

# 1.前言

经过前几周的学习，各位对于游戏开发在理论上都有了有一定的了解，也许已经开始期待之后的合营环节，想要和程序美术同学一块合作正式制作属于自己的游戏了。

不过在此之前，还需要经历一个小小的学习，那便是对游戏引擎的实际使用。无论是了解程序功能的简单原理、以便考量功能制作时的工期长短实现难度，亦或是在制作后期调整各个物体的性能参数、优化地图摆放布局等完善游戏细节的简单操作，都离不开对于引擎的基础使用。在帮助自己更深入了解开发流程的同时，学会使用引擎也有利于更好地和团队中其他成员的交流合作。

接下来的一周会给大家提供一套已经成型的Unity2D平台跳跃游戏素材，拥有一个包含移动、跳跃、三段攻击的可操作角色，两个类型的敌人，可设置的复活点、可收集的金币奖励物品、基础的UI显示，以及可编辑的配套地图场景素材



而这一周的目标就是利用这套素材，自行编排关卡地形、敌人位置、收集品位置等待，制作出一个完整流程3分钟左右的简单地图关卡，在此期间鼓励大家在群内随时交流进度、互相进行测试，最后会根据关卡流程的可玩性程度各自进行一个简单的打分。

当然，可能已经有不少同学在第一周或者更早之前就对unity有所了解和进行了相应的学习，甚至已经制作过自己的demo，有过参加Gamejam的合作经验，这部分同学可以借助自己的经验，在这套资源的基础上加入更多的关卡功能要素，如更多的角色可用动作、更多种类的敌人、功能性场景物体等等。

而才刚开始使用Unity引擎的同学也不用担心，素材内代码相关的部分已经提前写好，场景物体也已经挂载好了一套基础的组件和初始的变量值、并对每个功能和变量作用都进行了详细注释，项目保证了最基础的运行。因此更多的是会在可视化窗口内进行简单的编辑，考研关卡编辑能力。

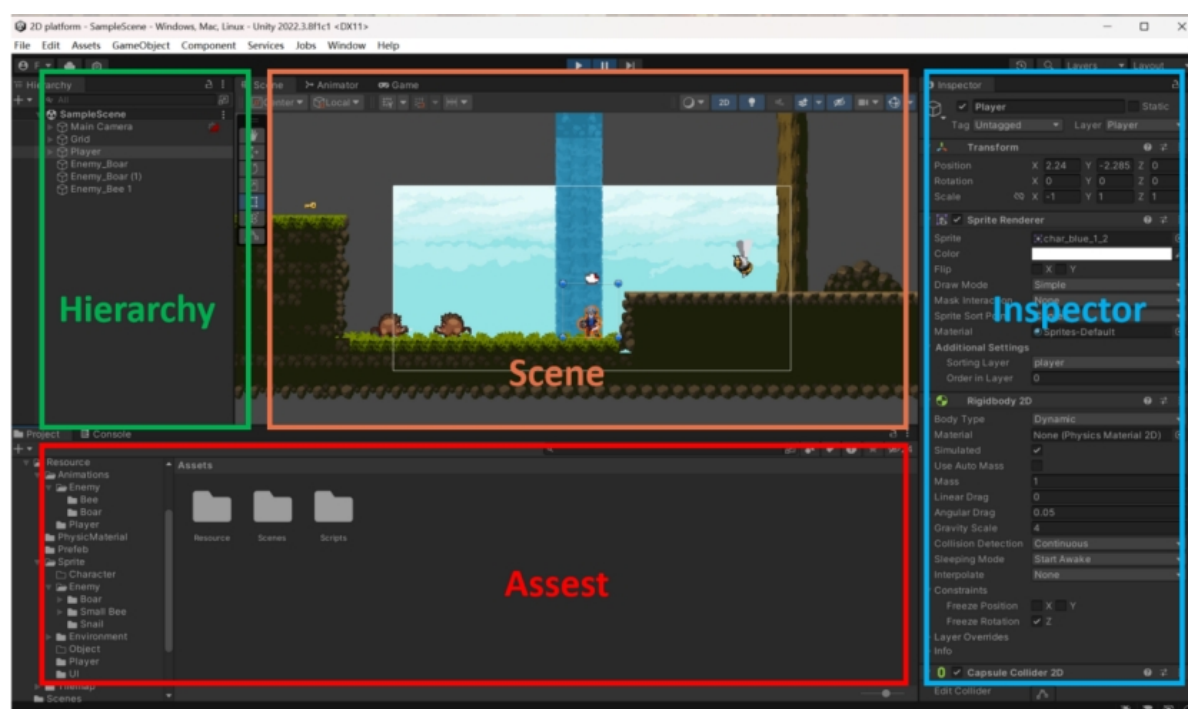
而课程也会从熟悉Unity的界面开始，到了解项目中的基础资源文件结构、形成对项目资源整体的认知；再到介绍已有场景中可以使用的各个物体，方便对变量进行详细调整；最后则是教学使用Unity内置的2D场景编辑器TileMap，正式开始编辑关卡。

想对Unity进行更进一步了解的同学也可以观看素材的原教学视频；这套项目素材是以麦蕊老师（B站ID：M\_Studio）的仍在更新的Unity入门课程《勇士传说》（<https://www.bilibili.com/video/BV1mL411o77x>）的前半部分为原型，在制作过程中遇到问题可以尝试跳转到教学视频中的对应部分或者直接咨询我（QQ：2431146252），由于本人并不是专业的程序，项目内许多功能的都只是为了便于理解的不规范写法，而一个功能的实现方式也会根据项目的实际情况进行调整，了解代码只是为了在遇到相应问题时能更便于与程序交流，实现仍是需要以程序为主导。

最后，本次的学习只是游戏开发的一个简单的小模型，旨在将游戏开发的过程更为具象化、调动各位参与开发的热情；不过由于个人原因写的时候时间比较仓促，后续内容省略的部分比较多。而即便是在训练营结束之后的时间里，无论是引擎功能的学习还是实际制作过程的关卡经验学习，都将是一个漫长的过程，未来更多还是需要依靠自己搜集资源、发现问题和解决问题，愿你我共勉。

## 2.Unity界面

在没有对Layout进行修改的情况下，Unity各界面的默认布局整体由四大块组成

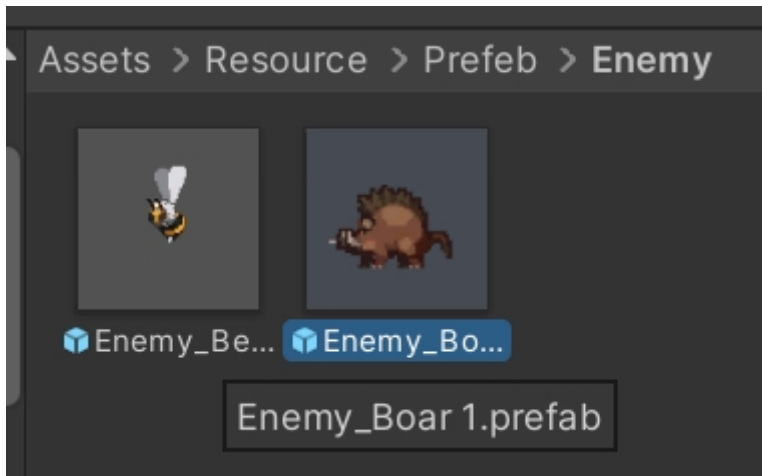


- **Scene**为游戏的场景，可以为各场景物体的摆放位置进行调整；
- **Hierarchy**为场景物体列表，显示场景内物体名称、层级关系、激活状态；
- **Inspector**为单个物体的详细面板，显示其身上所挂载的组件以及可以调整的各个参数；
- **Assest**为文件夹，显示项目内的文件存放位置和所属关系

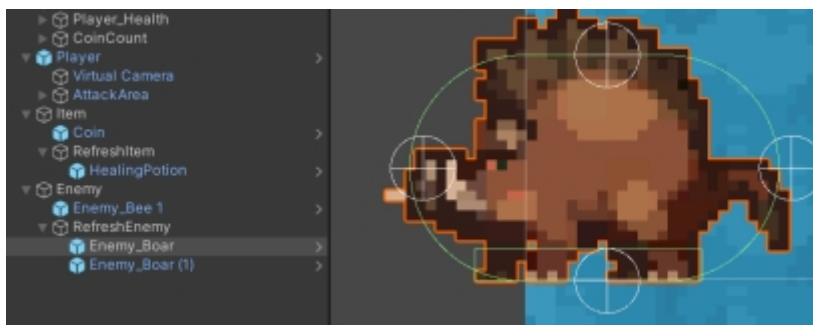
比方说我们现在需要在关卡内添加一个敌人，那么我们应该如何将其放置在对的位置，并设置好对应的参数呢



第一步：从Assest界面的Assest/Resource/Prefeb路径中找到敌人的预制体文件



第二步：按住鼠标左键将其选中并拖拽到Scene界面内，挂载到“Enemy”物体底下，再在场景内调整好摆放位置



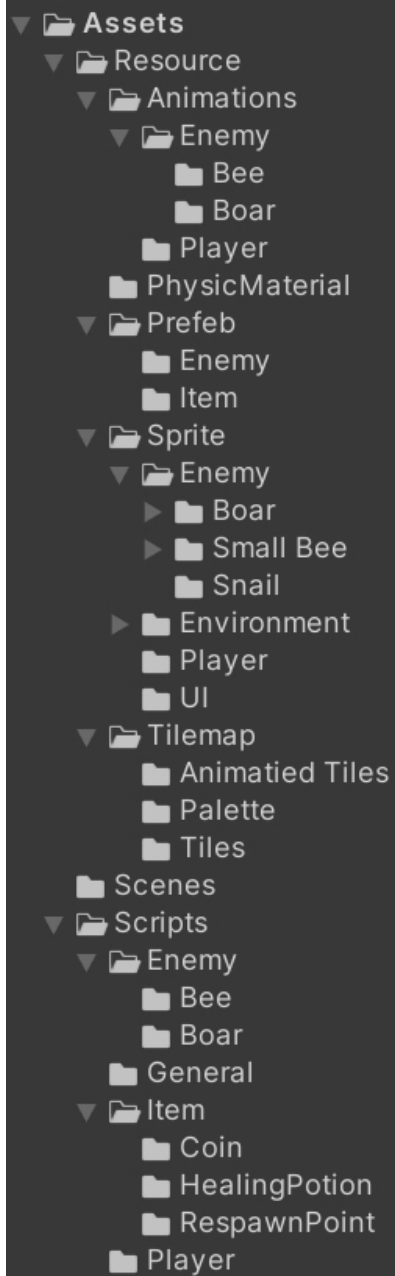
第三步：再在hierarchy界面中对其进行选中，修改其inspector面板下，“Enemy”组件的对应参数



### 3.文件夹结构

文件夹存储着项目所需要的所有资源，包括外部插件、运行环境、缓存文件等等等等....

