#### 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

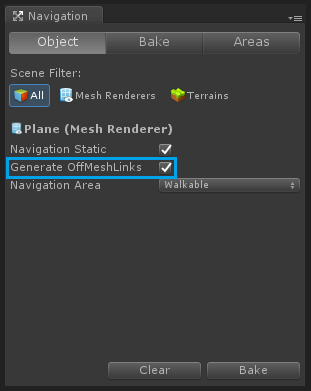
Off-Mesh links are links that creates a connection between two different meshes. In other words you can jump between multiple meshes using these links.

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.

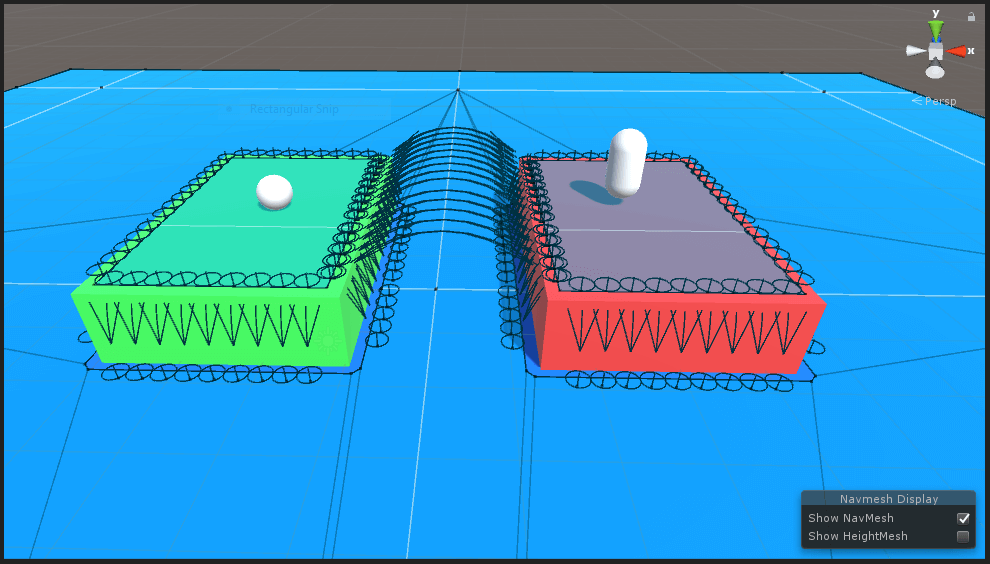
Let me show you what difference it can make while learning how to create them.

To create off-mesh links automatically:

1. Select the object from hierarchy you want links on in.
2. Check the Generate Off-mesh Links check box in Navigation window.

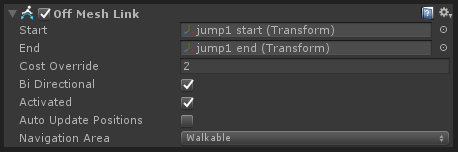


This image displays automatically generated Off-Mesh links.



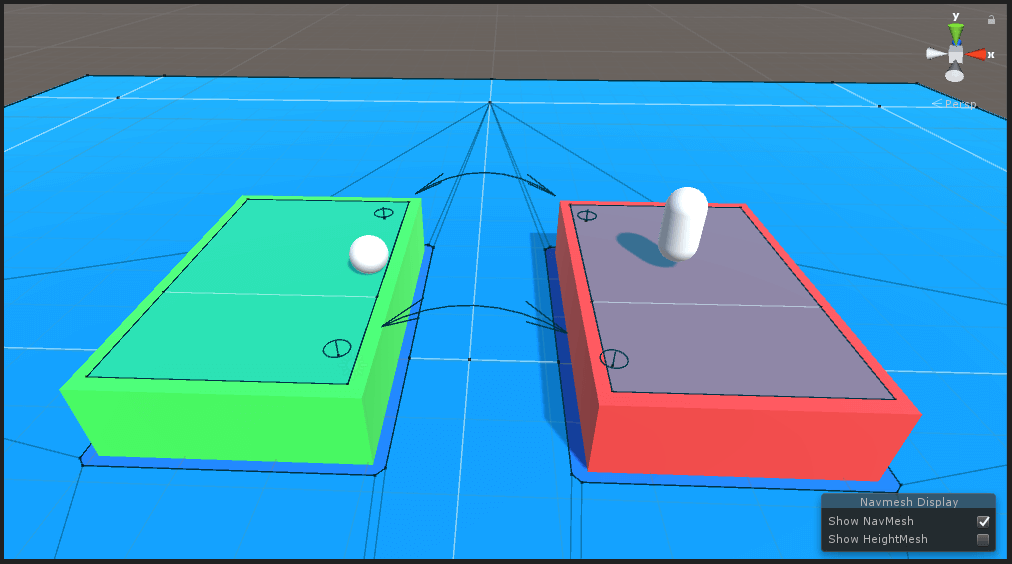
Now to create a link manually, follow these steps:

1. Create two empty GameObjects in hierarchy.
2. Add Off Mesh Link component to any one of them.



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

This image shows a link that I created.



Note

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.

See how it looks in action here:

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

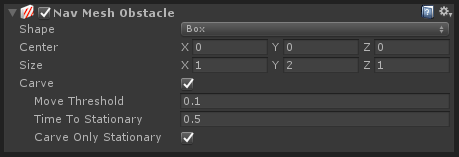
Step 4 NavMesh Obstacles

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

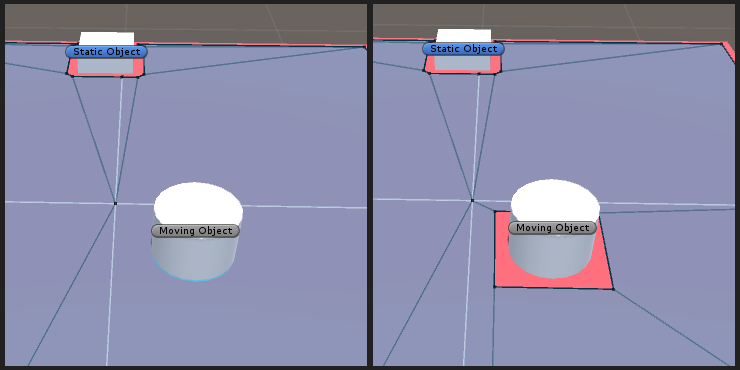
But what if we have a moving object that we need to be a part of our navigation mesh?

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

*First add NavMesh Obstacle component to the object.*



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



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

Some examples can be patrolling enemies or moving doors.

See the obstacles working live here:

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

Step 5 Conclusion

*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.*

Got an Idea of Game Development? What are you still waiting for? [Contact us](http://www.theappguruz.com/contact-us) now and see the Idea live soon. Our company has been named as one of the best [Unity 3D Game Development](http://www.theappguruz.com/3d-game-development) Company in India.

- See more at: http://www.theappguruz.com/blog/navmesh-techniques-to-improve-unity-navigation#sthash.nm7Y7I3c.dpuf