NavMesh Extension
Documentation

v1.2

Scripting Reference 1
Quick Start 2
NavMesh Manager 3
NavMesh Editing 4
Portal Manager 5
Agent Controller 5
Contact 6

Thank you for buying NavMesh Extension!
Your support is greatly appreciated.

Scripting Reference

www.rebound-games.com/docs/nav

Quick Start

NavMesh Extension provides two manager objects, which are extending **Unity's NavMesh system**. You can read more about Unity NavMesh in their documentation manual.

If you are using **Unity 5.6 or above**, then Unity's component based NavMesh scripts are already included in this asset. If you have imported them yourself too, then you might experience errors due to duplicated files. In this case, please delete either of them.

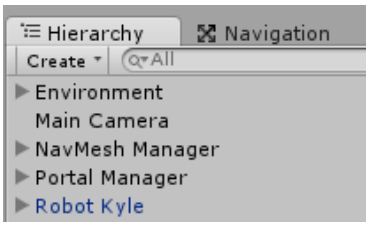
The following chapters will describe both manager objects of this asset:

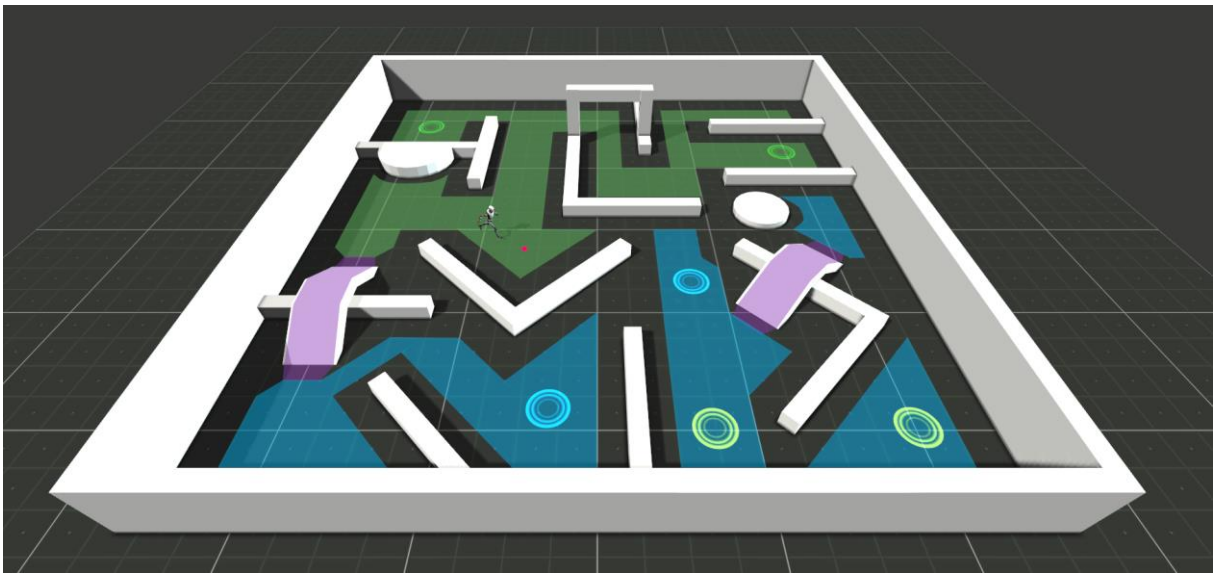
- **NavMesh Manager**
- **Portal Manager**

If you would like to wrap your head around a basic integration of both components, please open the **example scene** located under NavMeshExtension > Examples.

The example scene makes use of both manager objects and contains a **NavMesh agent** that walks to the position you click.

Its navigation mesh is built with the NavMesh Manager, and the agent will find the shortest way by using portals of the Portal Manager.





Press play to see the result!

