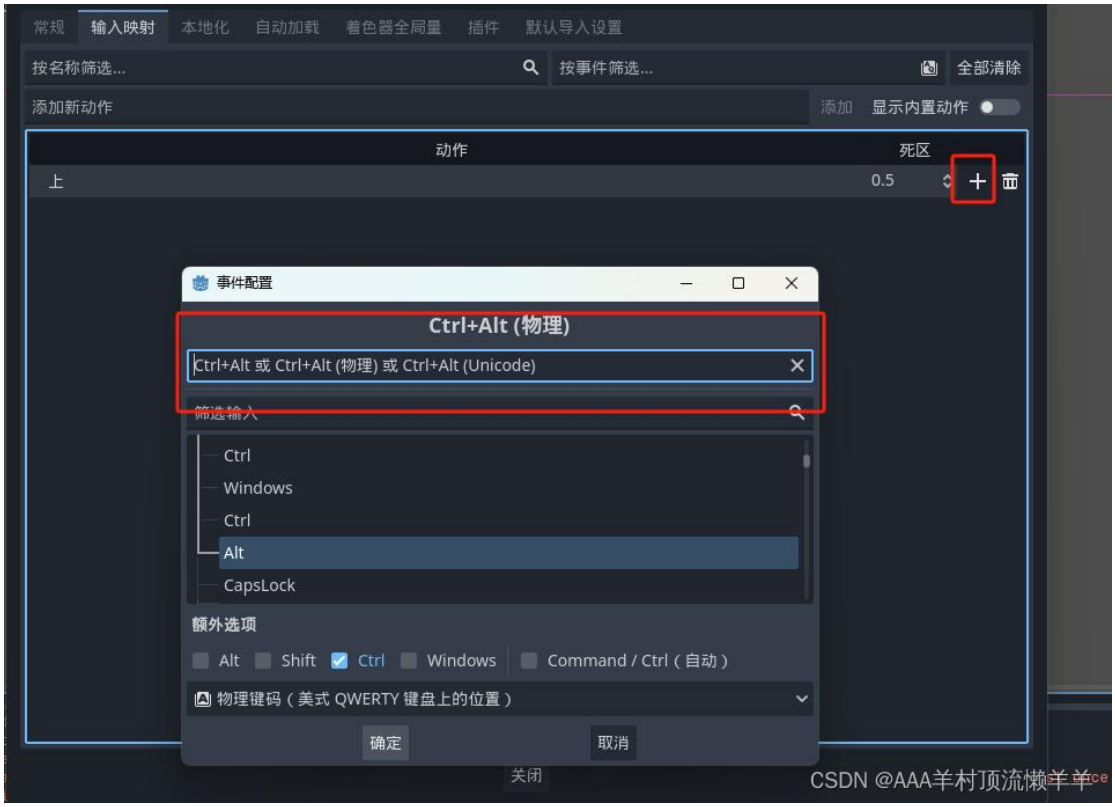
输入映射设置

我们点击左上角的项目→项目设置。最上面选中输入一映射，在此我们可以设置我们需要的按键及对应的功能



在添加新动作这里可以输入按键对应的名称，比如“上”，然后点击右边的添加

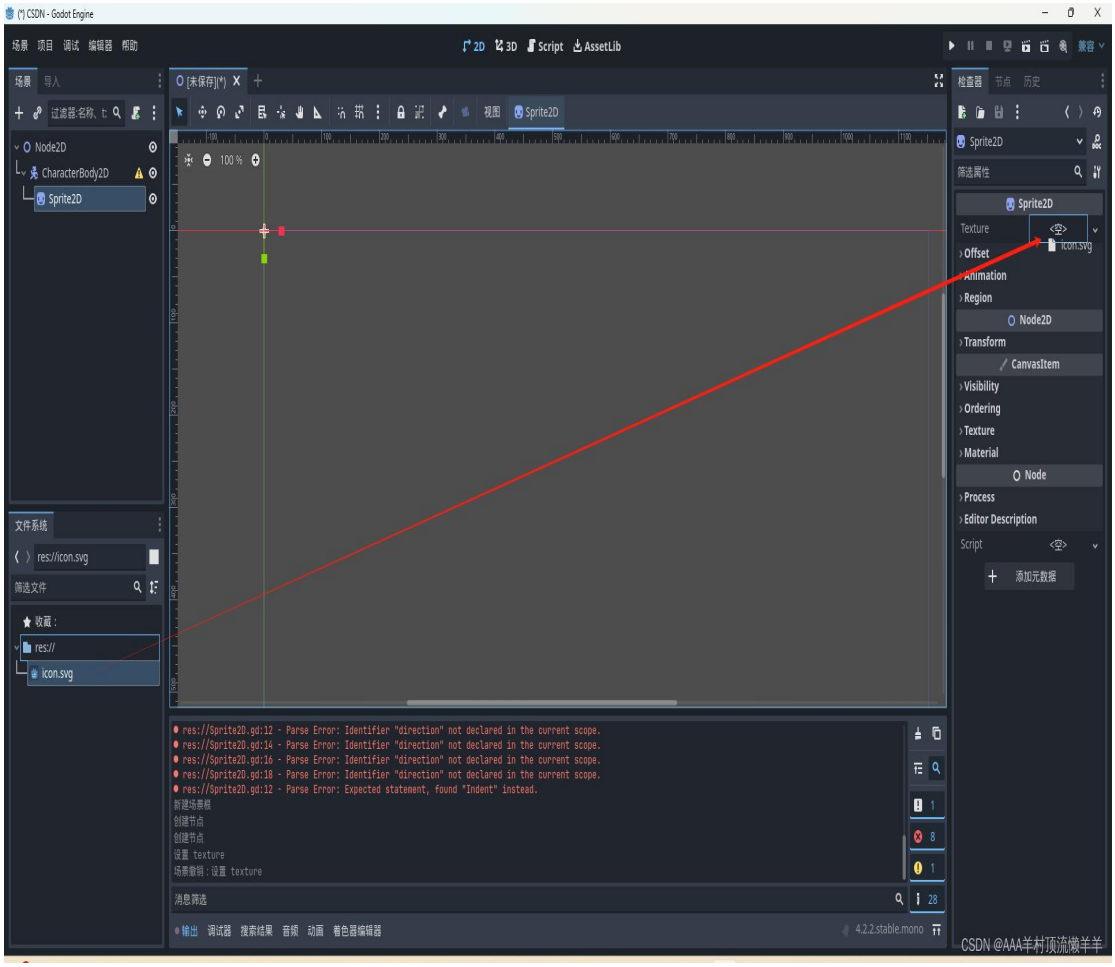
