#### Advanced NavMesh techniques to improve Unity Navigation

NavMesh进阶技术改善Unity导航

Objective

目标

The main objective of this blog post is to give you a complete and advanced idea of what NavMesh can do inside Unity and how to speedup your complex navigation in Unity.

 这篇博客的目的是给你一个完整的并且高级的建议告诉你内置Unity NavMesh能做什么并且如何在Unity里加速你的复杂的导航。

Step 1 Before starting NavMesh techniques to improve Unity Navigation

第一步 开始用NavMesh技术改善Unity导航前

I hope you’ve gone through my previous blog & have a basic idea on Unity’s NavMesh. If you haven’t read it, I strongly recommend you to get a [Basic Idea About NavMesh](http://www.theappguruz.com/blog/learn-unity-navigation-using-navmesh).

Let's move on to those techniques now.

我希望你已经阅读了我前面的博客并且有了一个对Unity的NavMesh基本的观点。如果你没有看过，我强烈建议你去读下 [NavMesh基本概念](http://www.theappguruz.com/blog/learn-unity-navigation-using-navmesh)

Step 2 Areas

第二步 区域

An Area in a NavMesh is simply a part of the mesh with a cost. You can find Area tab in the Navigation window.

NavMesh里的一个区域简单来说就是网格代价里的一部分。你可以在Navigation窗口里找到Area标签。



As you can see there are 3 built-in areas in Unity. You can change the cost values of all areas except Not Walkable *(because it’s Not Walkable, you can never go into that area regardless of the cost)*.

你可以看见Unity里这3个内嵌的区域。你可以更改所有区域的代价值除了不可行走部分（因为它不可行走，你永远不可能走进那部分区域不管它的代价是多少）。

You can also define your own area if you need too. Some examples can be mud, water, sand, etc.

The cost of an area plays a decisive role in navigation as our agent (not 007, NavMesh agent ;)) will try to follow the path that has the least possible cost.

你同样可以定义你自己的区域如果需要的话。例如可以是泥地，水域，沙地等。

一个区域的代价决定了角色在导航里成为我们的代理（不是007，是NavMesh的代理）将尝试跟随那个有最小代价可能的路径。

Let’s see how to use them now:

让我们看看如何使用他们：

1. Select the object you want a different area on it.

选择一个你想作为不同的区域的物体。

1. Select area in object tab of navigation window.

在物体的navigation窗口里选择area标签页。



You’ll see the effect after you bake the NavMesh.

制作NavMesh后你将看到效果。

Here’s a link you can refer:

这里是一个相关的连接：

* [NavMesh Baking - Unity Official Tutorials](https://www.youtube.com/watch?v=VcNly-cMZV4&amp;feature=youtu.be&amp;t=2m17s)
* [NavMesh制作-Unity官方教程](https://www.youtube.com/watch?v=VcNly-cMZV4&amp;feature=youtu.be&amp;t=2m17s)

Step 3 Off-Mesh links

第三步 Off-Mesh连接

Off-Mesh links are links that creates a connection between two different meshes. In other words you can jump between multiple meshes using these links.

They can be generated automatically or you can also generate them by off-mesh link component if you want a better control over the navigation behaviour.

Let me show you what difference it can make while learning how to create them.

Off-Mesh连接是在两个不同的网格之间创建一条连接。也就是说你可以在多个网格之间用连接进行跳跃。他们可以被自动生成，或者你可以用off-mesh连接组件来生成他们，如果你想要更好的控制导航表现的话。

To create off-mesh links automatically:

