1. 简述NavMesh的设置方法。

新建一个Cube为目标位置，起名TargetCube。然后创建一个capsule(胶囊)运动体，为该胶囊挂在一个Nav Mesh Agent(Component - Navigation - Nav Mesh Agent)；最后写一个脚本就可以实现自动寻路了。

1. 简述NavMeshAgent属性参数的使用方法。

Agent Size

Radius：半径。

Height：高度。

Base Offset：偏移值。

Steering（操控行为（详情见Game AI篇，AI底层））

Speed：在NavMesh上移动的最大速度

Angular Speed：角速度

Acceleration： 物体从静止到最大速度的加速度

Stopping Distance：停止距离（常与remainingDistance一起使用）

Auto Braking：是否自动减速。（不勾选的话如果速度太快，物体会冲过去目标点以一定的惯性返回目的地）

Obstacle Avoidance(避障)

Quality（质量）（枚举）

None ： 无

Low Quality：低质量

Medium Quality：中等质量

Good Quality： 高质量

High Quality： 高质量

Path Finding（路径跟随）

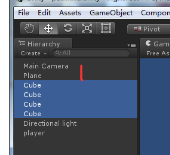
Auto Traverse Off Mesh Linked：

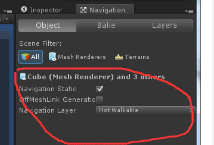
Auto Repath

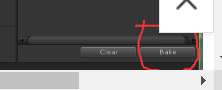
Area Mask

1. 简述寻路过程中路网烘焙过程。

