

WayNode 使用文档

目录

WayNode 使用文档.....	1
简介.....	2
功能.....	2
1) 编辑路径.....	2
2) 核心算法.....	2
3) 跨版本.....	2
使用步骤.....	2
1) 新建场景.....	2
2) 编辑路径.....	3
3) 添加寻路对象.....	3
4) 编写逻辑.....	4
总结.....	4

简介

WayNode 是 Unity3D 一款轻量级的点寻路插件，支持动态查看和编辑路径点，动态寻路，同时跨版本。

功能

1) 编辑路径

编辑路径是该插件最大的优势，对于动态寻路来说，无论是 Unity 自身的 NavMesh 或者是第三方的寻路插件，编辑路径区域都是一个重要功能点。插件提供菜单创建路径点管理器，开发者只需要选择菜单创建路径点，然后进行拖拽编辑路径点即可。

2) 核心算法

WayNode 核心是如何组织路径数据结构和计算寻路算法，路径点数据组织方式以每一个点都依赖一个或多个其他点（Point->N Point）思路，可理解为每一个点可以扩展 N 条路径。寻路算法采用是 Astar 算法（ $F=G+H$ ），F 代表到目标点移动距离，G 代表已移动距离，H 当前点到目标点距离。

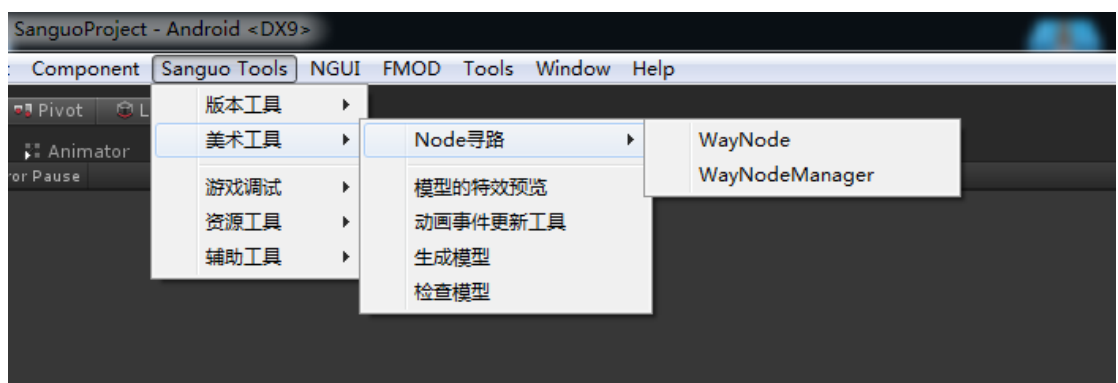
3) 跨版本

该插件支持 Unity 最通用的 API，支持 Unity4.X 或 Unity5.X 版本。

使用步骤

1) 新建场景

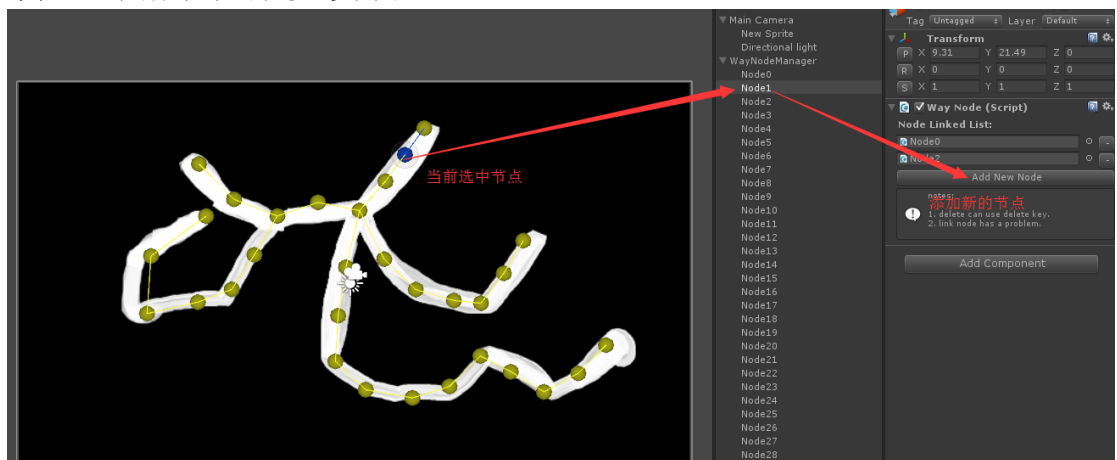
点击菜单创建路径编辑和管理对象，菜单参考下图：



菜单提供两个选项，第一个选项 **WayNode** 会创建一个路径节点同时创建一个路径节点管理器，**WayNodeManager** 只会创建一个 **GameObject** 挂载 **WayNodeManager** 组件节点管理器。

