1. 简述NavMesh的设置方法

（1）Nav Mesh Agent是基本的寻路组件，将场景中需要寻路的的物体先设置为static，然后对这些物体进行烘焙（选中需要烘焙的物体，点击Windows中的Navigation，然后再点击Bake就可以了），将Nav Mesh Agent添加到需要导航的物体上，通过在脚本中获取到这个组件并调用SetDestination（目标位置）方法就可以了。

（2）Off Mesh Link组件，分离网格链接组件，有时，场景中的静态几何体的导航网格不是全部相连在一起的，从而使代理不能从世界场景的这一部分移动到另外的部分。它的意思就是蓝色是烘焙后MPC的网格路线，它都是由多边形组成的，如果在两个不相邻的物体之间进行移动，就需要这个组件（例如图中的标有箭头之间的两个物体）

（3）Nav Mesh Obstacle组件，是导航网格障碍物，在导航网格上的固定障碍物，可以作为烘焙过程中的一部分设置。但是，也可能在场景具有动态阻挡物，这将通过Agent移动来避开。这种动态障碍物，可以指定Navmesh Obstacle组件设置，可以被添加任何游戏物体上并随该对象移动。

2、简述NavMeshAgent属性的参数的使用方法

（1）updateposition是控制navmeshagent所挂游戏物体的一个位置信息，当为false时，物体的寻路位置信息将失去作用。

（2）nextPosition指的的navmeshagent的位置信息。

3、简述寻路过程中路由的烘焙过程

（1）在window中打开Navigation窗口准备烘焙地形

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

（3）在Navigation窗口中进行导航网个的烘焙

（4）生成导航网格后，在需要自动导航的物体上添加NavMeshAgent组件

（5）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

4、对于寻路过程中障碍物绕行应该怎么处理

（1）重新计算路径

（2）路径剪接

（3）监事地图的改变

（4）预测障碍物移动

5、NavMeshAent组件的代理器移动到给定目标点需要利用那个函数？

Transform.Translate函数。   
（1）position：世界坐标中所在的位置。

（2）rotation：旋转的角度。例如X：90， 意味着在X轴旋转90°。采用左手定律，大拇指朝向X轴，四指指向的方向为旋转方向。

（3）scale：缩放比例。默认打开为1