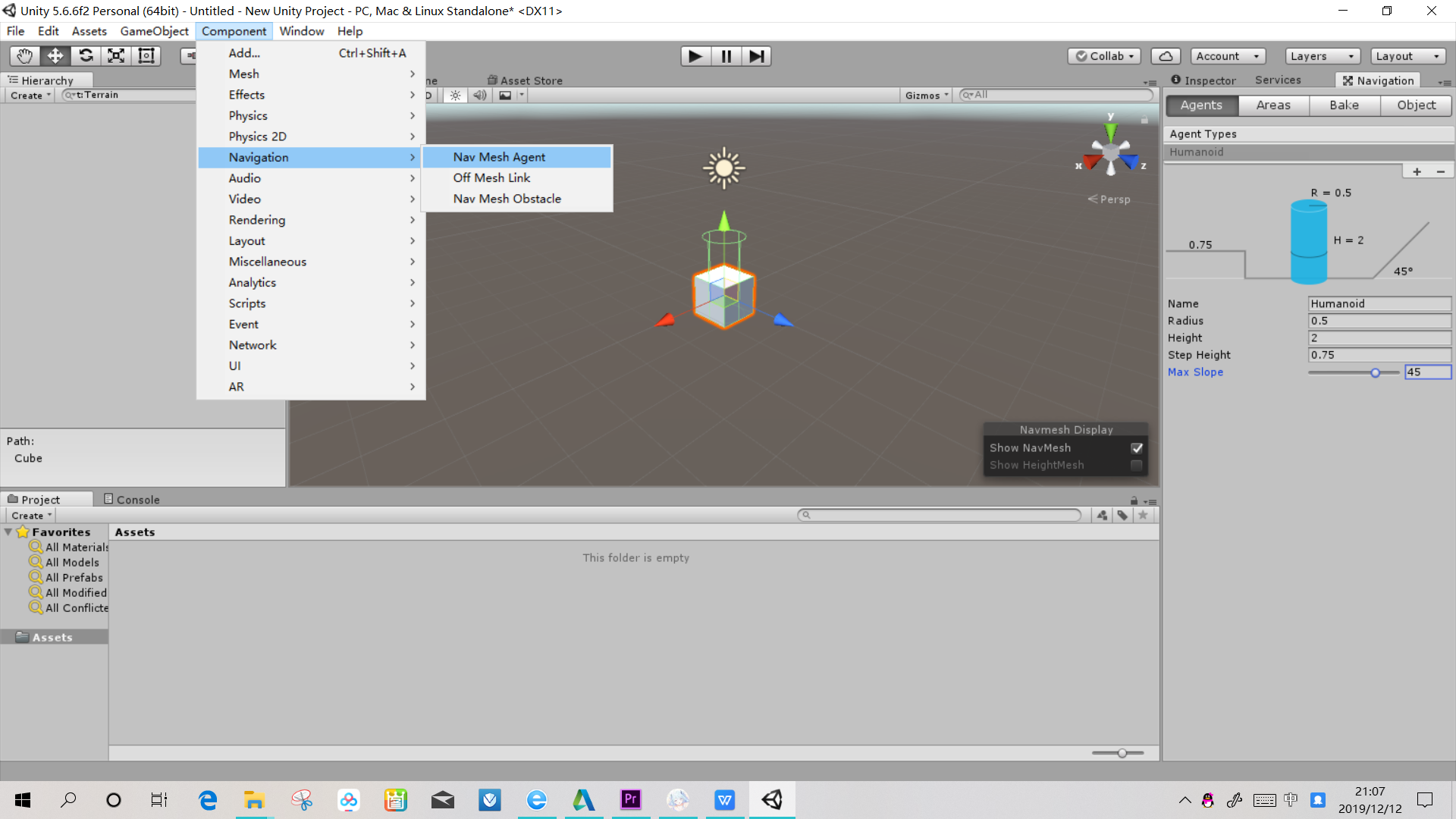
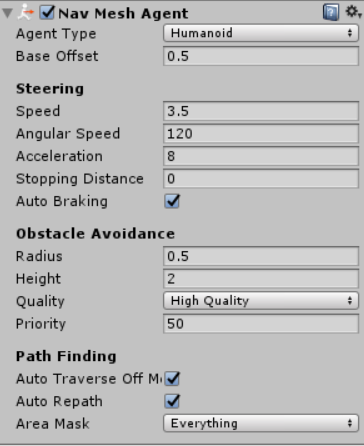
1. 简述NavMesh的设置方法。

