

# **Navigation And Pathfinding System**

## **Quick Start Guide**

## ***Step 1.***

### ***- Import NAPS package into current project***

In order to start working with NAPS just import NAPS package into current project. After that NAPS editor will be able in «Window» menu of Unity editor.

## ***Step 2.***

### ***- Open game level with designated collision model***

### ***- Create new navigation file***

### ***- Add new path type to this navigation file***

### ***- Create nodes inside this path type***

### ***- Save file***

Open game level scene you want to create navigation for and ensure that is collision model exist for this level. After that open NAPS editor (Window\NAPS), select option of new file creation, add one path type to this file and put some nodes inside it. In order to understand how to do this follow instructions of subsection «Getting started», «Path types. Creation, visualization, deleting», «Creating a path nodes» of section «Work with NAPS editor» of NAPS manual. You also can see the process of navigation creation in video version of this guide – Unity3d NAPS Quick Start Guide.

In this example navigation will be used by basic AI which are an example of simple path-following AI for NAPS. This AI can only generate random routes inside navigation grid and follow them using base movement methods. So place nodes on horizontal surfaces because by default this AI don't know how to move by vertical surfaces.

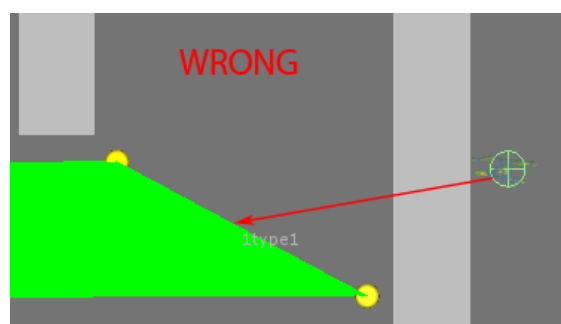
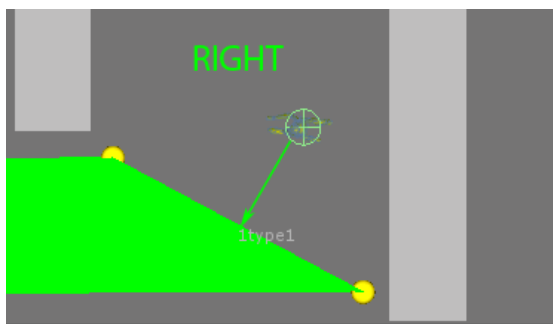
When all nodes was placed – save file.

### **Step 3.**

- **Add *PathfindingManager.cs* script to any active game object in scene, enter file name and file path**
- **Drag *NPC* prefab into scene**
- **Allow this *NPC* follow by created path type**

The script *PathfindingManager.cs* need only for loading navigation files when application was started. With it help you also can load navigation files at any moment in realtime.

When path type was created and nodes was added to it it's time to allow robot controlling by AI start move using it data. For this purpose drag a prefab named «Character» from folder «prefabs\characters» into scene, ensure that the robot is not stack in some collider and in the same time is in a direct view of some node of navigation grid.

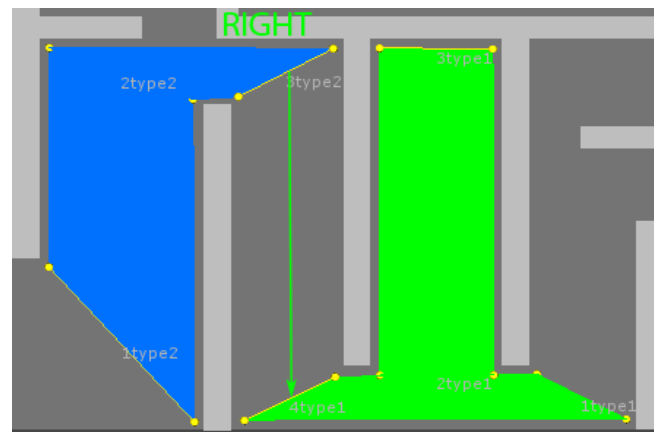
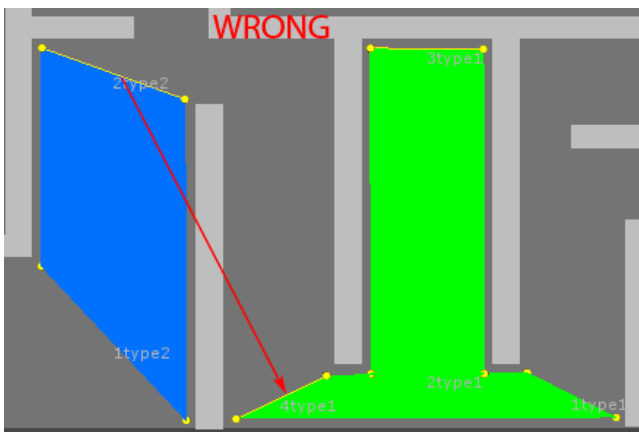