不过大部分时候我们都只需要管理Assest文件夹底下由我们自行导入的项目实际制作所需要使用的素材资源，也就是游戏所使用的图片、动画、音效、代码等。



一般而言，最基础的便是资源（Resource）、代码（Scripts）场景（Scene）这三个组成部分



具体到这个项目，以玩家为例

角色首先需要有一个能够显示其在场景中位置的外在表现形象，这一部分需要图片；

而角色的移动又需要实时监测键盘输入的指令并执行相应的功能，这一部分需要代码；

在移动的过程中还要播放与默认状态下不同的动画以显示角色当前的运动状态，这一部分需要动画.....

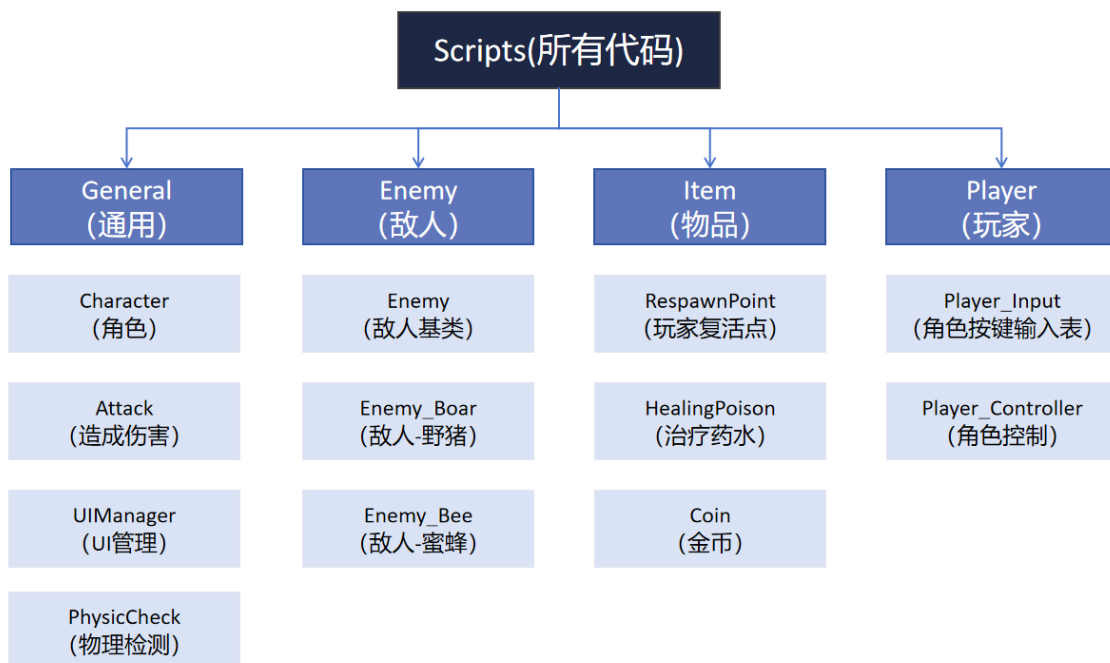
每一部分都环环相扣，且依赖于不同的素材资源。较为通用的做法便是按资源的类型进行分类，需要用到该类资源时再在内部查找对应的具体所属，是敌人/玩家/物体还是UI/场景等等....



(整个项目的完整资源结构大体如上)

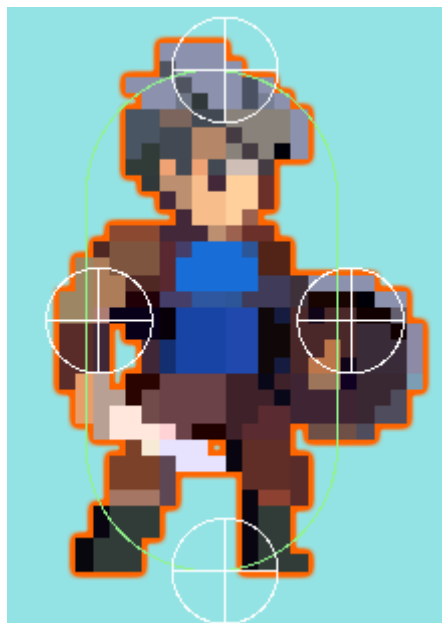
## 4.场景物体

在大概了解素材资源的结构后，接下来便是具体到每个物体的各自功能了，首先先对项目中所写的几个比较核心的代码有个简单的了解。



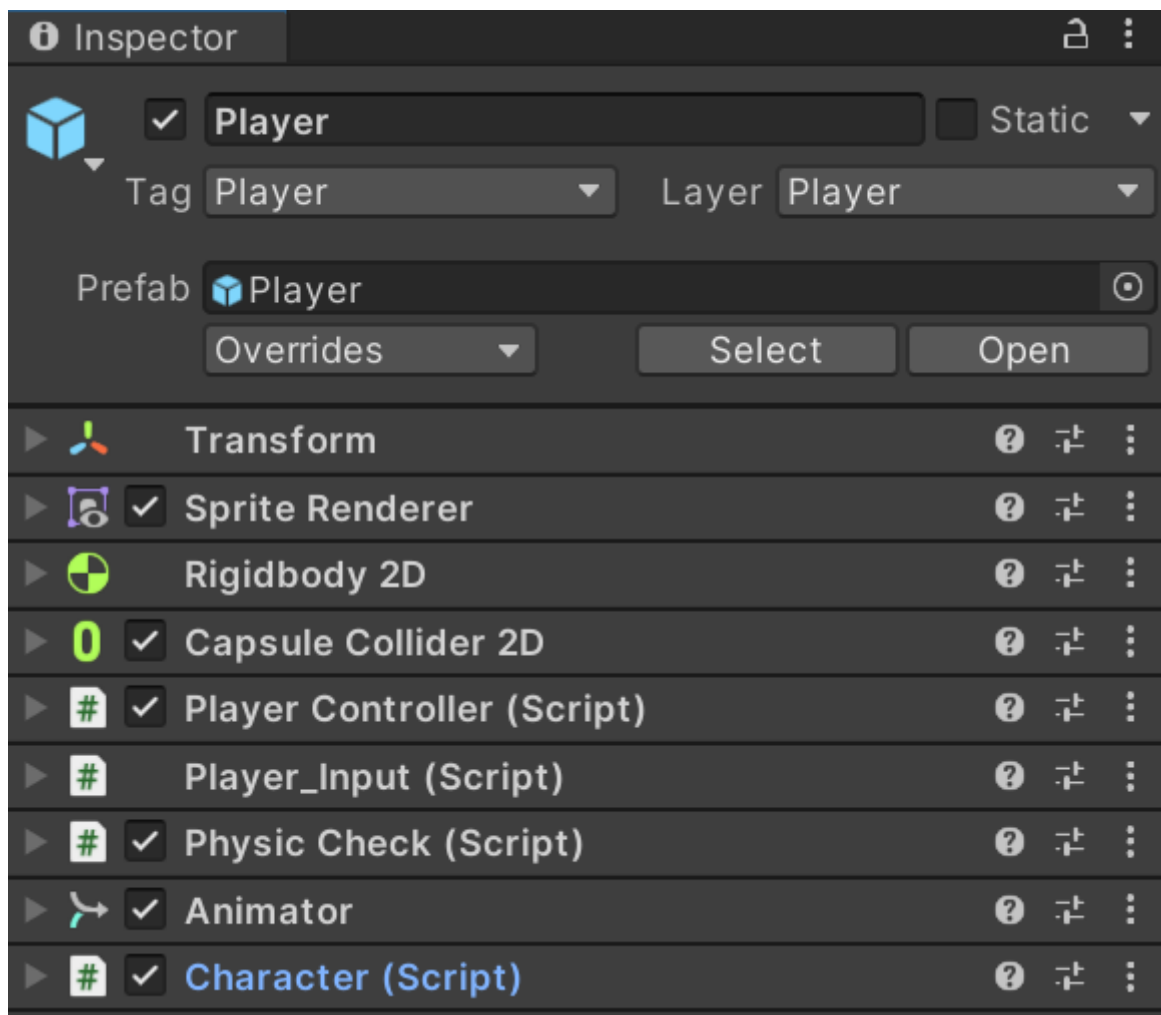
- Character是Enemy敌人和Player玩家都需要挂载的角色基础组件，承载着角色血量、受伤和死亡触发事件的功能；
- Attack是造成伤害的组件，敌人的碰撞伤害和角色的攻击伤害都来源于此
- UIManager是用于显示角色生命值和收集物品数量的UI显示组件
- EnemyGeneral是所有敌人的基础代码，承载敌人的移动、等待、死亡等事件