答：在Windows的菜单栏下点击Navigation，使它出现在如下图所示的面板，然后在Component中找到Nav Mesh Agent。



1. 简述NavMeshAgent属性参数的使用方法。

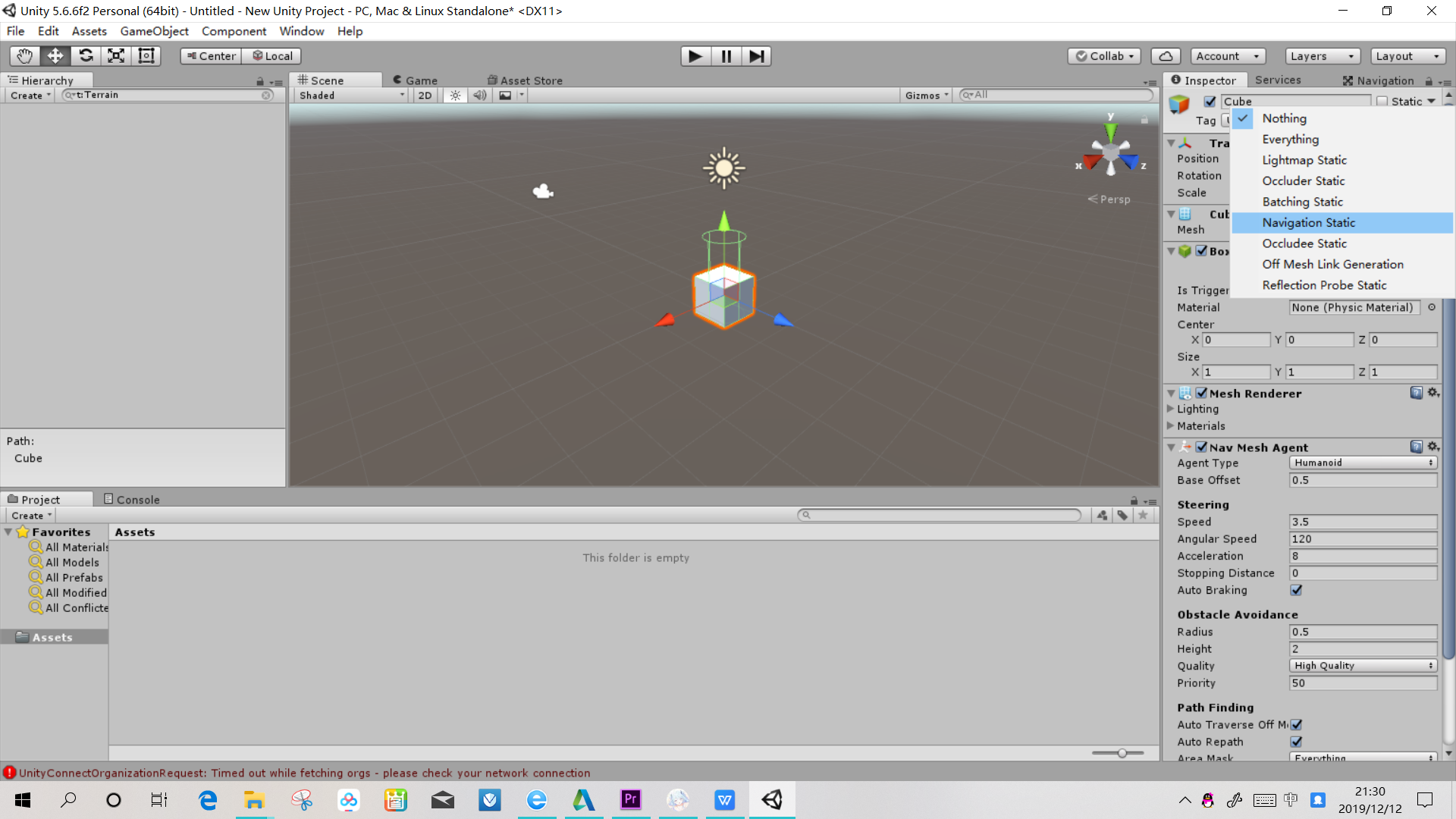
答：

（1）Radius 半径：代理的半径（仅用于寻路目的，可以跟实际对象的半径大小不一样，一般比实际对象的半径大）。

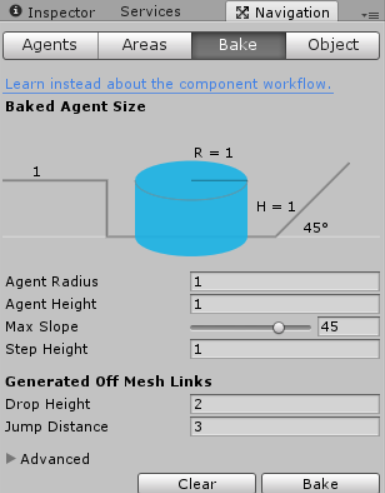
1. Speed 速度：代理可以周游世界，走向它的目的地的最大移动速度。
2. Acceleration 加速度：Maximum acceleration. 最大加速度。
3. Angular Speed 角速度：Maximum rotation speed in (deg/s). 最高转速（度/秒）。
4. Stopping distance制动距离：到目的地的距离小于这个值，代理减速。
5. Auto Traverse OffMesh Link自动遍历OffMesh链接：自动移动并关闭OffMeshLinks。
6. Auto Repath 自动重新寻路：如果现有的部分已失效，获得新的路径。
7. Height 高度：代理的高度（用于调试图形）。
8. Base offset基本偏移：碰撞几何体相对于实际几何体垂直的偏移。
9. Obstacle Avoidance Type障碍躲避类型：躲避的质量水平。
10. NavMesh Walkable导航网格行走：指定代理可以遍历的导航网格层类型。
11. 简述寻路过程中路网烘焙过程。

答：（1）在window中打开Navigation窗口准备烘焙地形。

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