We've uploaded a **timelapse** of building the example scene on our channel:

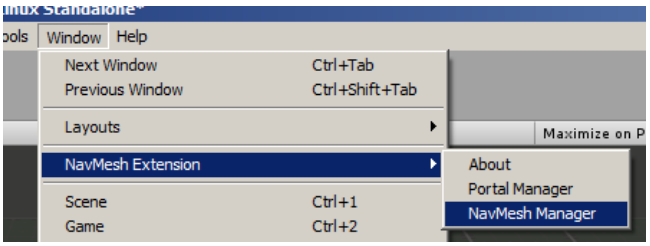
- https://www.youtube.com/watch?v=5YosSfqfN6w

Please read further to learn more about scripts and techniques when creating NavMesh areas as well as about portals and script access for Unity NavMesh agents.

NavMesh Manager

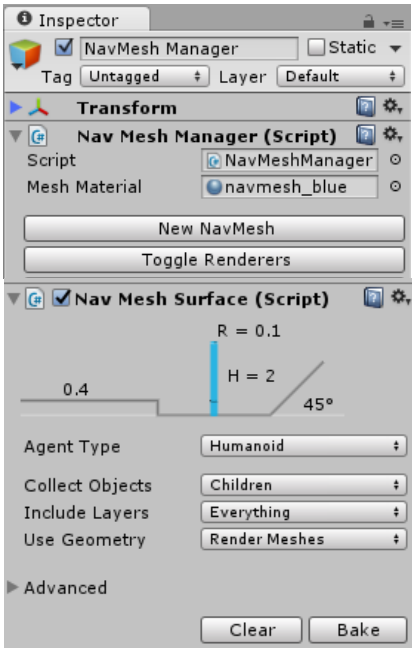
The **NavMesh Manager** allows you to build custom walkable areas for use with Unity's NavMesh system.

You can add a new NavMesh Manager component to your scene via Window > NavMesh Extension > NavMesh Manager.



Once added to your scene, you can create new NavMesh areas by clicking on the **New NavMesh** button. If you specify a **mesh material**, this material will be used on newly created meshes.

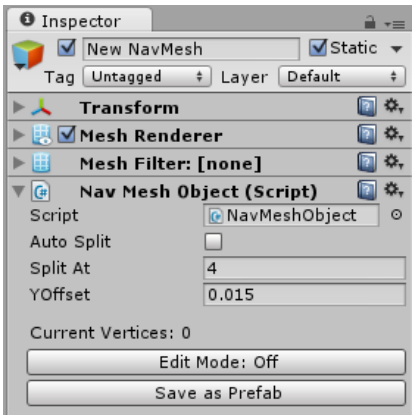
The **Toggle Renderers** button will show or hide all renderers of NavMesh areas parented to the manager object.



If you would like to **bake the NavMesh** areas created with this manager for your navigation agents, please use the bake button on the **NavMeshSurface** component directly.

After you've clicked on "New NavMesh", a new gameobject with a **NavMesh Object script** on it will be created. This script manages adding vertices and triangles in the edit mode, as well as combining submeshes into one mesh when leaving edit mode.

When placing vertices and **Auto Split** is enabled, new submeshes will be created automatically when the value of **Split At** is reached. In this case, a new submesh would be created every 4 vertices. If "Auto Split" is disabled, you are free to control the amount of vertices per mesh by yourself. More on that later.

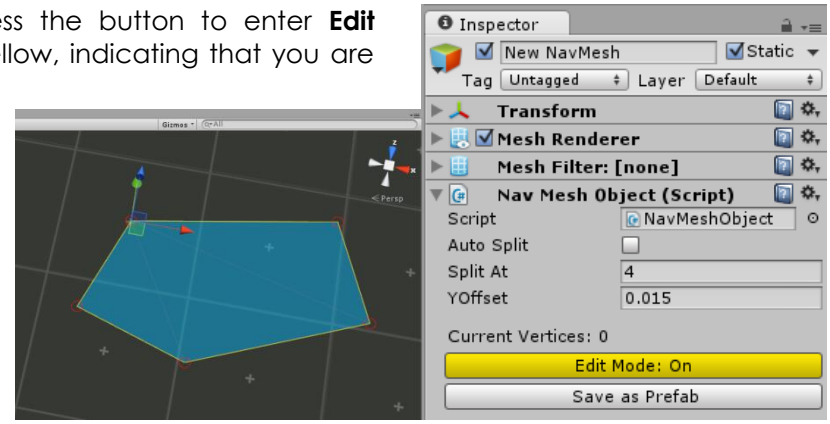


YOffset is an optional height value when adding vertices onto colliders.