添加之后我们点击这个按键右边对应的加号，跳出事件配置框，鼠标点击一下中间可以输入的这一行，这时候按什么键这一行会跳出你键盘上按了对应的键，比如按“W”，这一行会跳出 W,这时移动鼠标点击下方的确定，就完成了输入映射配置。这样就配置了“上”这个键对应的按键是“W”



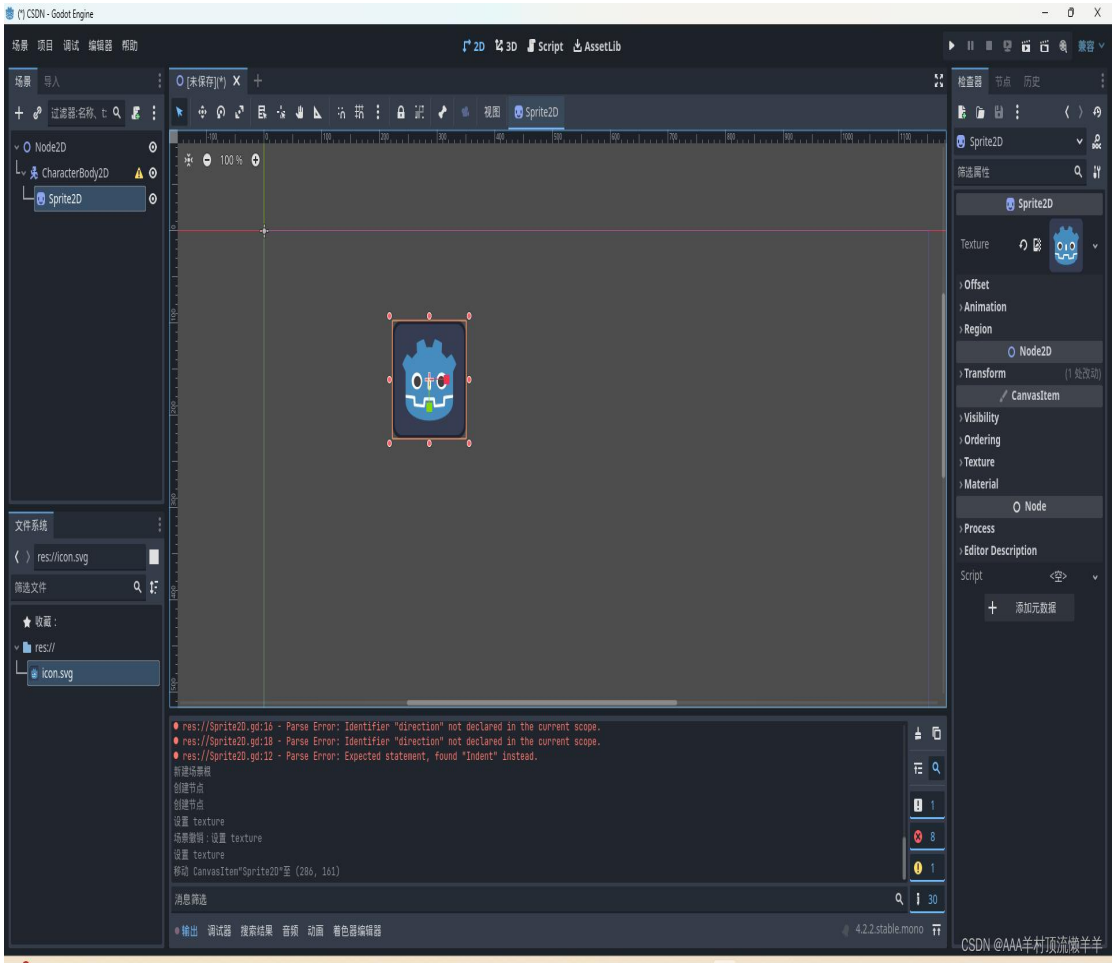
同样的方法我们把“上下左右”对应配置成“WSAD”