而当这些组件进行搭配组合，便形成了游戏场景中的具体物体



在具体的游戏场景中，玩具有以下几个通用的组件

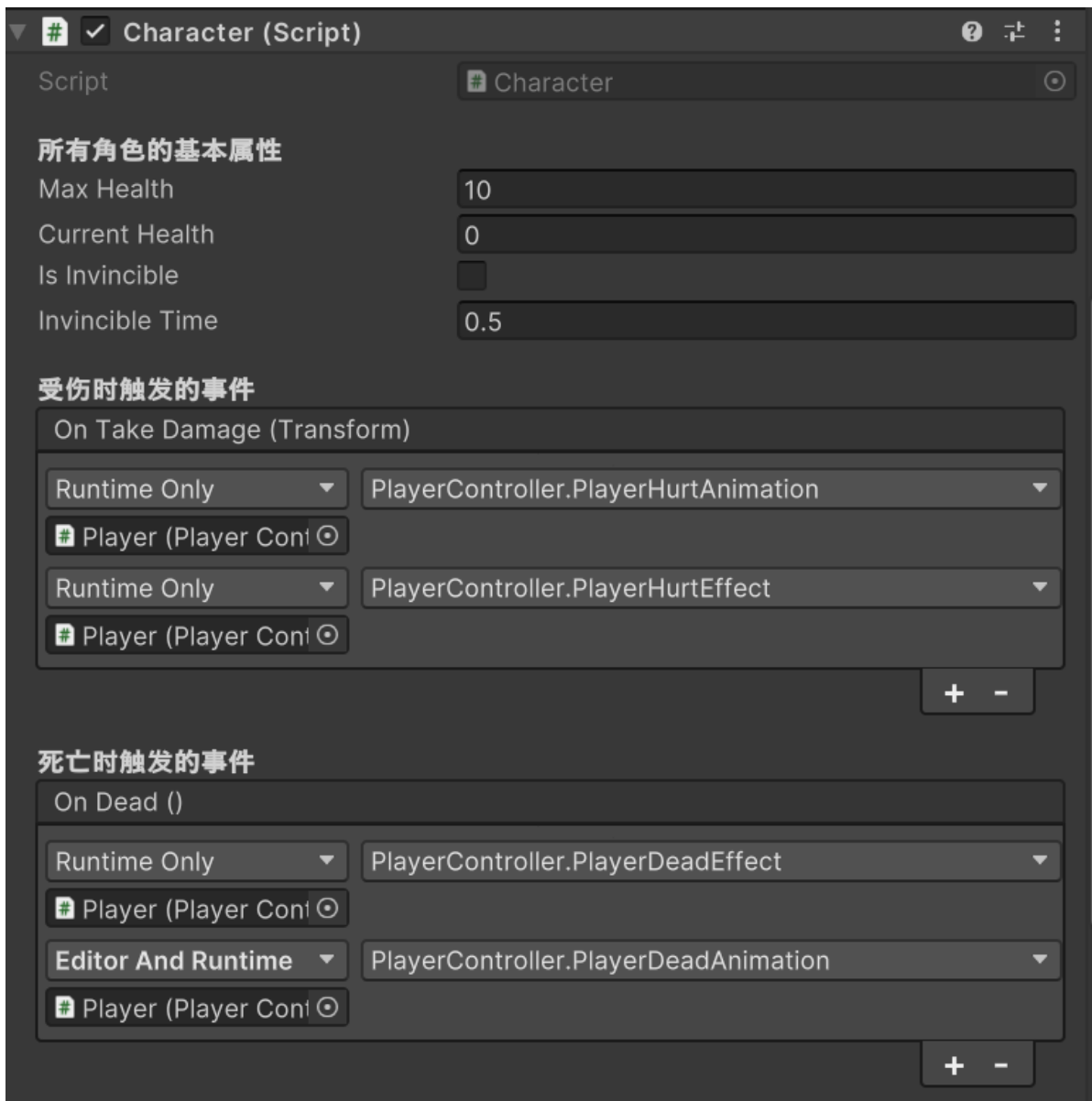
- Transform：玩家物体的坐标位置、缩放大小等
- SpriteRenderer：玩家角色图片的颜色、透明度等
- Rigidbody2D：玩家的物体的重力、摩擦力等
- CapsuleCollider2D：玩家物体的碰撞检测区域的形状和大小



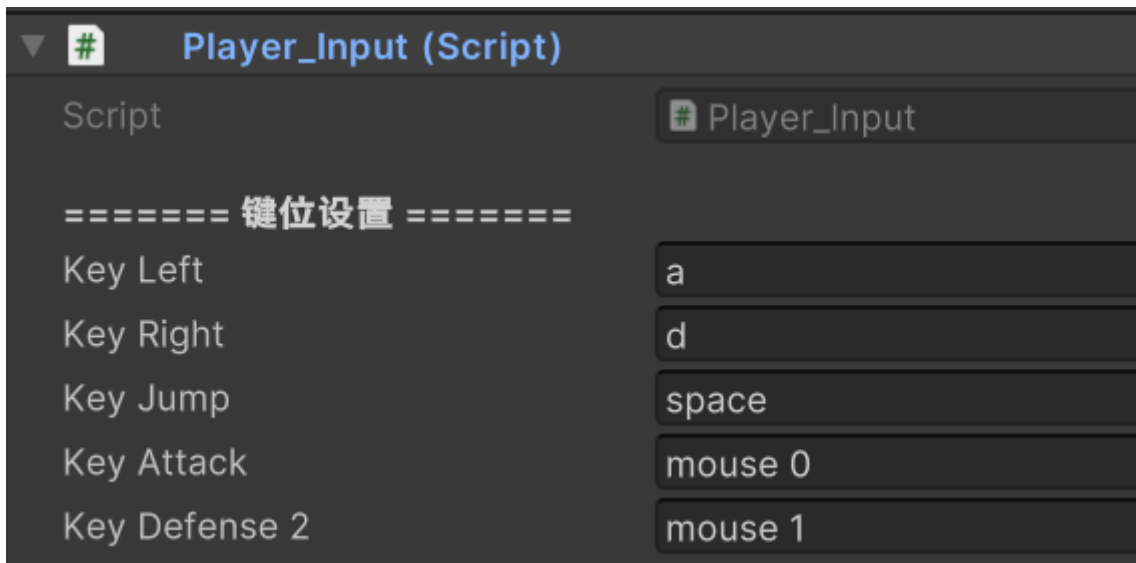
以及几个代码组件

- Character：角色通用类，包括角色的生命值、受伤无敌时间和受伤及死亡时触发的事件





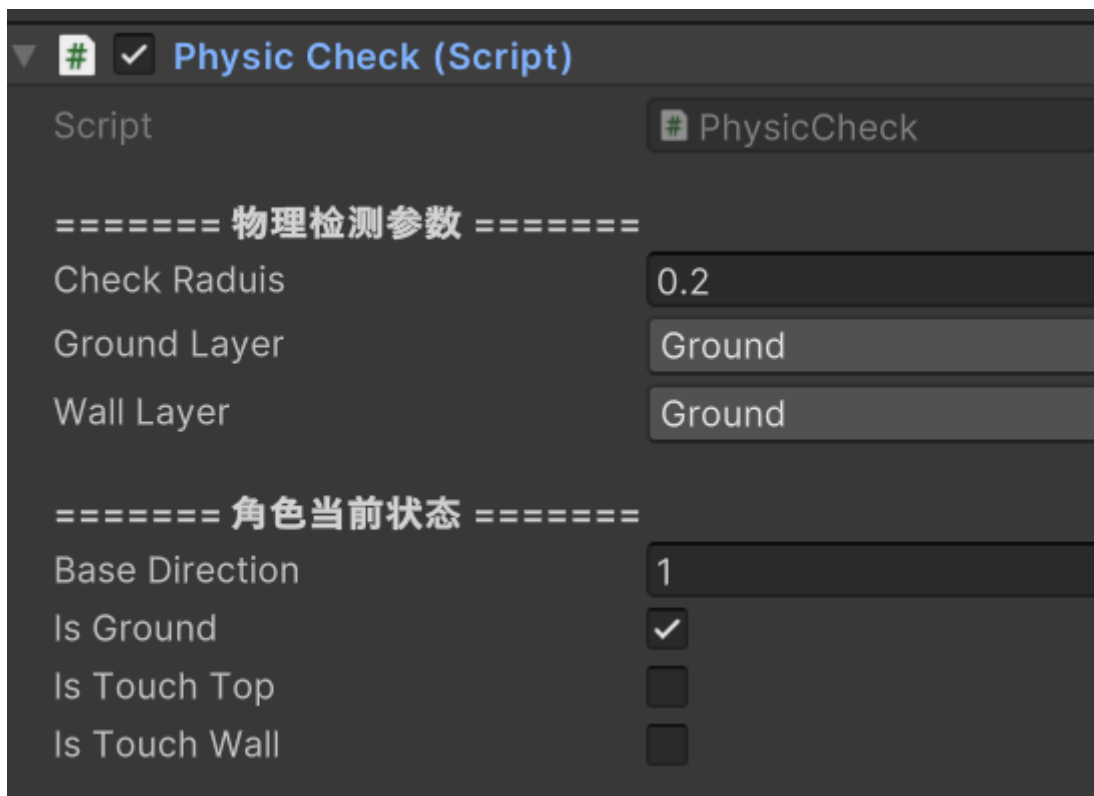
- Player\_Input: 键位设置，玩家左右移动、跳跃、攻击的对应键位



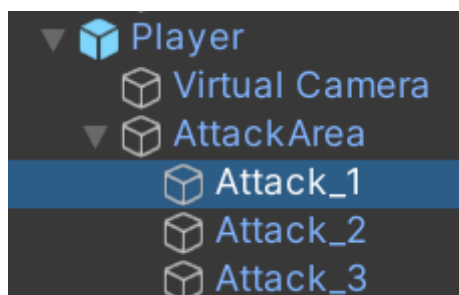
- Player\_Controller: 角色控制，包括角色的移动速度、跳跃力度、受伤可被击退的力度等



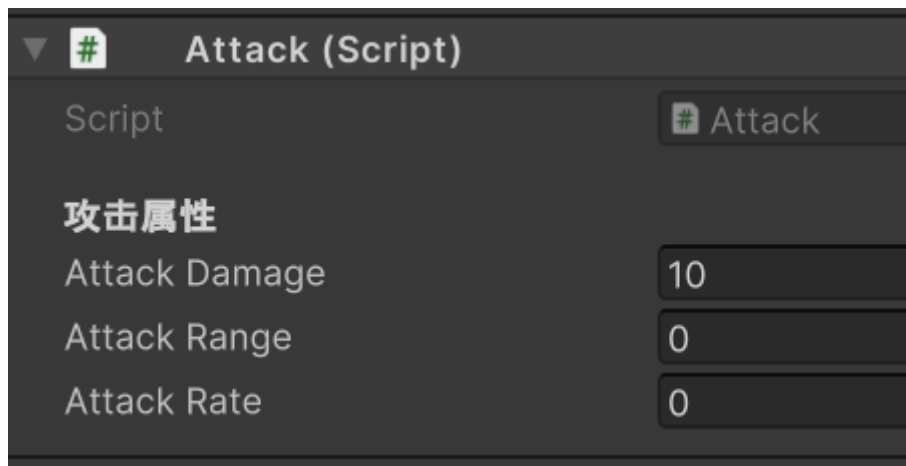
- PhysicCheck: 物理检测，包括检测的半径大小和检测图层以及是否有存在碰撞