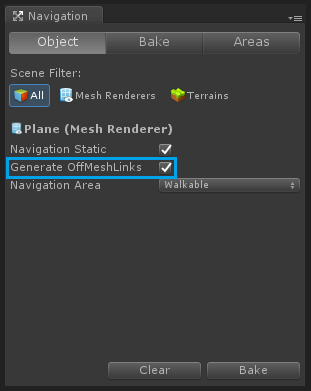
自动创建off-mesh连接：

1. Select the object from hierarchy you want links on in.

在面板上选择你想要连接的物体。

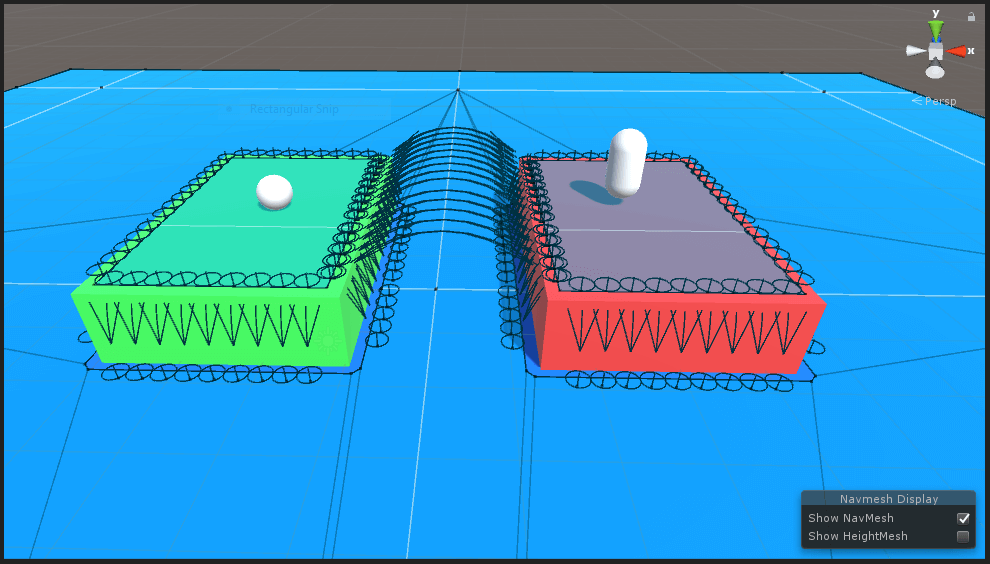
1. Check the Generate Off-mesh Links check box in Navigation window.

在导航窗口里勾选生成off-mesh连接的复选框。



This image displays automatically generated Off-Mesh links.

这张图展示了自动生成off-mesh连接的模样。



Now to create a link manually, follow these steps:

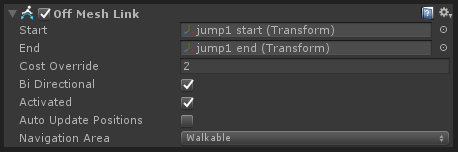
现在我们来手动创建连接，如下步骤：

1. Create two empty GameObjects in hierarchy.

创建两个空物体在面板上。

1. Add Off Mesh Link component to any one of them.

在他们上绑定off-mesh连接组件。

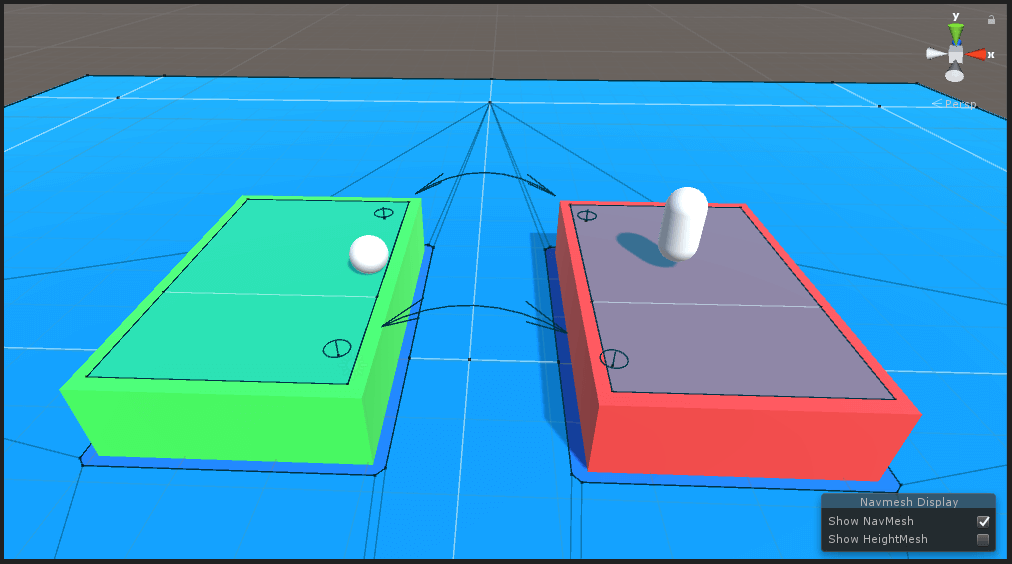


As you can see there are 2 variables named Start & End in it. Assign the respective GameObject in it.