添加一个精灵图片

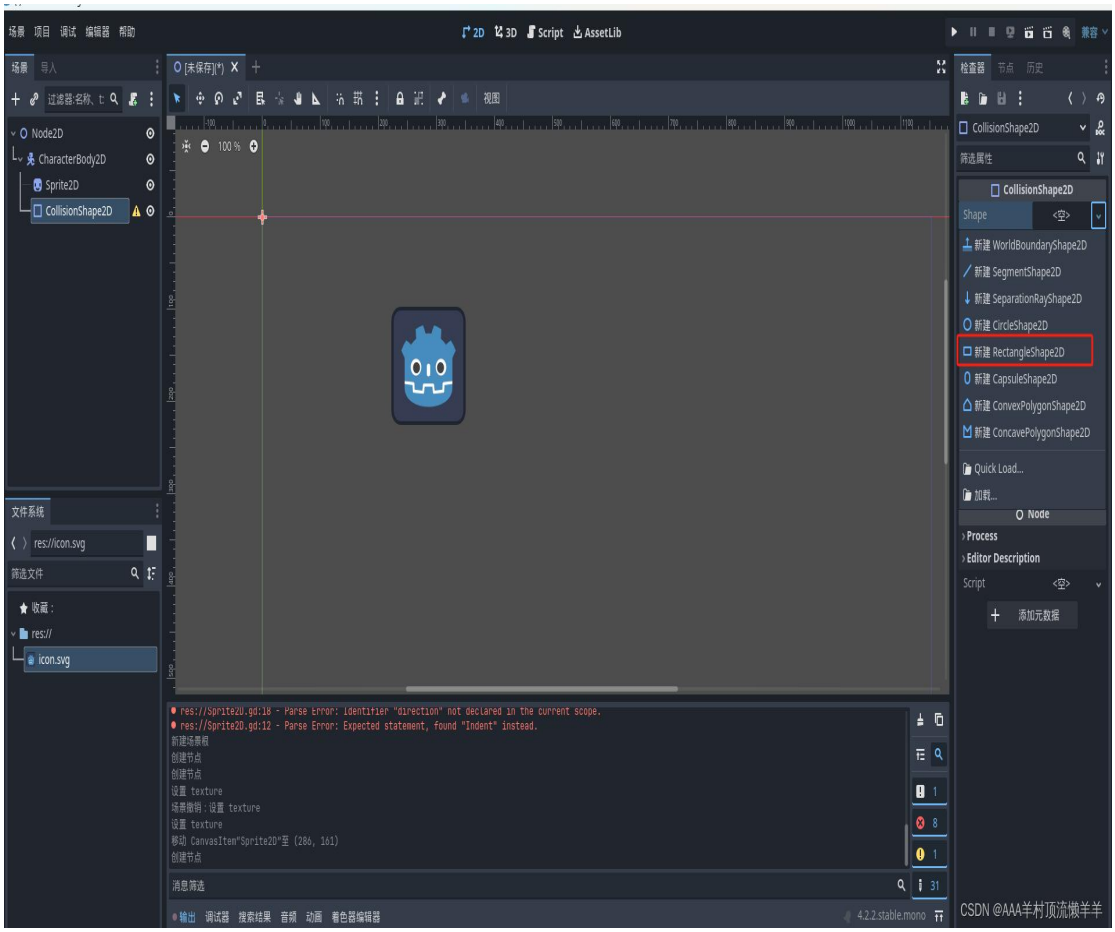
我们在左边 Node2D 节点下添加一个 CharacterBody2D 的节点，然后再该节点下再创建一个叫 Sprite2D 的节点。左下角“文件系统”里有一张创建项目开始自带的图片，我们选中 Sprite2D 这个节点我们把这张图片拖动到这个 2D 精灵节点右侧属性栏的“Texture”（贴图）中