以及挂载在“Player”底下“AttackArea”里的三段攻击



其中的Attack组件能够单独调整每一段攻击造成的伤害



因此想要调整角色的移动、攻击、生命值数值或者其他图片显示、手感之类的操作，都可以选中负责对应功能的组件进行参数调整。

在素材中，除了角色的基础移动以外，我也预设了两个敌人和三个物品，其中挂载的组件功能并不复杂且和玩家的有所重复，因此只简单介绍下他们的功能

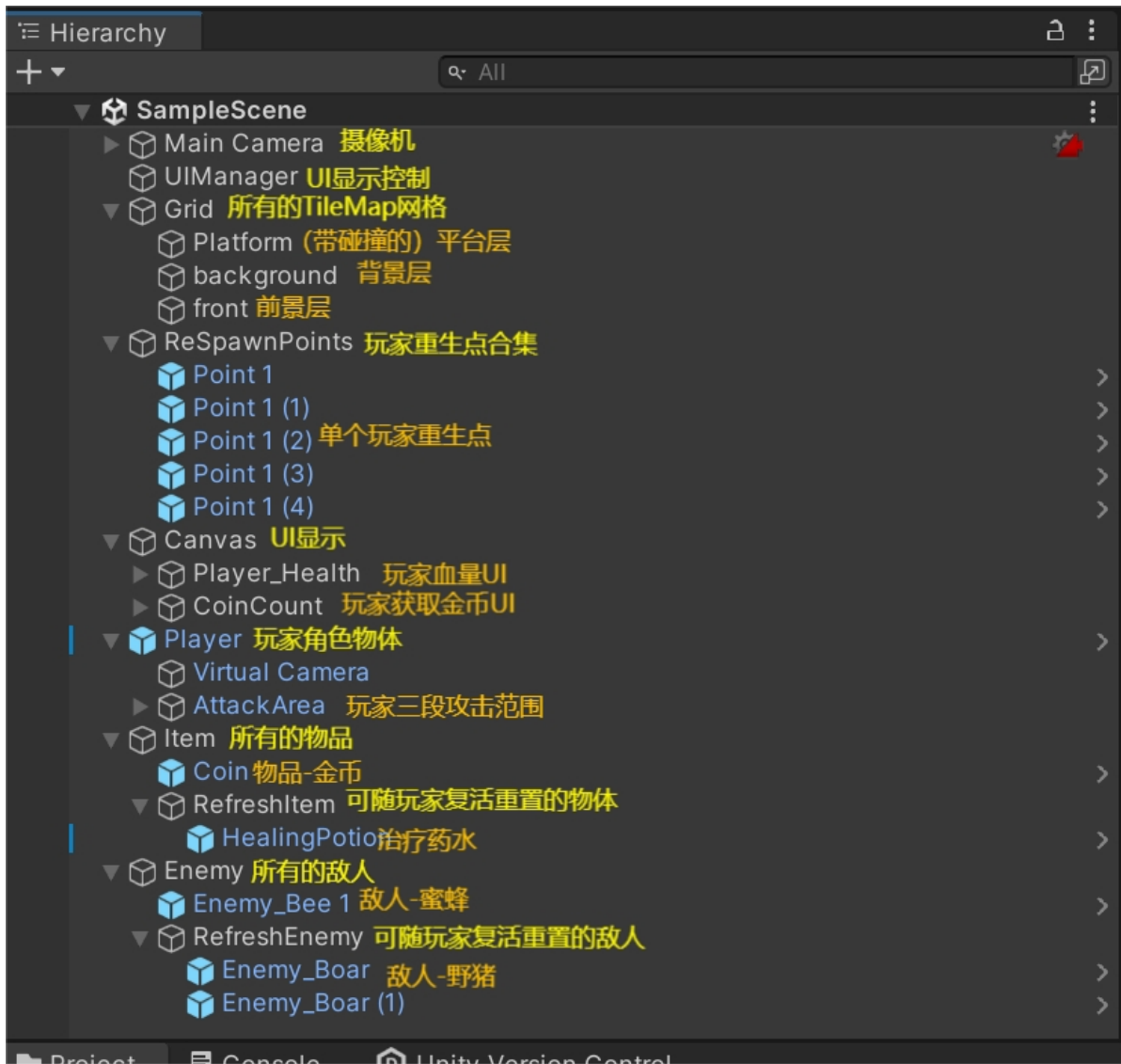


两个怪物：野猪和蜜蜂

- 野猪：默认水平左右移动；受到伤害时会朝受击来源的反方向进行一小段击退；接触到玩家时，玩家会受到伤害
- 蜜蜂：默认垂直上下移动，受到伤害时会朝受击来源的反方向进行一小段击退；接触到玩家时，玩家会受到伤害

三个物体：复活点、生命药水、金币

- 复活点：默认为蓝色，玩家最后接触到的复活点会变为绿色，且玩家死亡后会在改绿色复活点处复活
- 治疗药水：恢复玩家血量，血量满时不进行拾取
- 金币：增加显示在UI面板上的数量，无实际作用

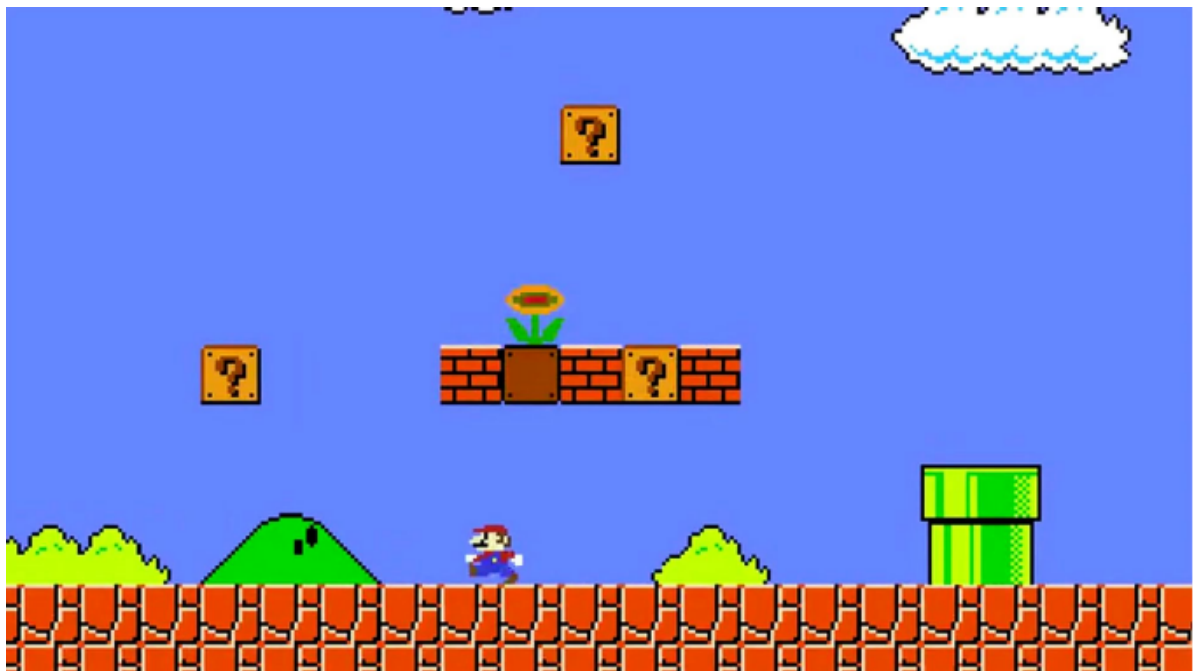
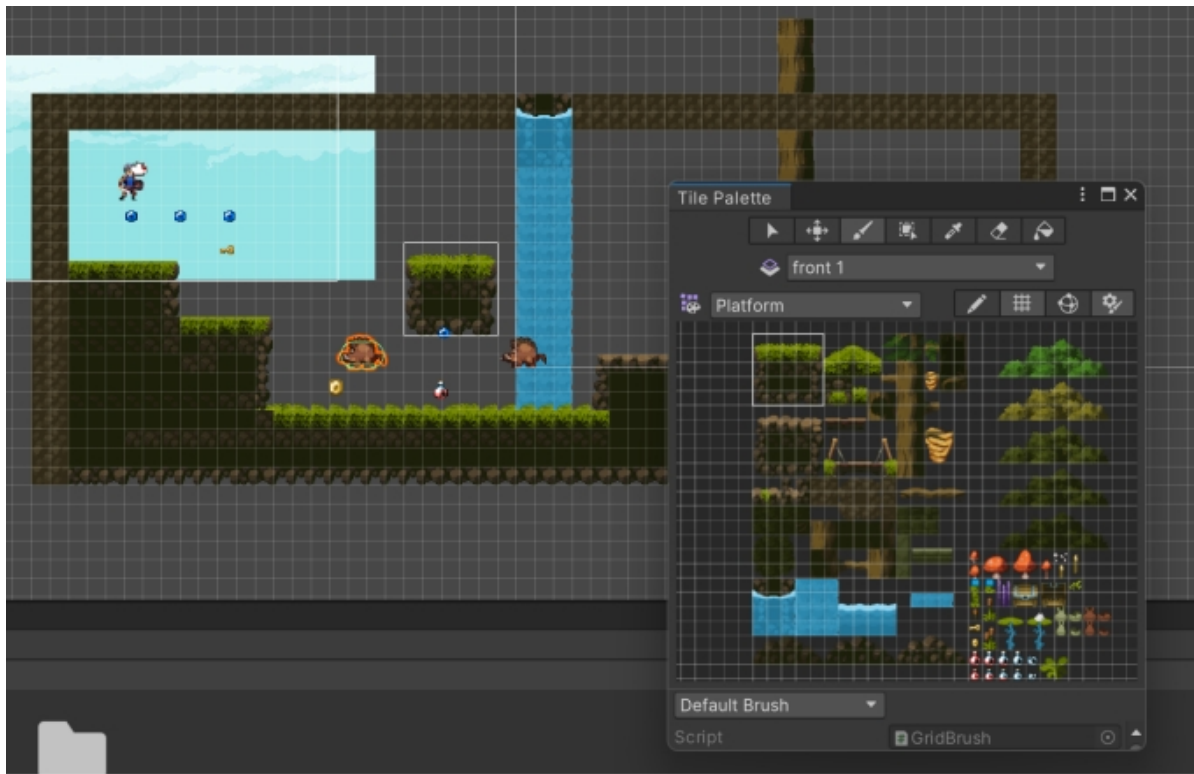


