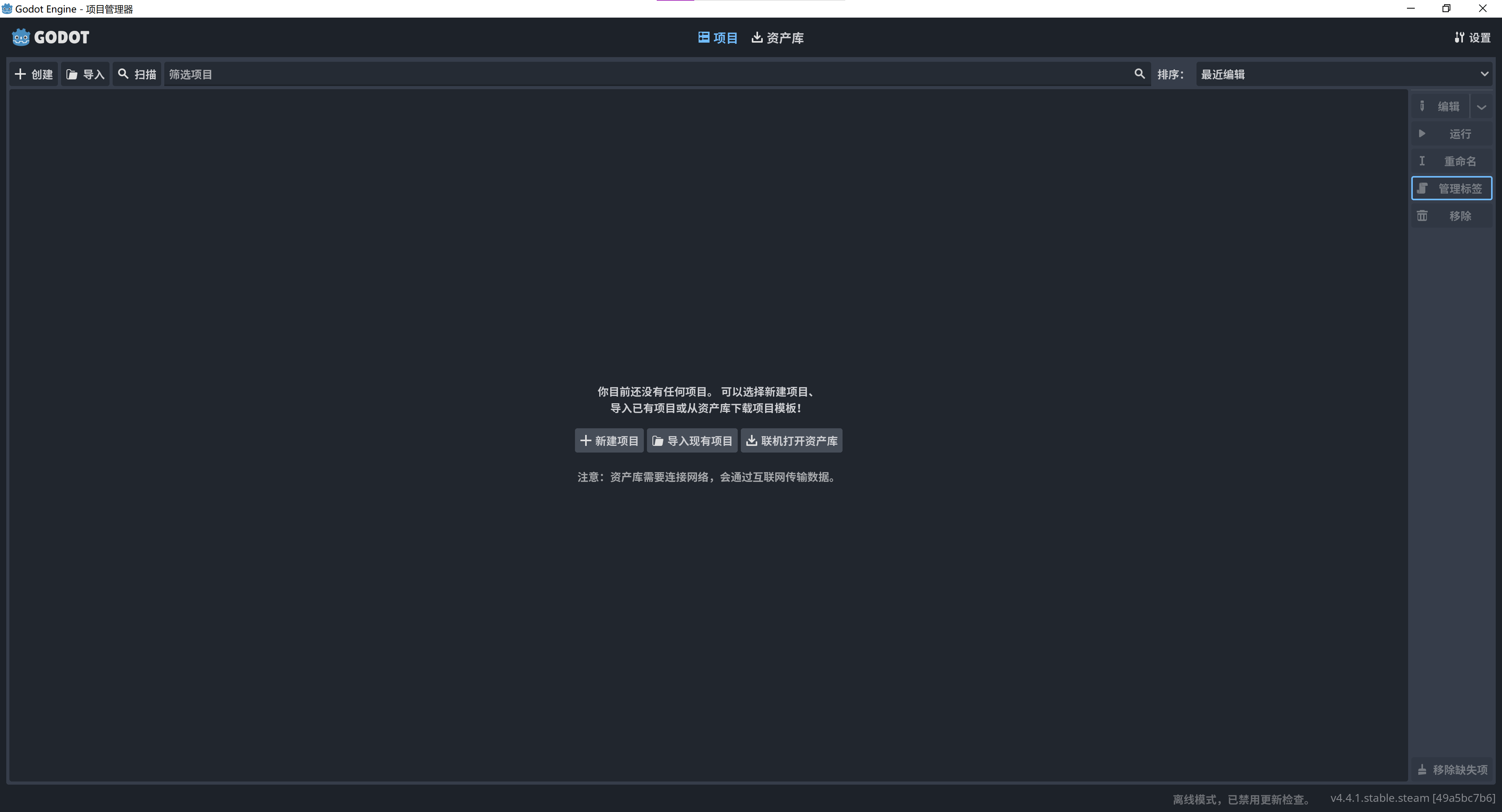
首先是下载游戏引擎，可以直接在Steam上下载

下载完成后打开



这里的资产库，类似于fab的存在

Godot支持C#编写，但本文使用GDScript来完成

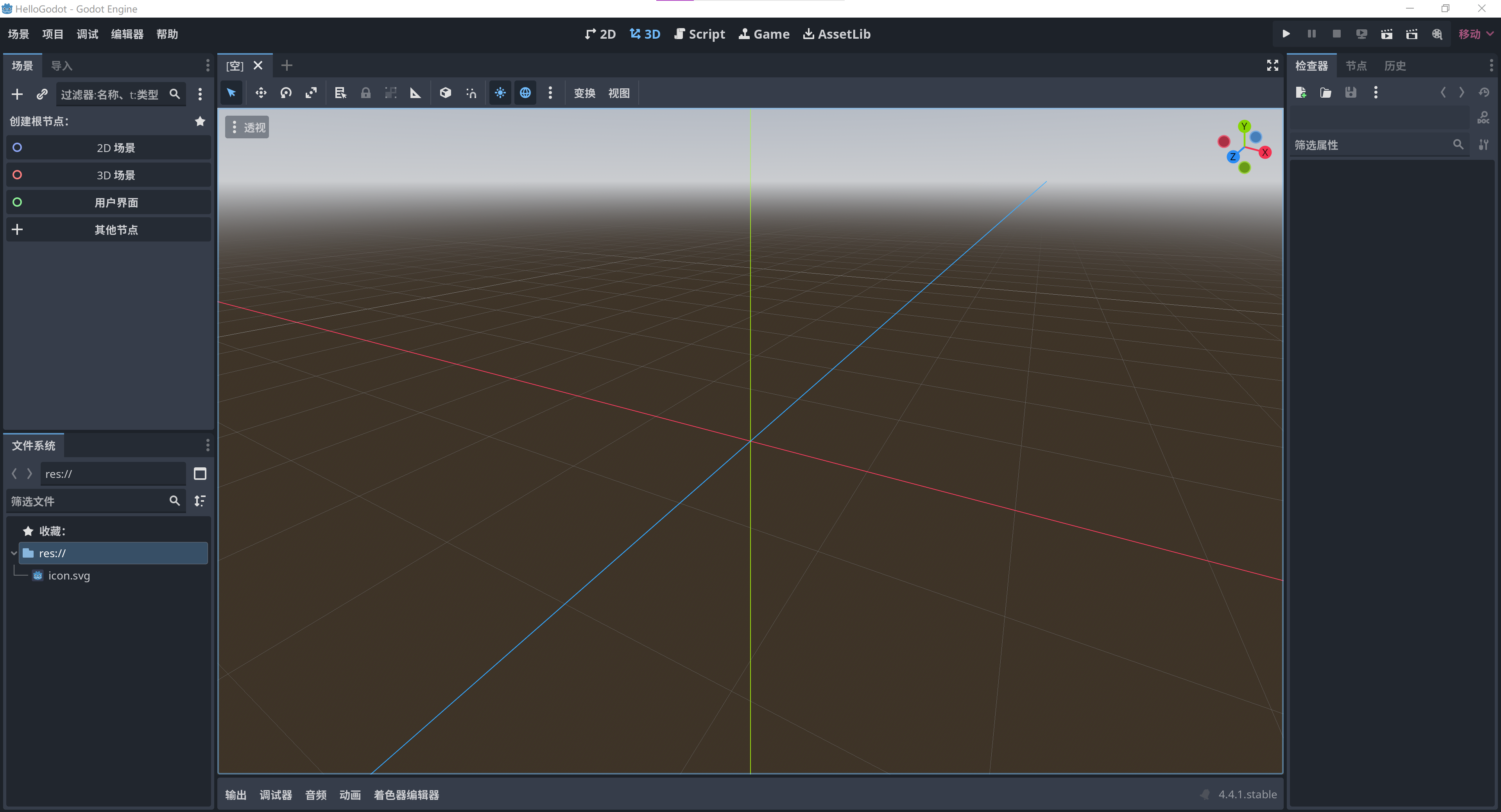
我们创建新项目



我们做的Godot，默认支持移动端和桌面端。移动模式能导出到手机和电脑，兼容模式甚至能将游戏导出到网页。

下面的教程从简单的2D场景入手，所以这里选择移动渲染器模式。

创建后就会进入项目页面



左上角用于浏览场景以及创建场景根节点

左下角时文件浏览器，用于浏览游戏资产

右侧是大纲等

中间是场景

中间的上面可以切换场景编辑器页面。2D模式和3D模式分别编辑不用的游戏场景。Script是脚本编辑器。AssetLib是资产平台，类似fab

## 2D场景

左上角创建一个2D场景，就会在场景中心创建一个节点

（godot中，节点，就是实现特定功能的类）

在右侧的属性浏览器中可以看到相关属性

如Transform中的Position就是当前节点相对于父节点的位置信息。父节点就是在场景树里位于上一级的那个节点

在移动节点位置时，移动的坐标其实是相对于父节点的坐标