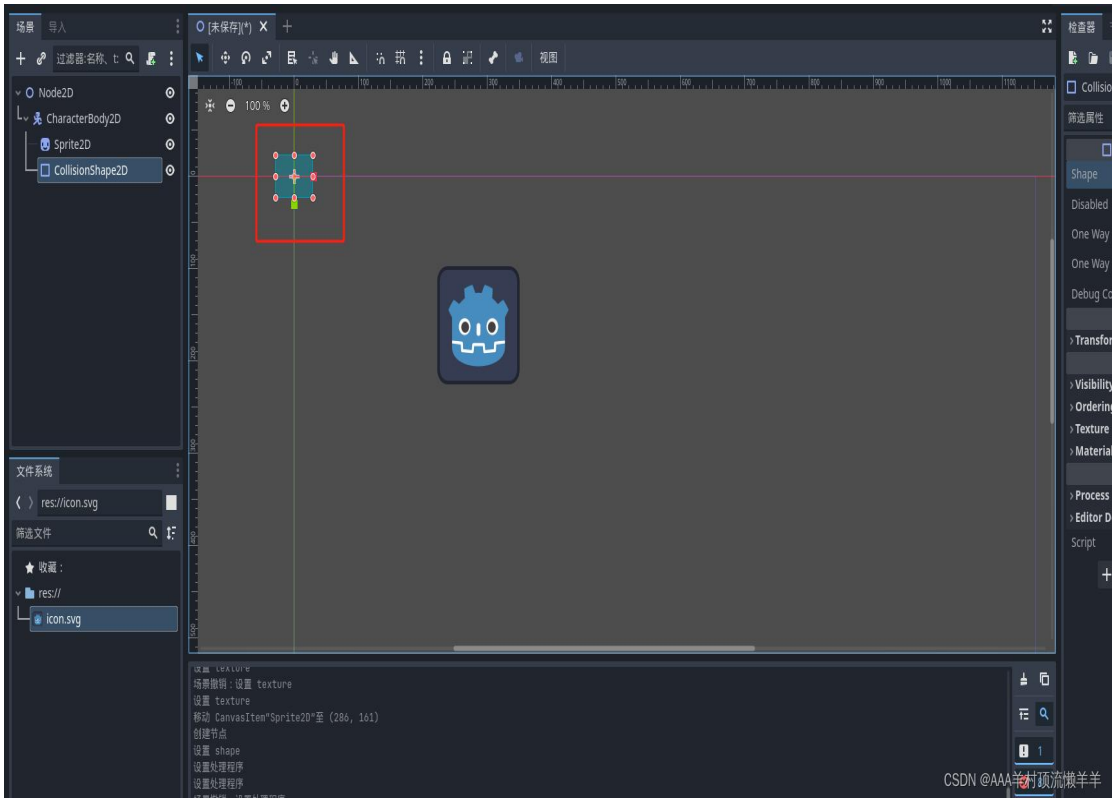
这样一张图片就被加到了这个节点中，可以在预览区中显示出来了，同理我们也可以改成别的图片，我们可以拖动他的位置