There is also the option to export the created mesh to a mesh asset and prefab by pressing **Save as Prefab**. These files will be exported to your project under NavMeshExtension > Prefabs. You can adjust this path in the NavMesh Object Editor script.

NavMesh Editing

On a NavMesh Object, press the button to enter **Edit Mode**. The button will turn yellow, indicating that you are actively placing vertices.

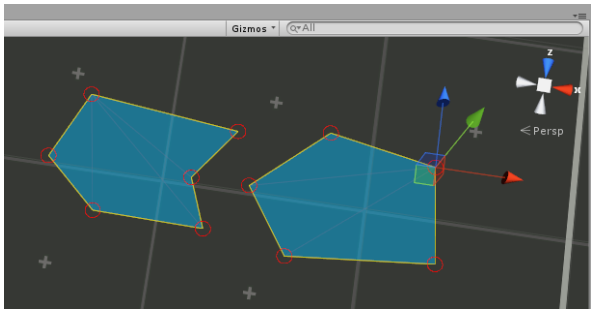


Now you are able to place vertices just by **clicking on colliders** in the scene view.

As "Auto Split" is disabled, you are able to place an unlimited amount of vertices per submesh, as seen in this image.

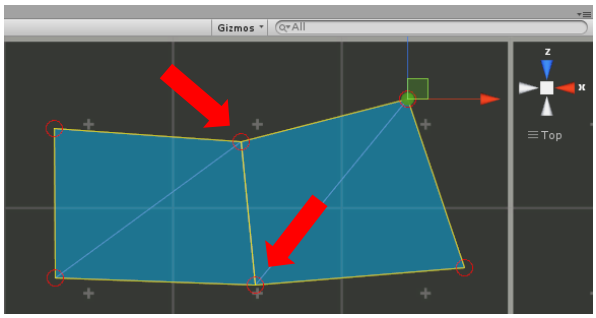
If you would like to create a new submesh that is not connected to the other ones, **hold CTRL + mouse click** on a collider in the scene view. Only hold CTRL **for the first vertex**, all further vertices will then be connected to this submesh.

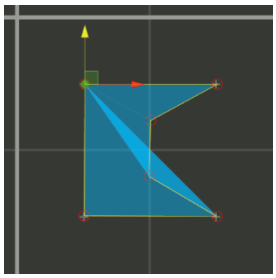
If "Auto Split" is enabled, this will happen automatically after the defined amount.



When placing vertices and clicking on the vertex on another submesh, both vertices will be merged into one vertex.

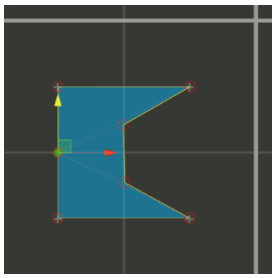
Please note that **complex structures**, such as **holes or overhangs**, cannot be created with one submesh. Also you will have to start the submesh at a position that makes sense for further vertices to avoid meshes like this:



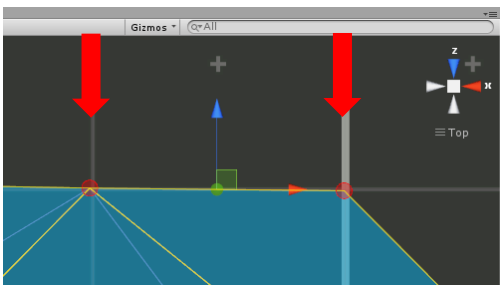


BAD

GOOD



To **leave the Edit Mode** and combine all submeshes into one mesh, press the Edit Mode button again. **Always do this**, otherwise no mesh will be generated!

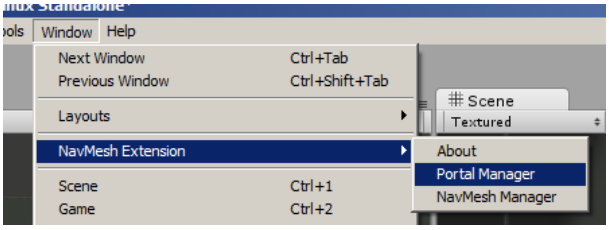


If you want to **move vertices** later, simply select them with the **left mouse button** while the **edit mode is off**. This will highlight them and show a centered handle. Delete selected vertices by pressing **backspace**. **Click them again** or **right-click** to deselect vertices.