搭建2D场景时，需要考虑窗口视口的大小。在2D场景浏览器我们会看到一个边框，这就是我们当前视口的大小，在项目设置中可以看到，默认值为1152\*648，可以修改

然后保存该2D场景

场景文件的后缀名为.tscn

将场景保存之后，我们就可以运行游戏了。在右上角运行

在运行之前，需要先给游戏指定一个主场景，也就是指定游戏打开后的第一个进入的场景。可以在项目设置中直接搜索“主场景”

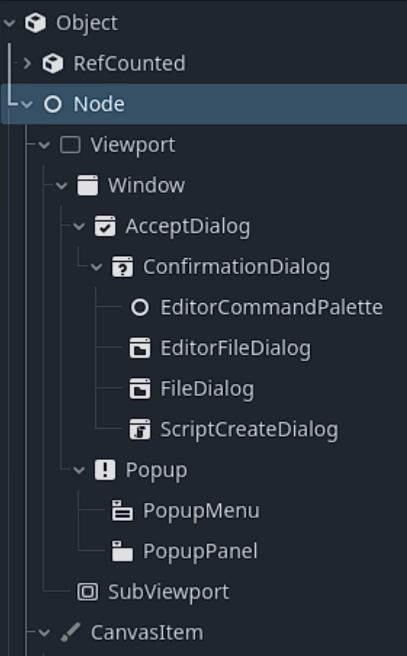
## 给场景添加子节点

右键左上角的2D场景节点，选择创建子节点。我们创建一个Sprite2D节点。这个节点负责显示一个2D的图片

属性面板中的Texture是纹理，我们可以将文件系统中的svg直接拖进去赋值。

属性面板中也有其父节点的相关属性，来定义该节点与其父节点的相关属性

（Godot也是用C++开发的，所以另一种说法，在创建节点的时候可以看出来，引擎中的Node2D类，是Sprite2D类的父类，所以子类携带父类的相关属性。与UE相似，基本上所有类的祖宗类都是Object类）



## 场景树

按下F1键可以寻求帮助

搜索SceneTree

作为最重要的类之一，SceneTree 管理着场景中节点的层次结构以及场景本身。节点可以被添加、获取和移除。整个场景树可以被暂停，包括当前场景。场景可以被加载、切换和重新加载。

