# NavMesh Obstacles and Off-Mesh Links

## Obstacles

Use these for destructible objects that may be blocking the navmesh now, but may be destroyed later.

Also use these for doors that open and close.

Place a NavMeshObstacle component on a GameObject, choose a shape (Box or Capsule) and scale it in the editor.

You can also set the “Carve” option, which determines if the object carves a hole in the NavMesh.

Without Carve, NavMeshAgents will try to path through it, but stop if it’s in the way.

With Carve, they will treat it as a permanent obstacle and try to path around it.

## Off-mesh Links

Off-mesh links allow a NavMeshAgent to drop or jump between navigatable areas.

### Automatic

You can set drops and horizontal jumps up automatically. On the Objects tab of Navigation in the Inspector, click Generate OffMesh Links for each objects where you want a link to start.

In the Bake tab, set the Drop height and to the largest distance an agent can drop safely, and Jump Distance for maximum horizontal jump.

### Manual

You can manually add an Off-Mesh Link as a component by specifying a start and end Transform for the Agent to Jump from. Unlike automatic links, these ones can go up.

The easiest way to work with this is to make a prefab – a transform with the OffMeshLink component, referencing itself as Start, and a child transform that is registered as the end.

## Walkable areas

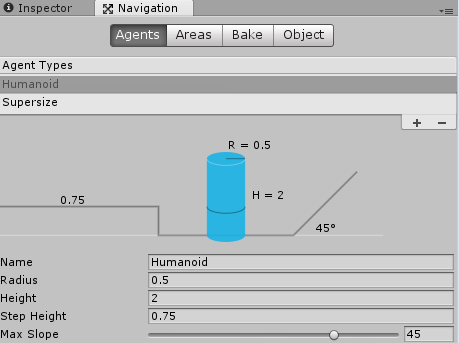
In the areas tab you can add new areas. I’ve created one called “Gold” in the Areas tab of the Navigation window, and set its Cost to 0.1, so its ten times preferable to walk on vs the standard walkable areas.

With the paths activated and baked, observe that the player agent prefers to use the paths to get around.

You can set it to ONLY use the paths by setting its NavMeshAgent’s Area Mask to Gold only.

## Agents of different sizes

You can make a second agent size in the Navigation: Agents tab



If you set a NavMeshAgent to use this type though, it will report that it isn’t in the NavMesh and it can’t navigate!

You have to get the NavMeshcomponents extra files from github here (<https://github.com/Unity-Technologies/NavMeshComponents/tree/2017.2>) or grab them from R drive.

1. Copy the NavMeshComponents folder into your project
2. Stick a NavMeshSurface on an object in your level
3. Set it to use the Agent Type you’ve specified.
4. Bake!
5. Your extra agent can now move around freely.

Note that you’ll need to rebake all of these when you make a new Navmesh.