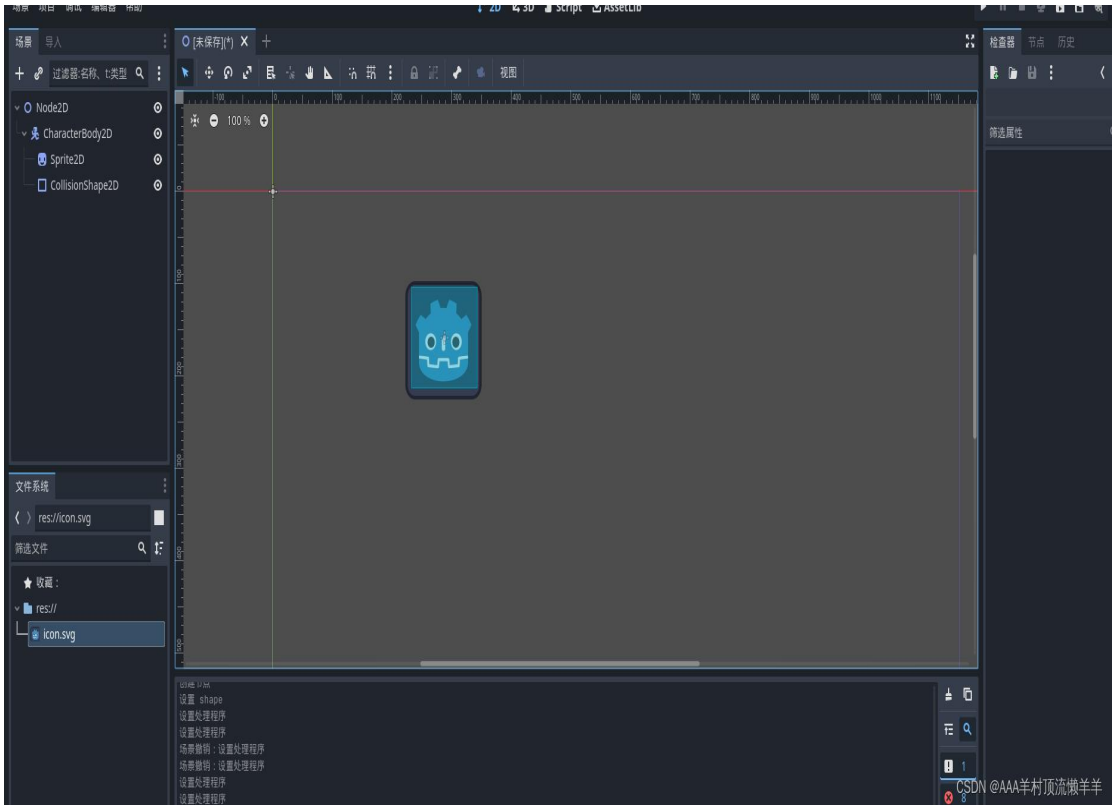


我们看见左边节点还是显示一个感叹号，因为我们最开始添加的 CharacterBody2D 这个小人一样的节点还需要一个碰撞区域，我们再添加一个叫 CollisionShape2D 的节点来给小人加个碰撞区域，添加好后选择右边 Shape 属性，选个差不多的形状



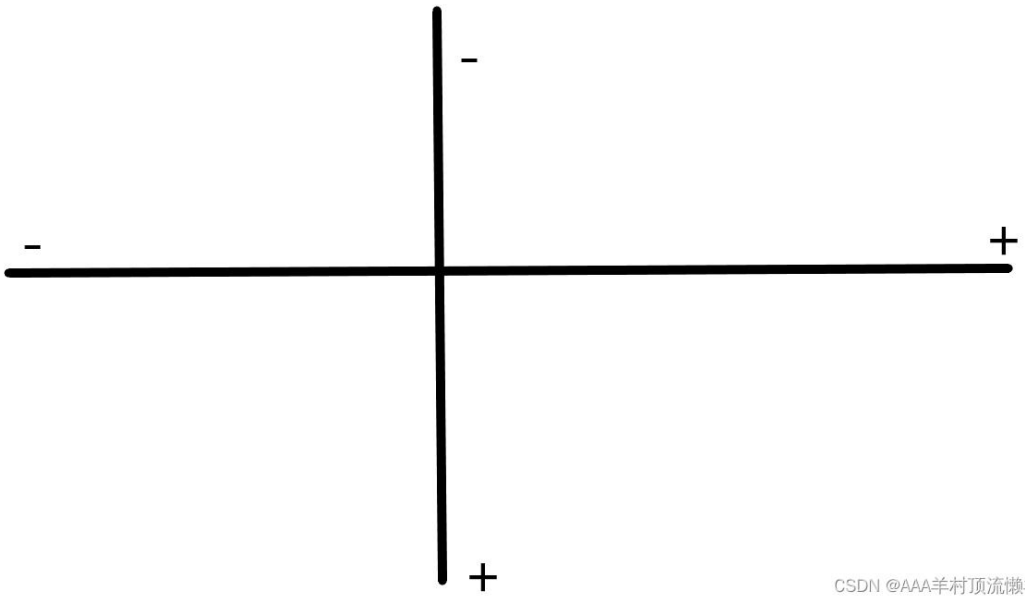
选好形状之后把弹出来的矩形框缩放大小，放的跟我们的贴图差不多大，然后拖给这个贴图就行


场景
导入
过滤器:名称、t:
Node2D
CharacterBody2D
Sprite2D
CollisionShape2D
[未保存](*)
视图
100 %
检查器
筛选属性
Shape
Disabled
One Way
One Way
Debug Co
Transfor
Visibility
Orderin
Texture
Materia
Process
Editor D
Script
文件系统
res://icon.svg
筛选文件
收藏：
res://
icon.svg
场景撤销：设置 texture
设置 texture
移动 CanvasItem"Sprite2D"至 (286, 161)
创建节点
设置 shape
设置处理程序
设置处理程序
1
CSDN @AAA羊村顶流懒羊羊