Then select «Character» object and find variable named «Types» in the field of *PathFollowingAI* script in the inspector. This variable is an array, set it size equal to 1 and in the field «element0» enter the name of path type created before. After that, when you press «Play» button you will see that the robot starts moving.

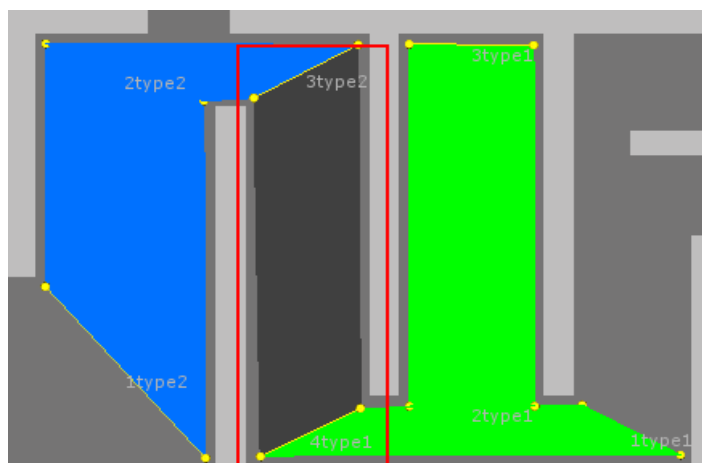
#### **Step 4.**

- **Add one more path type to the current file**
- **Create nodes inside this path type**
- **Add one more NPC into scene**
- **Allow this NPC follow by two path types**

Add one more path type to the current navigation file as was done before. That is enter any name in the field «New path» and press button «Add». In order to start placing nodes inside this path type select it with help of arrow buttons in the field «Current path type» and press button «Start placing». Place the nodes of this path type in the way, in that from some its node can be seen some node of previous path type.



Then connect this nodes of different path types by two-sided atypical connection. In order to understand how to do that see subsections «Work with connections of nodes» and «Atypical connections» of section «Work with NAPS editor» of NAPS manual. When atypical connection was added it will displayed in the scene view by grey color.



With help of atypical connection we combine two different path types into one. That mean that for now NPC can use both path types for movement as different navigation grids and as one big navigation grid.

Add one more NPC exemplar in scene, you can just duplicate existing NPC. Put this NPC near to previous NPC and set size of variable «Types» of PathFollowingAI script equal to 2. In the field «element0» enter the number of first path type, in the field «element1» - name of the second path type. After you press button «Play» you will see that one of the robots walk through both navigation grids while the second – only through one.

### **Step 5.**

**- Create new path type for camera using single points**

**- Add a curves between connected nodes**

**- Drag into scene CinematicCamera prefab and set up route to following**

In order to make path for camera, create new path type as was done before, but using only single points. Place that points in key positions. Read subsection «Creating a path nodes» of section «Work with NAPS editor» of NAPS manual to understand how to make single points. After that add a curves between connected nodes and set required shapes for this curves, in order to understand how to make curves read subsection «Work with curves» of section «Work with NAPS editor» of NAPS manual.

Drag into scene CinematicCamera prefab. Select its game object in hierarchy view and set the size of variable «Route» equal to number of nodes in the path for camera. After that set up route by filling fields with nodes parameters in the order in which they connected by curves.

If all done right, then after pressing button «Play» camera will start moving by created path. If you want to use this route as a cycle, then the size of route should be increased by one, because the last point of the route must be equal to first point and «Cycle route» flag should be set to true. In this case camera will be cyclically move by this path.

NOTE: You can also add a curves to any connected nodes and use the sequence of this nodes as a path for camera or any other type of movement by this curves. But if you want to create specialized path just for camera i. e. for movement only by curves, then it is not necessary to create bipoints, use single points instead.

For more information about using NAPS read NAPS manual and explore examples of source codes.