**1.简述NavMesh的设置方法**

**答：**1）.需要设置Radius和Height（代理圆柱的半径和高度），还有下面的Stopping Distance,就是距离目标多远算导航结束。

2）在Windows菜单下找到Navigation窗口，点击Bake烘焙导航网格

**2.简述NavMeshAgent属性参数的使用方法**

**答：**

Areas（区域）：相当于障碍物的标签，可以在寻路物体身上的Navm eshagent组件中的Area Mask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 Height： 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 Height： 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 Height： 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 off MeshLinks：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

**3.简述寻路过程中路网烘焙过程。**

**答：**首先选择需要行走的路块进行烘焙（点击最上方的Windows下拉菜单的 Navigation），选择完成后，点击右下角的Bake，即可烘焙完成。在需要寻路的人物上，加上自动寻路Nav Mesh Agent组件，选中需要烘焙的路面以及物体，打上红勾，可以行走的区域选择Walkable，不可行走的选择Not Walkable。接下来再写个简单的小脚本，就可以让我们的人物动起来了。

**4.对于寻路过程中的障碍物绕行应该怎样处理？**

**答：**通过 Navmesh 插件实现主角遇到障碍物时自动绕行效果。执行 File→Save Scene as 命令，将 Navigation 场景另存为 Obstacle 场景新建一个障碍物，将其放置在主角的前方，执行 Component→Navigation→Nav Mesh Obstacle 命令添加 Nav Mesh Obstacle 组件，则主角就会绕过障碍物并到达终点

**5. Nav Mesh Agent组件的代理器移动到给定目标点需要利用哪几个函数？该函数有几个参数？其含义分别是什么？**

答：

NavMeshAgent函数

NavMeshAgent.Move 移动

NavMeshAgent.nextPosition 下个位置

NavMeshAgent.speed 速度

NavMeshAgent.isOnOffMeshLink 是否在OffMeshLink上

NavMeshAgent.nextOffMeshLinkData 下一个OffMeshLink数据

NavMeshAgent.obstacleAvoidanceType 障碍逃避类型

NavMeshAgent.height 高度

NavMeshAgent.FindClosestEdge 寻找最近边缘

NavMeshAgent.GetAreaCost 获取区域成本

NavMeshAgent.hasPath 有路径

NavMeshAgent.height 高度

NavMeshAgent.isOnNavMesh 是否在导航网格上