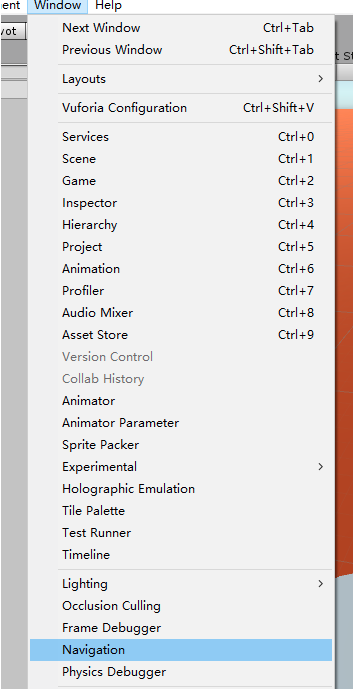
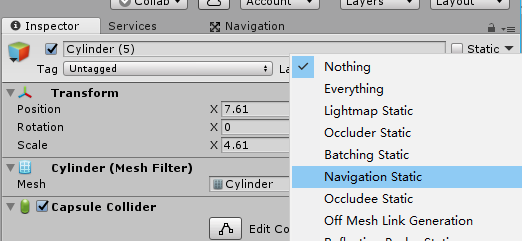
1. 在window中打开Navigation窗口准备烘焙地形



二、在inspector界面中将场景中的障碍物勾选为静态（ navigation static）



三、在Navigation窗口中进行导航网个的烘焙

Navigation中的主要参数

    //Areas（区域）：相当于障碍物的标签，可以在寻路物体身上的Navmeshagent组件中的AreaMask中屏蔽某一区域

    //Bake（烘焙）：生成指定的寻路网格

        //Agent Radius： 寻路物体的半径

        //Agent Heigh： 驯鹿物体的高度

        //Max Slope： 寻路物体可前进的最大坡度（最大为60度）

        //Step Heigh： 寻路物体前进可跨过的最大高度（最大不能超过寻路物体的身高）

        //Drop Heigh： 寻路物体下落的最大高度

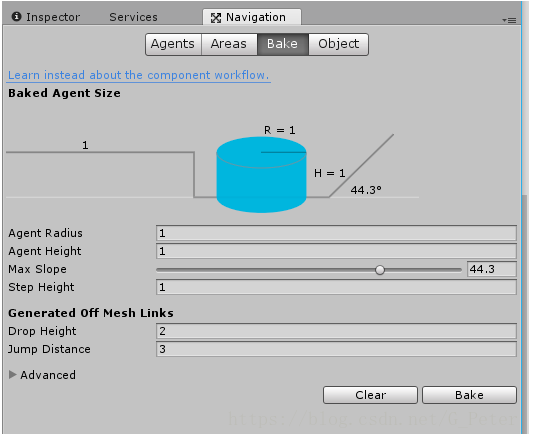
        //Jump Distance 驯鹿物体跳跃的最大距离

   //Object（对象，障碍物）：设置指定游戏物体的寻路烘焙属性

   //Navigation Static:设置当前游戏物体为寻路静态状态

   //Generate offMeshLinks：是否生成网格连接，如果勾选，网格之间会指定跳点烘焙

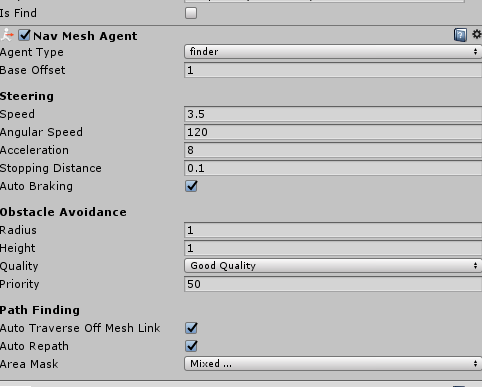
   //Navigation Area：设置当前游戏物体的导航区域



设置完成后，直接点击Bake场景中会生成导航网格

蓝色部分是导航代理可以行走的部分

四、生成导航网格后，在需要自动导航的物体上添加NavMeshAgent组件



其中几个主要参数

Agent Type ：导航代理的类型，这个参数下一节会用到

Base Offset：导航代理和当前游戏物体位置的差值

speed：移动速度

Angular Speed:转向速度

Acceleration：移动加速度

Radiuse： 导航代理的半径

Heigh：导航代理的高度

注意：需要导航是的游戏物体上不要同时出现刚体和NavmeshAgent两个组件，他们都会作用于物体的位置信息，可能会发生冲突导致导航失败

五、就接下来在代码中动态设置导航的终点

使用NavMeshAgent类中的函数SetDestination(Vector3 Target);函数来指定导航的终点就可以了