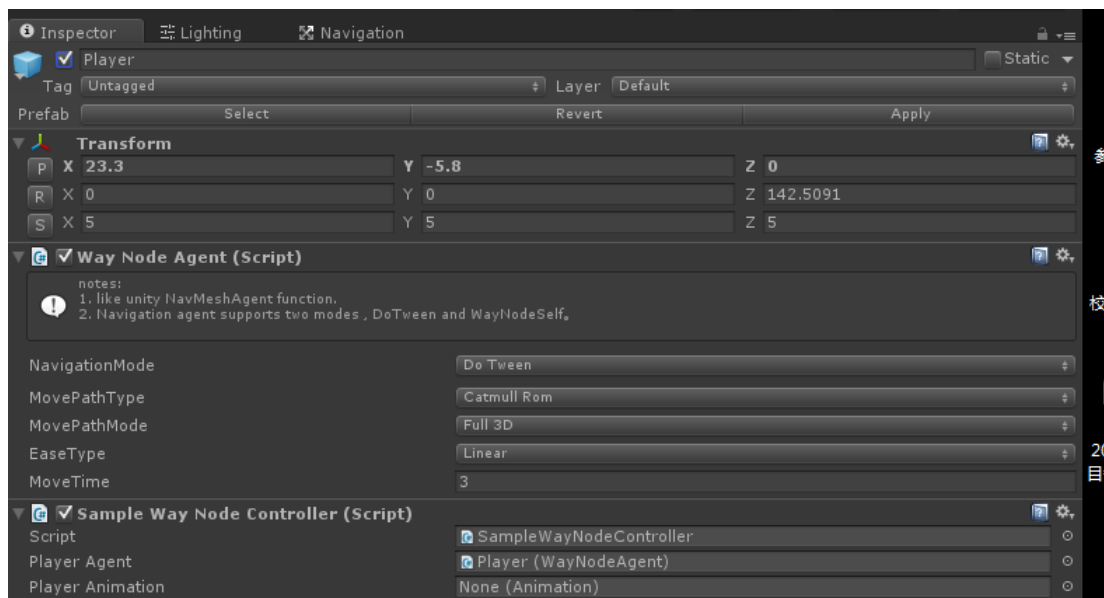
2) 编辑路径

创建一个路径节点和路径节点管理后，选中当前节点，点击 **Add Node** 节点，便会添加一个新的连接到该节点的路径点，当然我们可以选择已经存在的任意节点，然后点击 **Add Node** 按钮添加新的分支节点，最终的结果就会形成一个导航路径点，这里简称这种路径点寻路为 **WayNode**，还有一点值得注意的是 **WayNodeManager** 组件提高了设置节点属性的一些基本信息，比如节点的选中的颜色，节点 **Gizmos** 大小等基本信息。具体效果可以参考下图：



3) 添加寻路对象

这里寻路对象相当于我们平时使用 **NPC** 对象，这里就添加小怪，要想使用 **WayNode** 进行寻路，必须挂载一个组件 **WayNodeAgent**，这个组件功能类似于 Unity 自带导航网格 **NavMeshAgent** 一样，主要实现的功能是提供导航寻路接口，如 **WayNodeAgent.Move(Vector3 from, Vector3 to)**。导航代理主要参数如下图：



4) 编写逻辑

编写一个逻辑控制组件就可以实现 WayNode 寻路，这里写一个简单控制组件 SimpleWayNodeController 即可实现 NPC 按照对应点寻路，组件如下图：

```
public WayNodeAgent PlayerAgent;
public Animation PlayerAnimation;

0 个引用
private void Start()
{
    if (PlayerAgent == null) {
        PlayerAgent = GetComponent<WayNodeAgent>();
    }
    if (PlayerAgent != null) {
        PlayerAgent.OnAgentStartMove += OnStartMove;
        PlayerAgent.OnAgentCompleteMove += OnCompleteMove;
    }
    if (PlayerAnimation == null) {
        PlayerAnimation = GetComponentInChildren<Animation>();
    }
}

0 个引用
private void Update()
{
    if (Input.GetMouseButtonDown(0) && PlayerAgent != null) {
        PlayerAgent.Move(transform.position, Camera.main.ScreenToWorldPoint(Input.mousePosition));
    }
}
```

总结

Unity 一个非常重要的开发流程便是插件式开发，该功能正式 Unity 这种开发模式一个产物，这里希望开发者在开发项目过程中把能独立处理的功能尽量做成插件，这样其他功能也可以使用，下一个项目也可以使用，也非常方便和扩展。