设置Cube为不可通过物体，首先我们点击Window - Navigation 窗口，然后选中4个Cube，按照下图设置这4个Cube为不可通过，然后烘培。

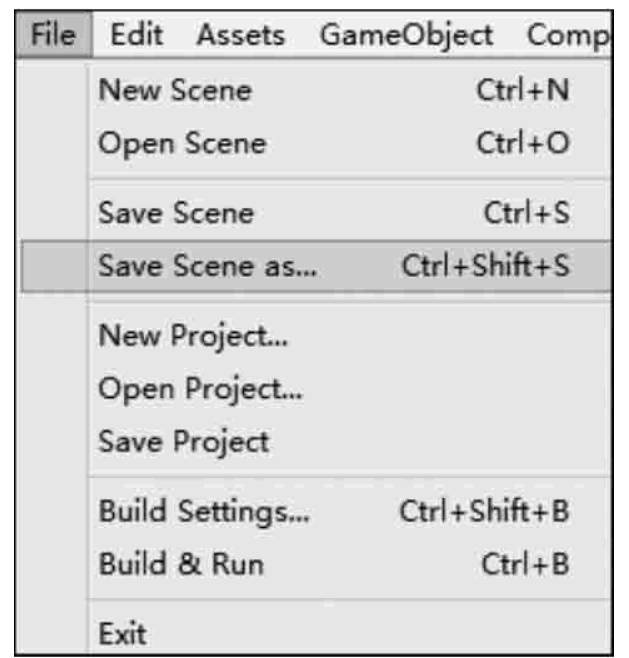




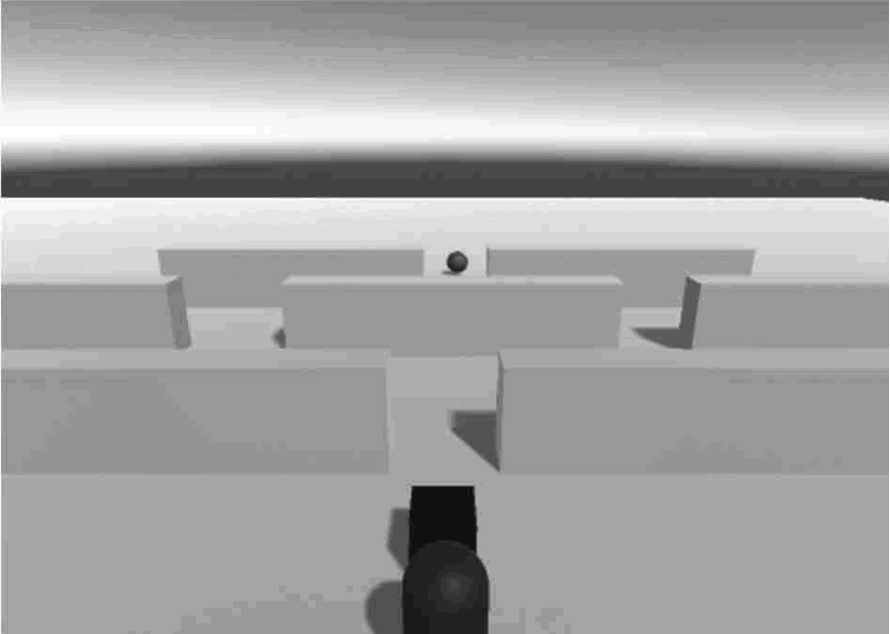


1. 对于寻路过程中障碍物绕行应该怎样处理。

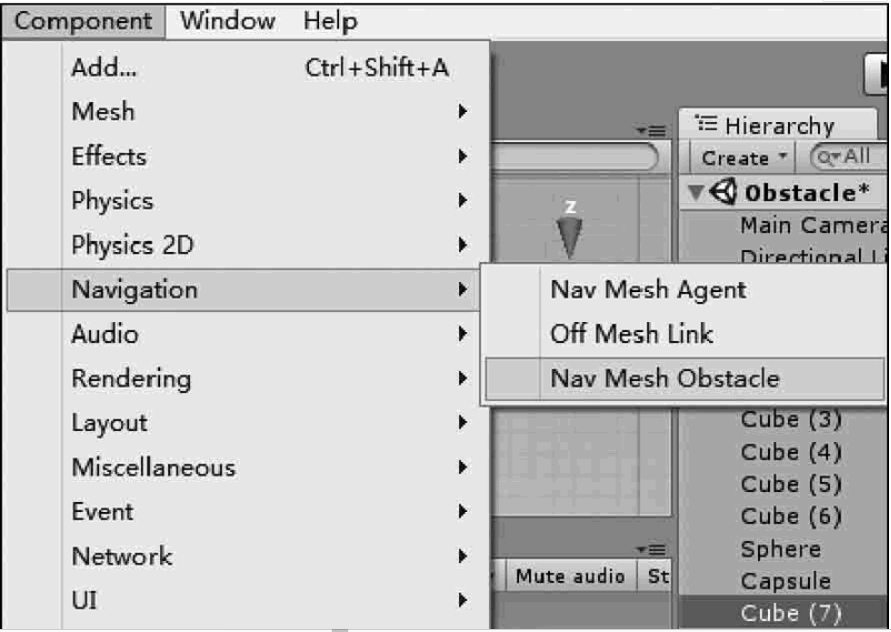
步骤 1)：执行 File→Save Scene as 命令，将 Navigation 场景另存为 Obstacle 场景，如下图所示。



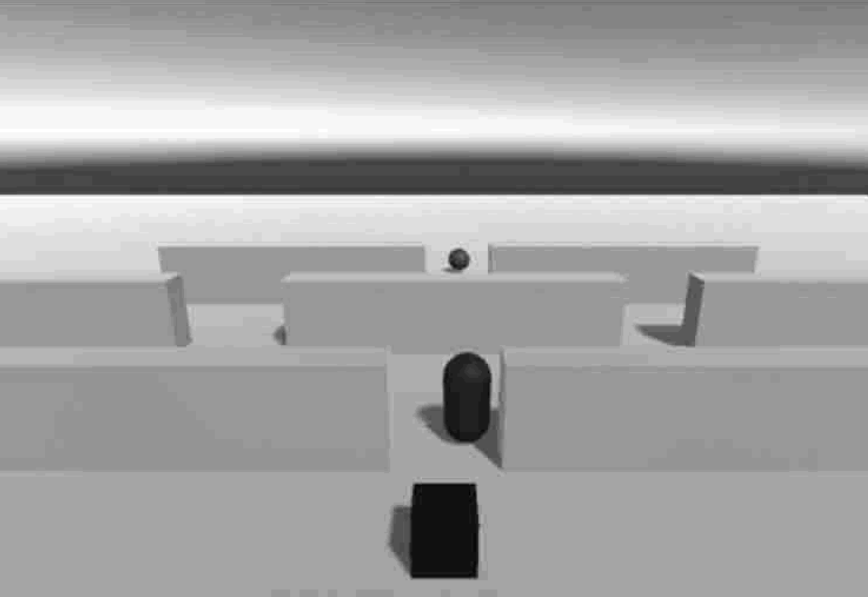
步骤 2)：执行 GameObject→3D Object→Cube 命令新建一个障碍物，将其放置在主角的前方，并赋予黑色材质，如下图所示。

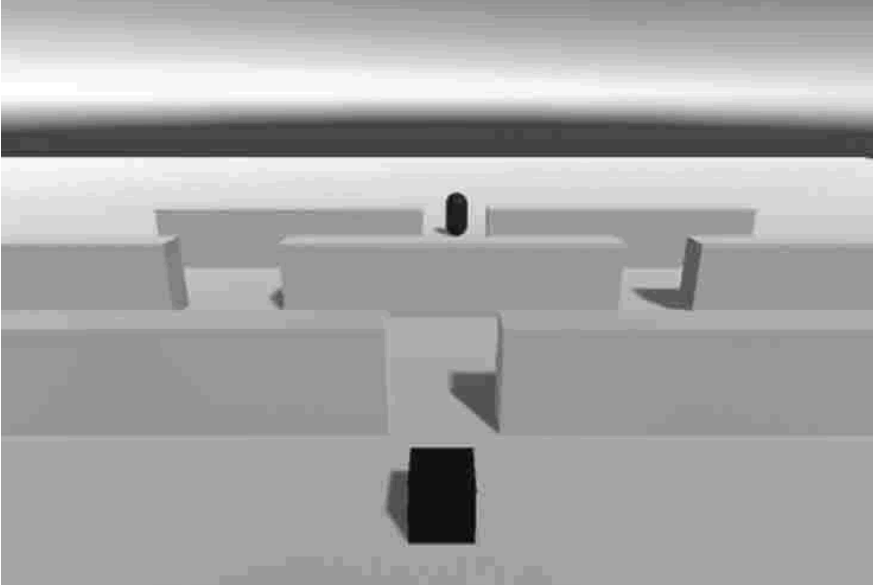


步骤 3)：执行 Component→Navigation→Nav Mesh Obstacle 命令添加 Nav Mesh Obstacle 组件，如下图所示。



步骤 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，效果如下图所示。





1. 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？

用到的的是 NavMeshAgent的destination 和nextPosition这两个 自带的属性变量。

用 NavMeshAgent的nav.remainingDistance参数作为判断 是否到达目的地的参数。

destination是NavMeshAgent组件的 目的地

nextPosition是NavMeshAgent组件的 及时的位置信息