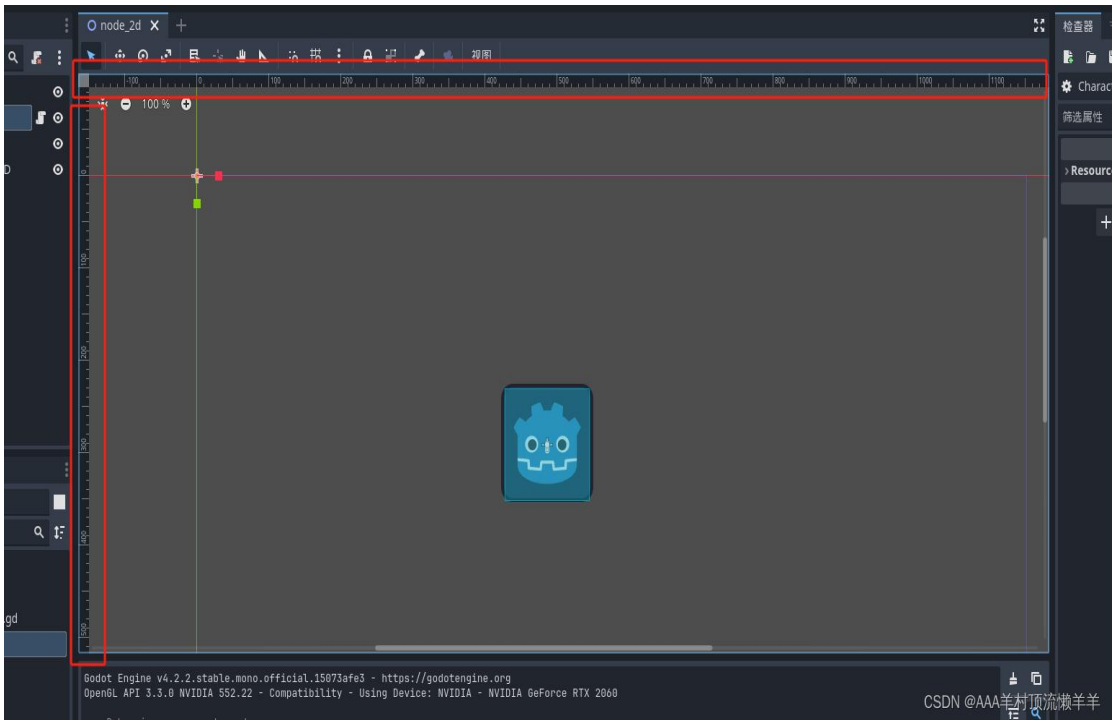

2D
3D
Script
AssetLib
场景
导入
过滤器:名称、t:类型
Node2D
CharacterBody2D
Sprite2D
CollisionShape2D
[未保存](*)
视图
100 %
检查器
节点
历史
筛选属性
文件系统
res://icon.svg
筛选文件
收藏：
res://
icon.svg
设置 shape
设置处理程序
设置处理程序
场景撤销：设置处理程序
场景撤销：设置处理程序
设置处理程序
设置处理程序
1
CSDN @AAA羊村顶流懒羊羊

