

# Unity寻路系统

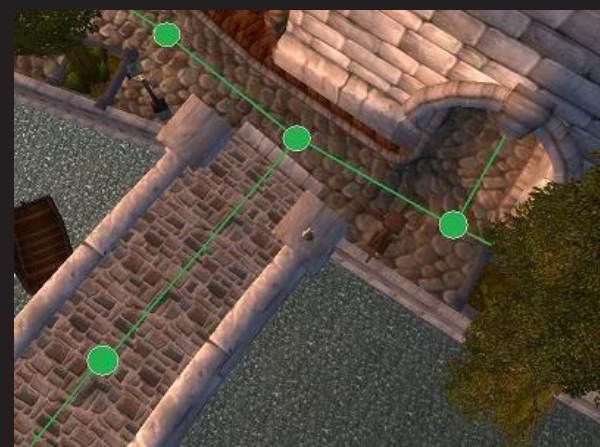
寻路系统

# 常见寻路方法



# 常见寻路方法

- 路点寻路
- 单元格寻路
- 网格寻路



# 实现寻路方法



# 实现寻路方法

1. 确定寻路者。
2. 烘焙寻路路面。
3. 程序实现寻路算法。



# Unity实现寻路步骤：



# 实现寻路步骤：

1. 将场景中不动的物体勾选static。
2. 烘焙寻路网格。
3. 添加NavMeshAgent组件。

(Unity主要通过NavMeshAgent组件实现自动寻路的功能，角色添加NavMeshAgent组件后，就可在NavMesh寻路网格上寻找最优路径找到目标物体.)

4. 给需要寻路的物体添加脚本。



# 寻路物体脚本：

```
NavMeshAgent myNav;  
public Transform target;  
void Start () {  
    myNav = GetComponent<NavMeshAgent>();  
}  
void Update () {  
    myNav.SetDestination(target.position);  
}
```





# NavMeshAgent组件属性



# NavMeshAgent组件属性

**Radius** 寻路的碰撞半径

**Speed** 寻路物体的速度

**StoppingDistance** 停止的距离

**Height** 寻路的碰撞高度

**BaseOffset** 寻路碰撞的位置

**NavMesh Walkable** 寻路物体可寻路的寻路网格层



# 寻路系统区域遮罩



# 寻路系统区域遮罩使用

寻路者可以通过**区域遮罩**，选择不同路面进行寻路效果。



- 避免重复计算（位置不变的话就不要重新计算）。
- Unity寻路系统的动态碰撞有局限性。



# 其他常用寻路插件

- **A\* Pathfinding**
- **SimplePath**