你也可以使用 SceneTree 将你的节点组织成组：每个节点都可以被添加到你想要创建的任意多个组中，例如“敌人”组。然后你可以遍历这些组，甚至可以在属于任何给定组的所有节点上调用方法并设置属性。

SceneTree 是引擎所使用的默认 MainLoop 实现，因此负责游戏循环。

SceneTree可以控制游戏程序的大部分运行流程。我们编辑的场景，其实就是位于场景树下的，由SceneTree进行管理。

简单的说，SceneTree管理游戏的当前场景，我们编辑的场景的根节点在游戏运行时都是由SceneTree管理的

运行一下游戏，同时观察左上角，出现了“远程”和“本地”两个东西



本地代表我们编辑的场景的节点树，

远程页面展示了游戏运行中处于场景树内的所有节点

在游戏开始时，场景树会默认创建一个root节点，这个节点就是我们当前程序窗口节点

而当前的场景位于root节点下，也就是我们编辑的主场景Node2D



打开游戏时看到的窗口就是root节点，它是一个Window类型的节点

Godot在开始游戏后，会先生成一个场景树，然后场景树创建一个root节点，最后把我们制作的场景加入到root中

## 导入资产

导入资产很简单，和UE一样直接拉到文件系统中就行

## 项目设置

可以自己看一下项目设置中的东西