整个场景中的游戏物体大体如上，可以在制作时自行调整数量

需要注意的是，物品和敌人分为“可重置”和“不可重置两类”，需要随玩家复活后重置的物体要挂载在“Refresh”前缀的Item/Enemy下

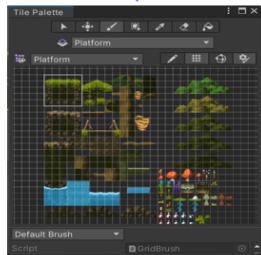
## 5.Tilemap（瓦片地图）

Unity内置的Tilemap是2D游戏中较为场景的快速地图编辑功能，正如名字“瓦片地图”，它可以允许我们像拼接地砖一样，将场景细化到以一个个格子单位来进行编辑排布



想想我们绘制绘画需要什么步骤呢？首先是要有一个能够承载颜料的调色板，然后再往上倾注颜料，再选取一张可以承载色彩的画布，最后再是开始用画笔作画

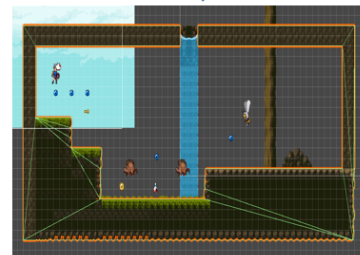
而在Tilemap编辑器中，TilePalette便充当了调色板的角色，其中的每一个Tiles瓦片便是不同的色彩，而场景中的Grid便是画纸