精灵移动

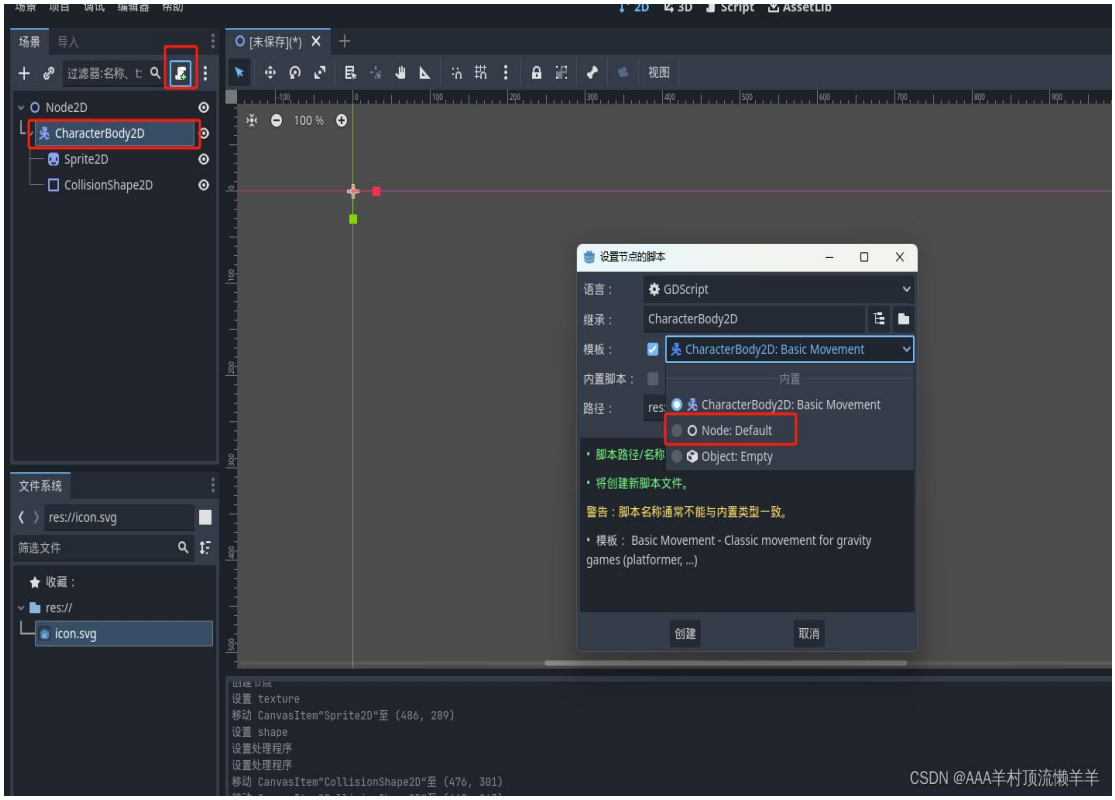
我首先要了解原理，我们在游戏中有一个坐标轴，控制精灵在坐标轴上的位置也就实现了移动，在这个引擎中坐标轴分为横轴和纵轴，横轴右边为正，左边为负。纵轴下面为正，上面为负。



