1. 简述NavMesh的设置方法。

答：对场景中的物体进行标记，然后进行路径烘焙，烘培前，在Window🡪Navigation🡪Bake选项卡下设置Agent参数，一般需要设置Agent的半径、高度、可以行走的最大斜坡角度、可以越过的高度。产生网格数据为要进行寻路的物体添加寻路组件（NavMeshAgent），通过NavMeshAgent组件的属性或方法进行移动。

1. 简述NavMeshAgent属性参数的使用方法。

答：NavMeshAgent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，将NavMeshAgent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。

1. 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；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，设置完成后，直接点击Bake场景中会生成导航网格；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)，在代码中动态设置导航的终点。

1. 对于寻路过程中的障碍物绕行应该怎样处理？

答：首先要获取运动数据并进行处理，然后进行路径烘焙，而Unity 官方内置的寻路插件 Navmesh 完美地解决了这个问题，通过NavMeshAgent组件的属性或方法进行移动来躲避障碍物。

5、 NavMeshAgent组件的代理器移动到给定目标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函数：可以设置目标点（SetDestination(target)方法相同效果）。

①类型：Vector3设置或者获得代理的目的点，世界坐标系。

②获得：返回设置给代理的目的点。

•如果一个目的点被设置了并且寻路过程还没结束，将会返回一个有效的，离预先设置的位置(position)最近的navmesh位置。

•如果代理没有路径或被请求的路径，返回代理自己的位置。

•如果代理没有映射到navmesh，返回infinity。

③设置：请求代理移动到离目的点最近的有效的位置。

• 路径结果可能是无效的直到几帧之后。使用pathPending获取结果。

• 如果找不到一个有效的最近的位置，将不会请求路径。使用SetDestination并且检查返回值，如果你需要明确的掌握这些情况。