Tile Palette



Tiles

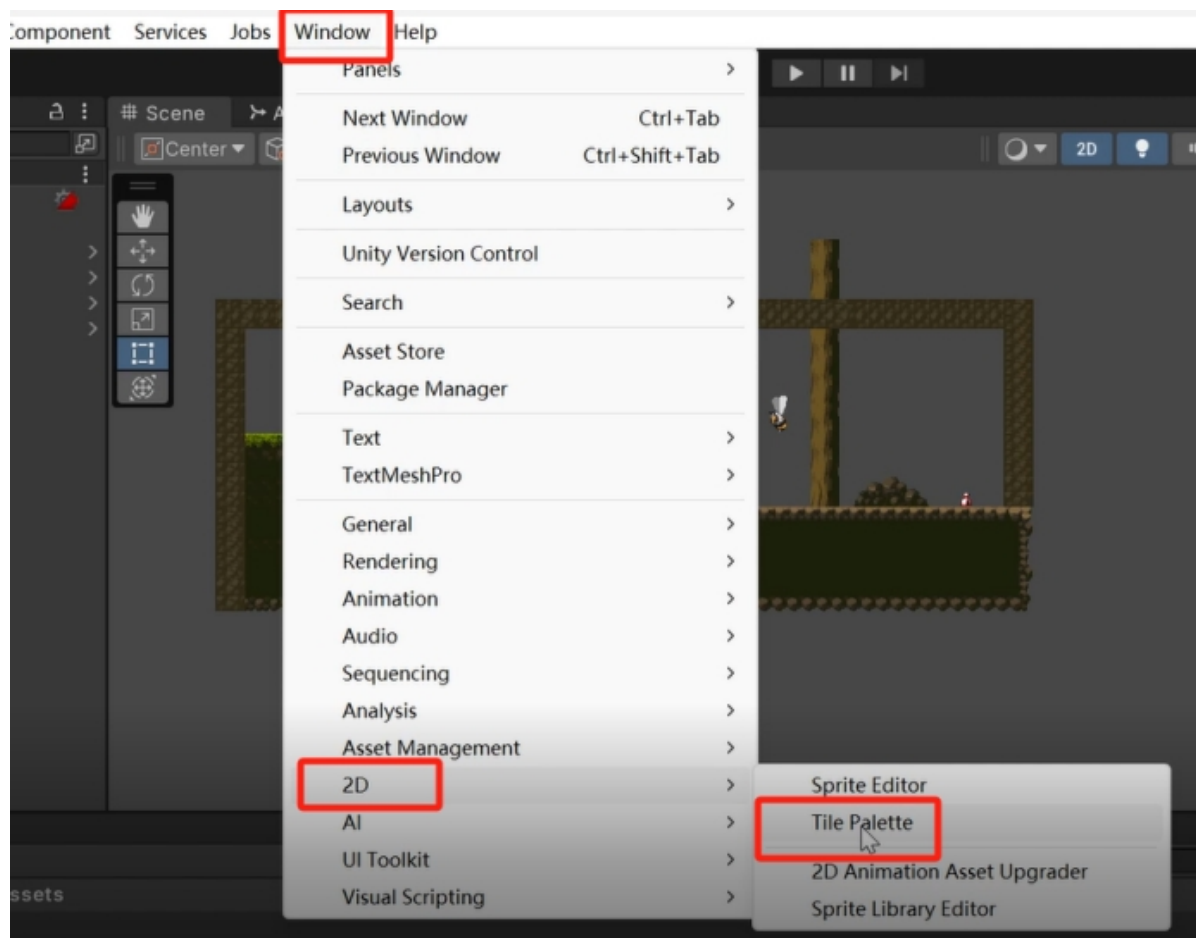


Grid

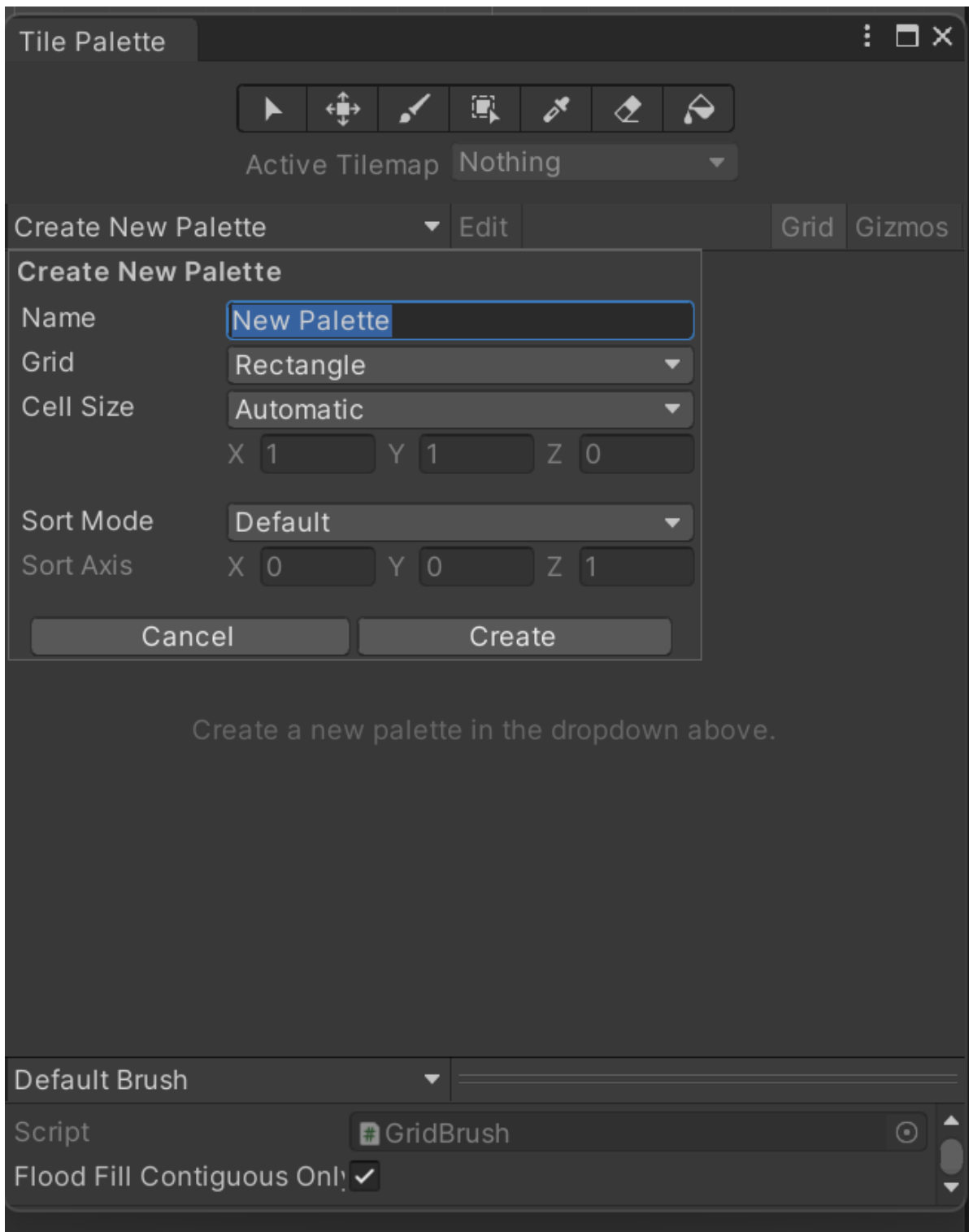
## 5.1 创建TilePalette

按照这个顺序，第一步便是创建自己的调色板

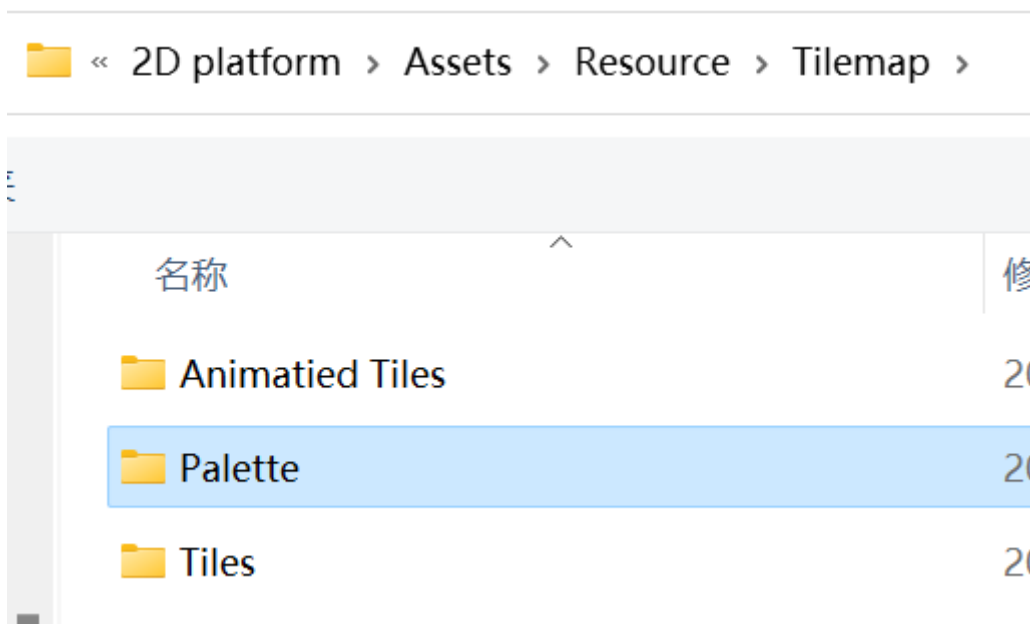
开启方式：Window-2D-TilePalette



开启后，对于新项目而言首先需要将TilePalette(调色板)在项目文件中进行创建，点击界面中的“Create New Palette”选项，填入名字后点击右下角的Create

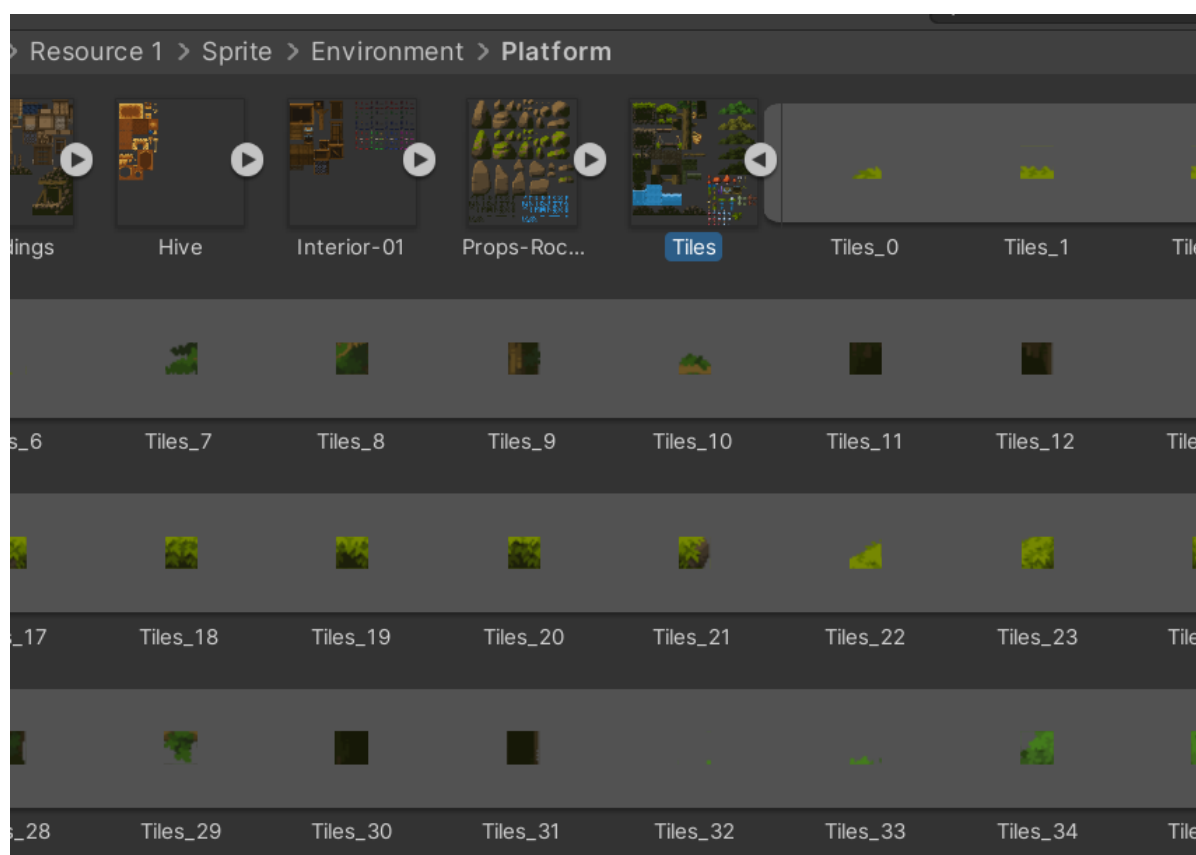


选中存放该TilePalette文件的文件夹位置，一个调色板便创建完成了



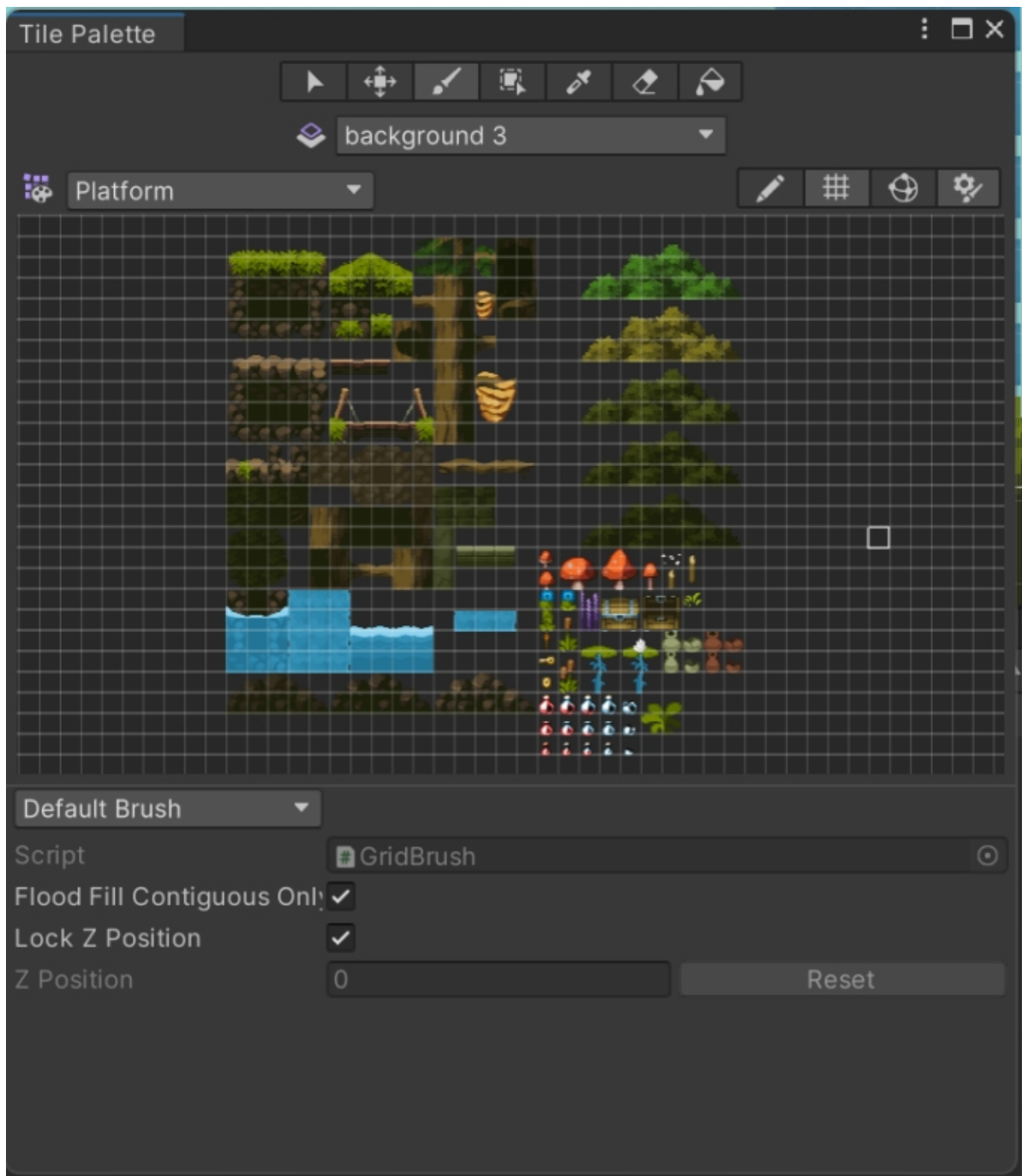
## 5.2 创建Tiles

当然，目前这个调色板中还是空白的，接下来的第二步便是往其中填入Tiles（颜料），方法也很简单，便是在对应的地图素材文件夹中，寻找到已经切割处理好的图片素材并将其拖拽到Tile Palette之中