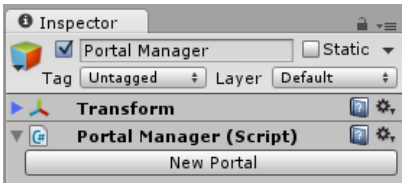
Portal Manager

The **Portal Manager** component connects two positions on a baked NavMesh via portals, so that NavMesh agents can or cannot utilize them to find the shortest way to a target destination.

To add a new Portal Manager component to your scene, select Window > NavMesh Extension > Portal Manager.

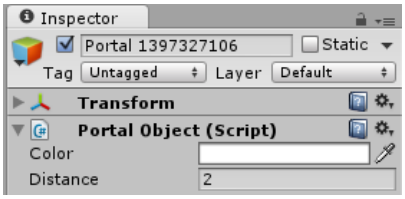


There's not much to see on the Portal Manager script – just a button to create new portals. Press **New Portal** to add them to your scene.

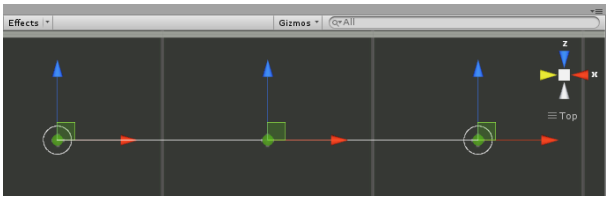


This will create a new portal parent gameobject with a unique name and two child objects, the actual portals.

You can assign a custom **gizmo color** in the inspector. When moving them in the scene view, **Distance** will display the linear distance between the two portals.



Make sure to place the portals on top of your NavMesh as precise as possible.



Agent Controller

NavMesh Extension comes with a sample integration of NavMesh agents, the **Agent Controller** script. This script controls the NavMesh agent in the example scene and listens for user input to set the agent's target destination.

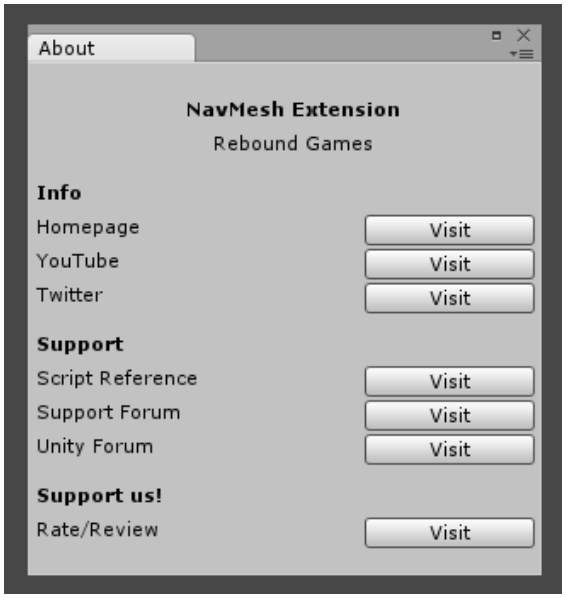
When setting the target destination, it calls **PortalManager.GetPath** with the **starting and ending position**. The Portal Manager component then calculates a path to the destination while taking all portals into account. After the calculation, an **array of Vector3 positions** is returned that contains multiple start and end points (e.g. start to portal 1, portal 1 to target).

Let Unity move the agent over these points on the NavMesh and that's it!

Contact

As full source code is provided and every line is well-documented, please feel free to take a look at the scripts and modify them to fit your needs.

If you have any questions, comments, suggestions or other concerns about our product, do not hesitate to contact us. You will find all important links in our About window, located under Window > NavMesh Extension.



For private questions, you can also email us at
info@rebound-games.com

If you would like to support us on the Unity Asset Store, please write a short review there so other developers can form an opinion. Again, thanks for your support, and good luck with your apps!

Rebound Games