你可以看到这里有2个变量叫Start和End。分别复制物体给它。

This image shows a link that I created.

这张图展示了我创建的连接。



Note

注意

If the link you created is not a valid link (both ends are placed on a mesh surface), then the color of link will turn into gray rather than black. To see the effect just move any one end object of the link upwards (+y direction) and notice the change in color.

I guess you got the difference by now.

If you opt for automatic link generation, Unity will create every possible link to connect meshes which might not be ideal for your case. If you want a specific route to be followed you can use Off Mesh Link component.

You can choose any of these two methods that suites your requirement.

如果你创建的不是一个有效的连接（两端都被放置在同一个网格上），那么连接的颜色将会变成灰色而不是黑色。为了看这个效果，只要移动任何一端的物体向上（正y方向）就能看到颜色改变了。我猜你现在已经知道不同点了。

如果你选择自动生成连接，Unity将会创建每个可能的连接来连接网格，可能不是你理想的情况。如果你想要一个特殊的路线来跟随，你可以使用off-mesh连接组件。

你能选择这个两个方法的任意一个来适应的你需求。

See how it looks in action here:

在这里看它是如何执行的：

* [Unity - Off-mesh Links Tutorial](https://unity3d.com/learn/tutorials/topics/navigation/mesh-links)
* [Unity – Off-mesh 连接教程](https://unity3d.com/learn/tutorials/topics/navigation/mesh-links)

Step 4 NavMesh Obstacles

第四步 NavMesh障碍物

Until now we were dealing with static objects that don’t move around when the game is running.

But what if we have a moving object that we need to be a part of our navigation mesh?

*NavMesh Obstacle component is the answer to that. The process is simple.*

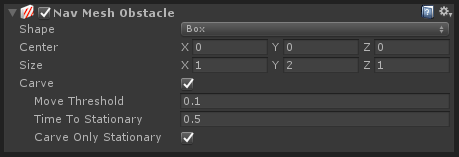
*First add NavMesh Obstacle component to the object.*

至今我们还在处理在游戏中不会到处移动的静态物体。

但是如果我们有一个移动的物体想成为我们导航网格的一部分呢？

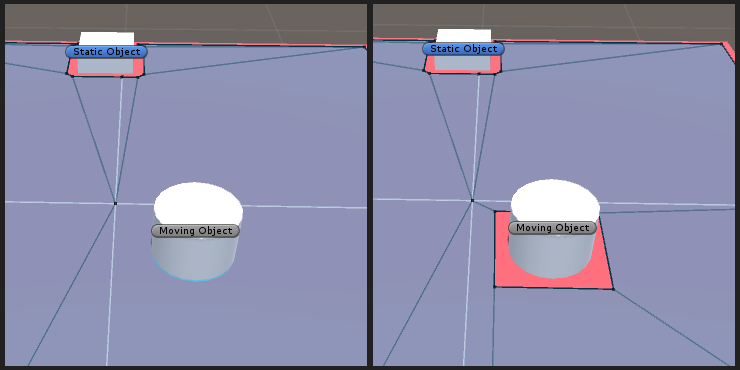
NavMesh障碍物组件回答了这个问题。处理起来很简单。

首先在物体上绑定NavMesh Obstacle组件



You won’t see any difference now. Check the carve option and you should be seeing a hole in the mesh.

你看不到任何的不同。勾选carve选项，你应该能看见在Mesh里一个块。



Character with NavMesh Agent component will avoid this part of navmesh in pathfinding process.

Some examples can be patrolling enemies or moving doors.

绑定着NavMesh Agent组件的角色会在处理寻路时避开这部分Navmesh。

See the obstacles working live here:

看到障碍物在这里运行：

* [Unity - NavMesh Obstacles Tutorial](https://unity3d.com/learn/tutorials/topics/navigation/navmesh-obstacles)
* [Unity – NavMesh 障碍物教程](https://unity3d.com/learn/tutorials/topics/navigation/navmesh-obstacles)

Step 5 Conclusion

第五步 结论

*I hope you got all the things i wanted to pass on about Unity’s NavMesh. This is where I end this post and you start experimenting on navigation meshes.*

*Try it on your game and don’t forget to drop a comment if you have any query or suggestions for this post.*

*我希望你能学到所有我说的关于Unity NavMesh的经验。文章在这里结束了，你开始在导航网格里试验吧。尝试运用在你的游戏里，并且不要忘了如果你有任何疑问或建议请留下评论。*