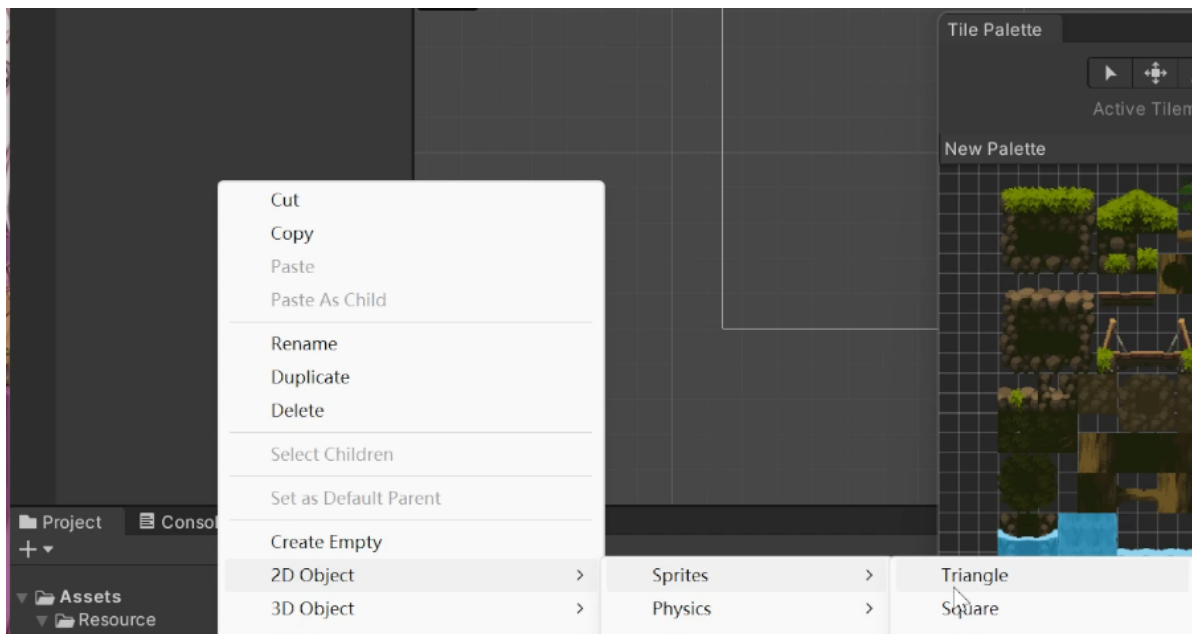
（与创建Tile Palette文件一样，Tiles也需要在对应位置创建一个新文件夹用于存放每个Tile）



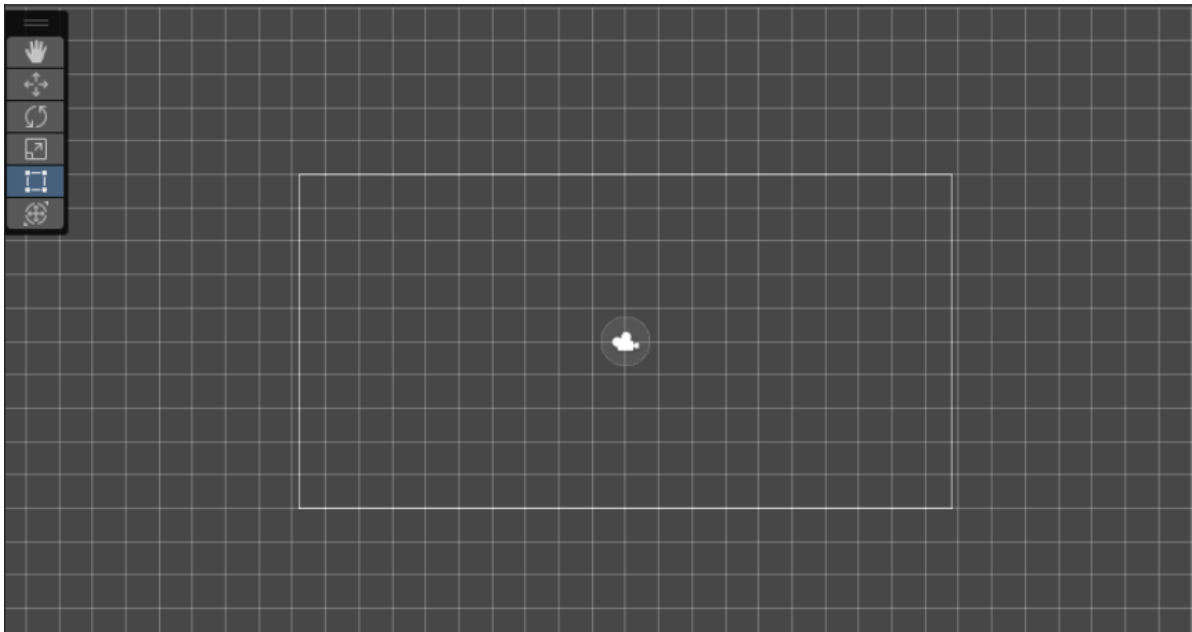


## 5.3 创建Grid

最后一步便是在Hierarchy界面右键，选择2D Object-Sprites-Triangle创建画布Grid

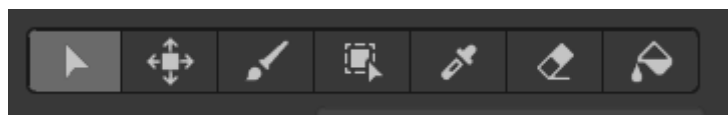


此时就可以看到地图中的一个网格，并在TilePalette中选择Tile，进行编辑了



(尽管以上步骤在项目素材中都已经完成，在从零开始制作项目的Tilemap时仍需要这样一步步创建)