1. 在Navigation窗口中进行导航网个的烘焙。



1. 点击Bake，场景中会生成导航网格。
2. 对于寻路过程中的障碍物绕行应该怎样处理?

答：（1）执行 File→Save Scene as 命令，将 Navigation 场景另存为 Obstacle 场景。

1. 执行 GameObject→3D Object→Cube 命令新建一个障碍物，将其放置在主角的前方，并赋予适当材质。
2. 执行 Component→Navigation→Nav Mesh Obstacle 命令添加 Nav Mesh Obstacle 组件。
3. 单击 Play 按钮进行测试，可以发现主角会绕过黑色立方体并到达终点。
4. NavMeshAgent组件的代理器移动到给定目标点需要利用哪个函数?该函数有几个参数?其含义分别是什么？

答：destination属性：可以设置目标点（SetDestination(target)方法相同效果）

nextPosition:获取或设置模拟navmesh代理的位置，可以手动的驱动角色进去移动（配合动画使用）

remainingDistance：物体当前位置与目标点的剩余距离。（设置了StopDistance后会出现一个问题，到时可以用这两个参数去调节）

UpdatePosition，UpdateRotation两个值都为bool值，可以通过更改他们俩，停止使用Agent组件来驱动物体移动和旋转。

常用方法：

isOnNavMesh： 判断物体是否在导航网格上。

isOnOffMeshLink： 判断物体是否在分离网格链接上。

SetDestination(Vector3 position)： 设置目标点。

Stop(): 停止寻路。

ResetPath（）：清空当前路径。

Resume（）： 停止游戏角色进行移动，并重新开始移动（原路径）。

Wrap（Vector3 position）：将物体瞬移到目标点。