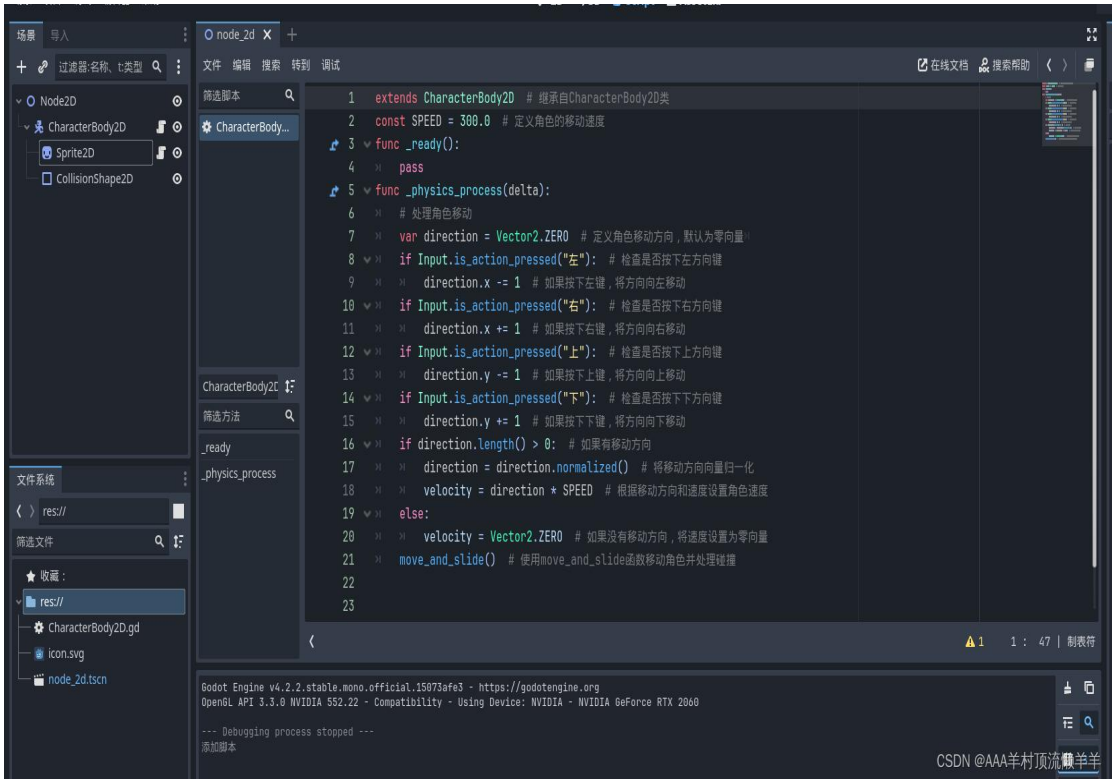
可以在边框上看到具体横纵轴刻度



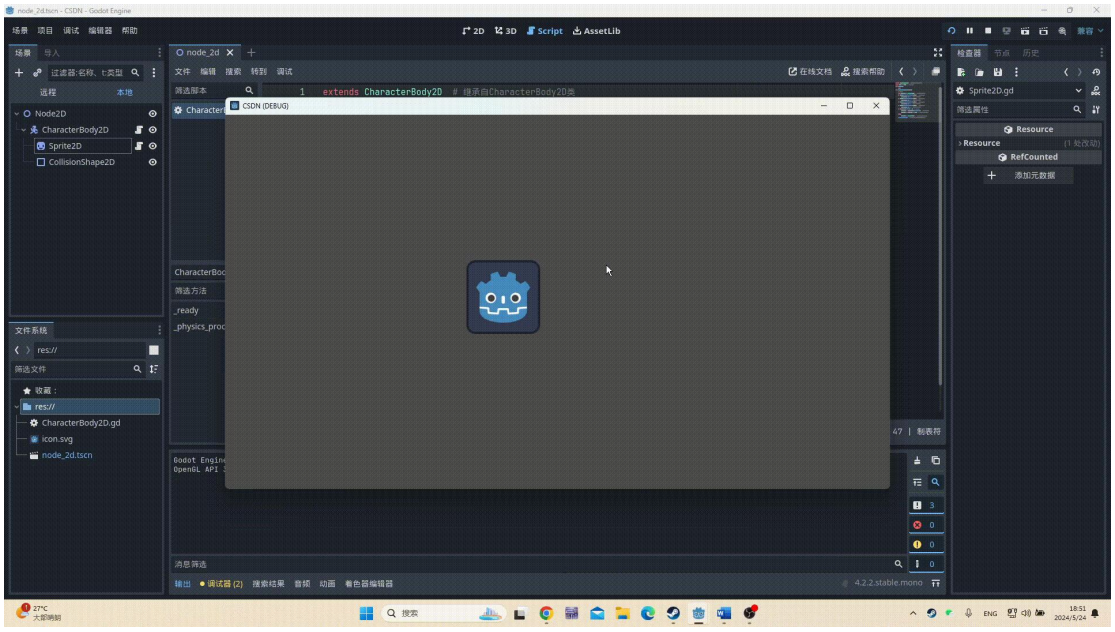
我们选中 CharacterBody2D 节点为他添加一个脚本，这里看到多了一个 Basic Movement 的脚本，这是这个节点默认给的一个移动脚本，这个脚本还带有重力，我们这次的教学不需要重力，所以选择个有基础模版的 Default 脚本即可



代码及注释如下，按下对应的按键对对应的横纵轴进行加减来实现移动



运行后效果预览



代码部分：

```
extends CharacterBody2D  # 继承自 CharacterBody2D 类
const SPEED = 300.0  # 定义角色的移动速度
func _ready():
```

```
        pass
func _physics_process(delta):
        # 处理角色移动
        var direction = Vector2.ZERO  # 定义角色移动方向，默认为零向量
        if Input.is_action_pressed("左"):  # 检查是否按下左方向键
                direction.x -= 1  # 如果按下左键，将方向向左移动
        if Input.is_action_pressed("右"):  # 检查是否按下右方向键
                direction.x += 1  # 如果按下右键，将方向向右移动
        if Input.is_action_pressed("上"):  # 检查是否按下上方向键
                direction.y -= 1  # 如果按下上键，将方向向上移动
        if Input.is_action_pressed("下"):  # 检查是否按下下方向键
                direction.y += 1  # 如果按下下键，将方向向下移动
        if direction.length() > 0:  # 如果有移动方向
                direction = direction.normalized()  # 将移动方向向量归一化
                velocity = direction * SPEED  # 根据移动方向和速度设置角色
速度
        else:
                velocity = Vector2.ZERO  # 如果没有移动方向，将速度设置为零
向量
        move_and_slide()  # 使用 move_and_slide 函数移动角色并处理碰撞
```

小结

本节先讲了如何进行输入映射，然后讲了如何建立一个 2D 的精灵节点也就是一个人物节点，给他如何加上碰撞体积，并写入脚本，通过输入控制移动。这里的移动是上下左右的移动，使用于 2D 俯视角游戏。还有一种加入重力的移动，适用于 2D 横版过关，类似经典的超级马里奥游戏，这种移动方法会在下期讲，如有不足欢迎大家指出。