## 5.4 Palette界面的基础功能



从左到右分别是：选中、拖拽、涂改、范围填充、取色、填充

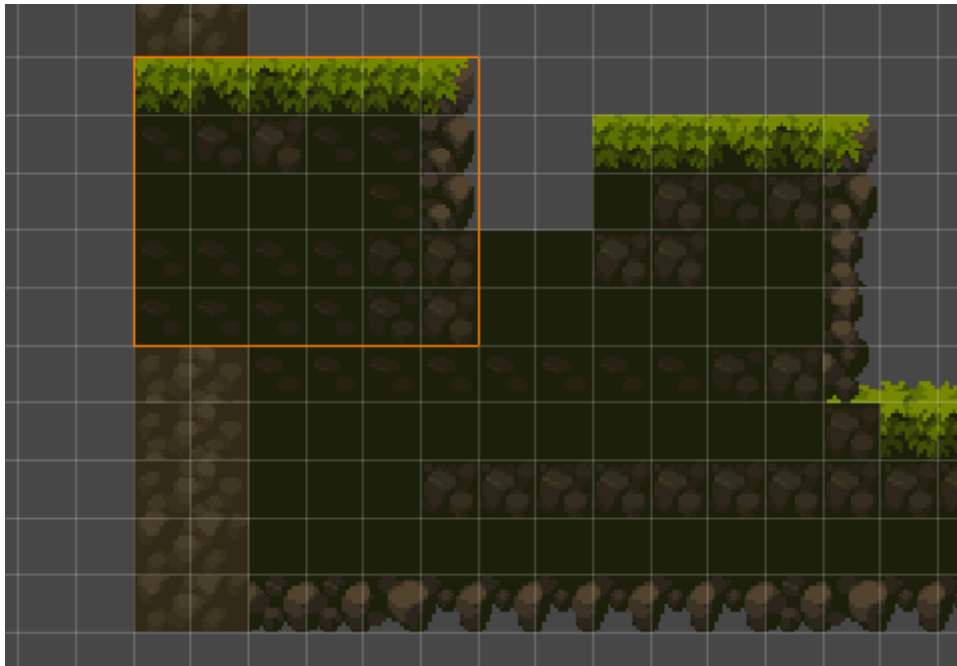
- 选中（快捷键：S）

通过按住拖拽的方式，选中一片已经绘制矩形区域



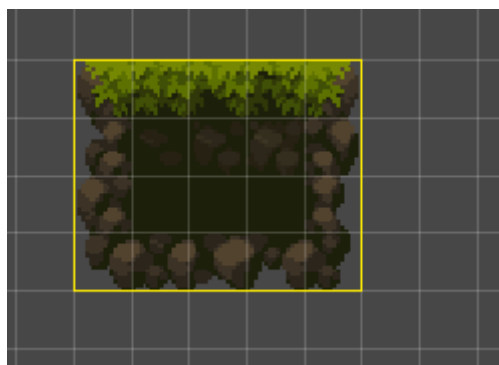
- 拖拽（快捷键：M）

对已经选中的区域进行整体移动



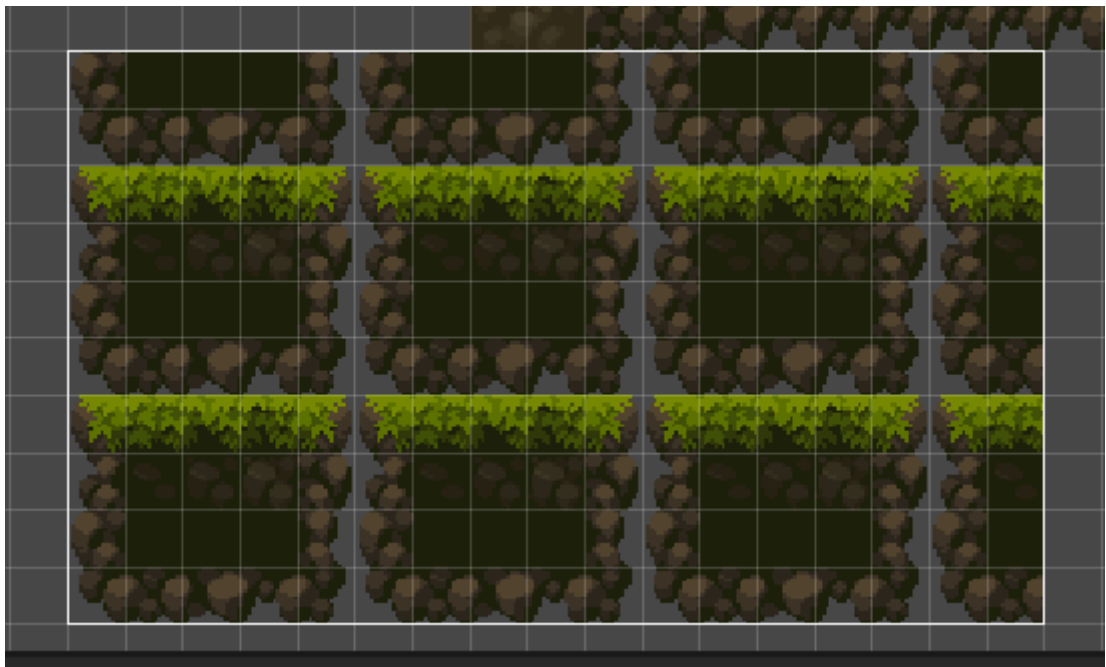
- 涂写（快捷键：B）

使用Palette中选中的Tile，进行涂写



范围填充（快捷键：U）

在Grid中选区域后，将区域全部以Palette中选中的Tile进行填充



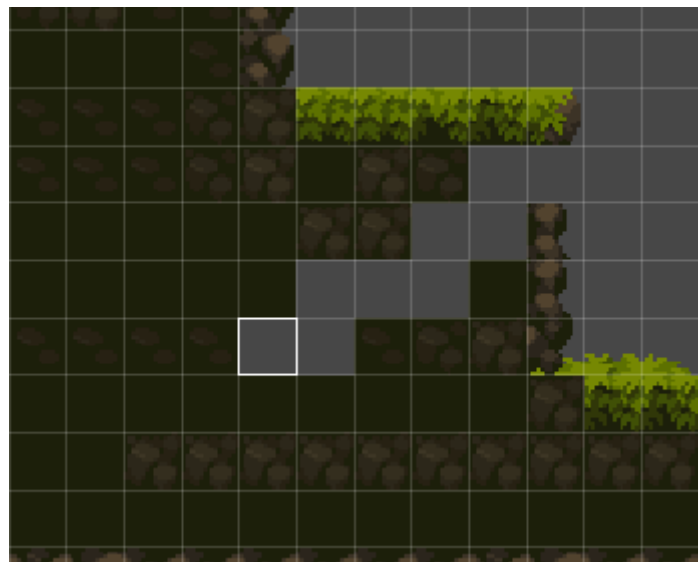
- 取色 (快捷键: I)

选取Grid中已绘制的单个Tile，快速在palette中将对应的tile进行选中



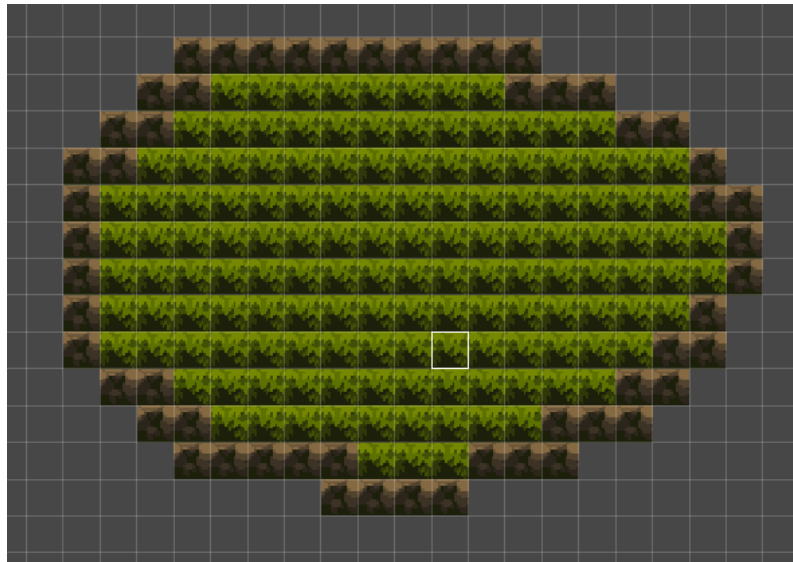
- 擦除 (快捷键: D)

对场景中已绘制部分进行擦除

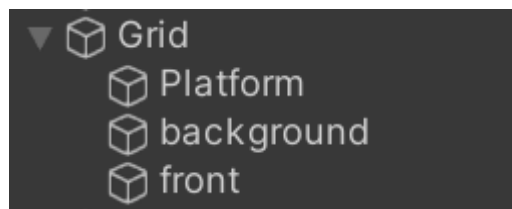


- 填充（快捷键：G）

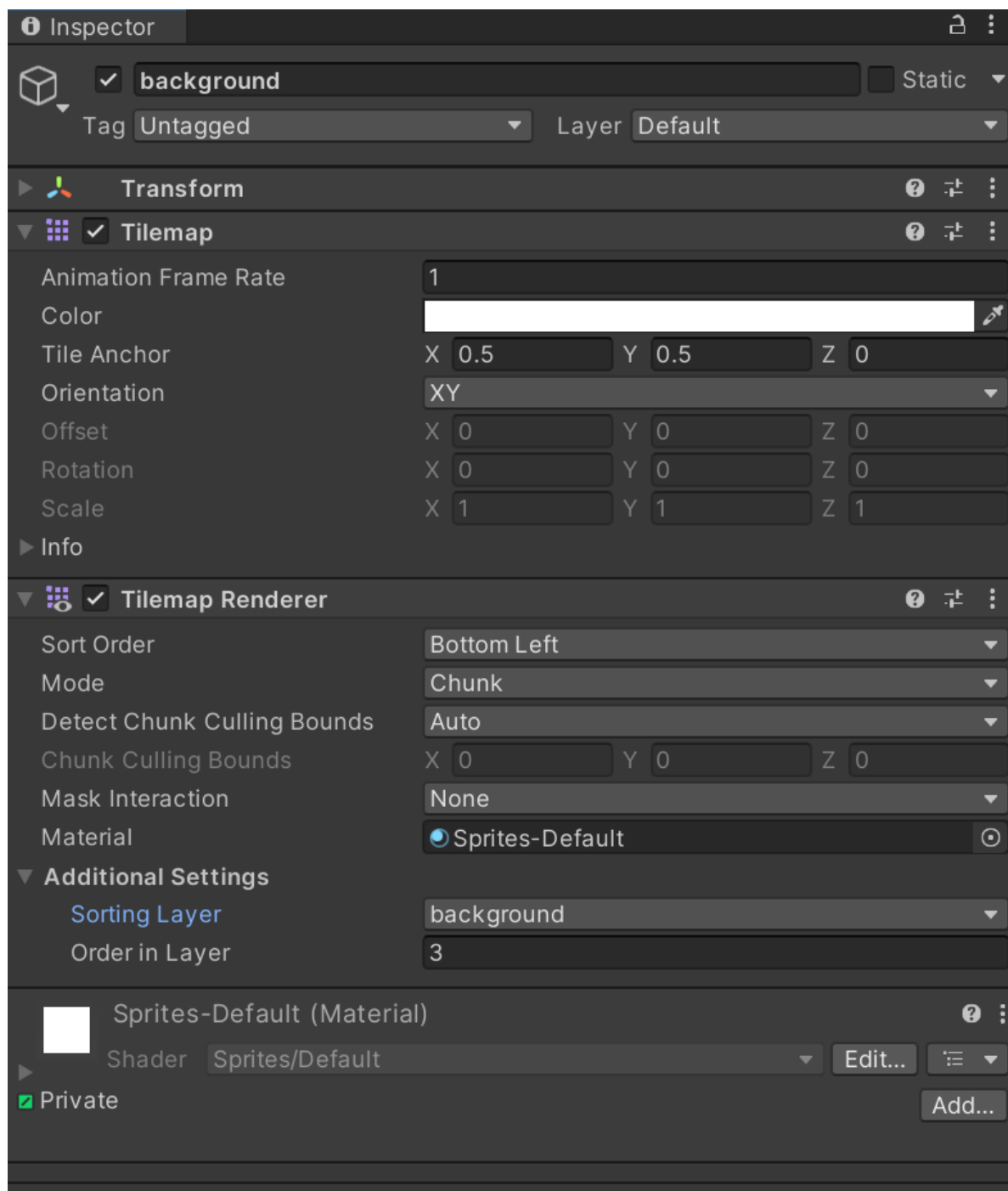
将一片密闭空间内的空白部分快速用选中的Tile进行填满



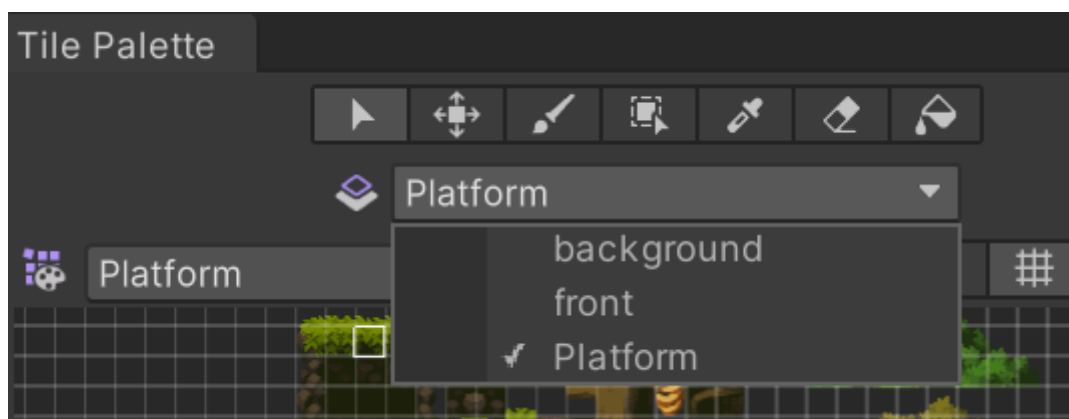
需注意的是，同一个Grid内可以存在多个不同层级的网格



每一层都可以单独挂载组件（调整碰撞体区域、显示顺序等）



因此，使用Palette对每一层内部的修改不会影响其他图层，如果在进行上述操作后发现没有对应产生效果，可以检查下是否是在同一图层内进行绘制



以上仅是Tilemap的一小部分常规应用，此外还有在场景中动态播放的AnimatedTile、绘制时能够按照一定规则自动排列的RuleTile.....详细的TileMap教程可以参考文章[https://blog.csdn.net/qq\\_36303853/article/details/132381914](https://blog.csdn.net/qq_36303853/article/details/132381914)