一. 简述NavMesh的设置方法。

1. 选择场景中需要生成寻路的几何体-可行走表面和障碍物。

2. 在NavMesh面板中选择需要烘焙寻路的物体，检测是否勾选Navigation Static.

3. 根据你的agent大小来调整bake 面板的设置。

4. 点击bake按钮烘焙NavMesh。

二. 简述NavMeshAgent属性参教的使用方法。

NavMeshAgent属性:

Radius：导航代理的半径，我们可以适当的调节一下这个值  
Speed ：这个属性代表这个导航网格代理寻路时可以达到的最大速率

Acceleration ：加速度，表示代理的速度从0加速到Speed时的最大的加速度  
Angular Speed ：最高的角速度  
Stopping distance ： 制动距离，当代理据目的地的距离小于这个值时开始减速  
Auto Traverse OffMesh Link ：自动移动并关闭OffMeshLinks，这个选项对于我们利用程序来操纵后面我要介绍的OffMeshLink很关键，  
Auto Repath   自动重新寻路，如果发现现有路径已失效，那么它将获得新的路径，这个选项我们一般将其勾选上  
Height  ： 导航代理的高度。  
Base Offset ： 基本偏移，我们可以通过调整这个变量来调整代理自身的包围盒  
Obstacle Avoidace Type  ： 代理躲避的水平，一般我们选默认的High Quality就行了  
NavMesh Walkable ：导航网格代理可以通过的网格层类型

三.简述寻路过程中路网烘培过程。

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、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.

四. 对于寻路过程中的障碍物绕行应该怎样处理?

1.执行 File→Save Scene as 命令，将 Navigation 场景另存为 Obstacle 场景。

2.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

3.执行 Component→Navigation→Nav Mesh Obstacle 命令添加 Nav Mesh Obstacle 组件。

4.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

五. 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？

NavMeshAgent.Move 移动

NavMeshAgent.acceleration 加速度

NavMeshAgent.angularSpeed 角速度

NavMeshAgent.areaMask 区域遮挡

NavMeshAgent.autoBraking 自动制动

NavMeshAgent.autoRepath 自动重新获取路径

NavMeshAgent.desiredVelocity 需求速度

NavMeshAgent.destination 目的地

NavMeshAgent.FindClosestEdge 寻找最近边缘

NavMeshAgent.hasPath 有路径

NavMeshAgent.height 高度

NavMeshAgent.isOnNavMesh 是否在导航网格上

NavMeshAgent.steeringTarget 转向目标

NavMeshAgent.hasPath 有路径

NavMeshAgent.obstacleAvoidanceType 障碍逃避类