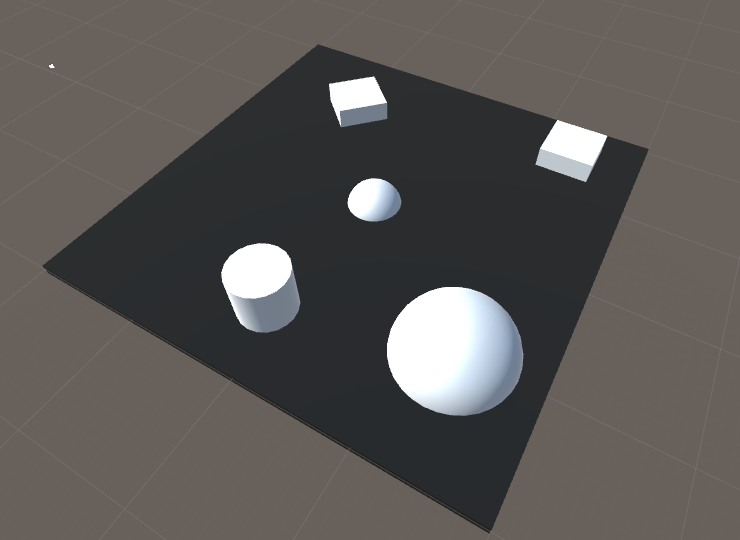
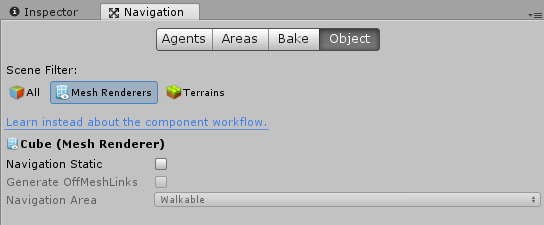
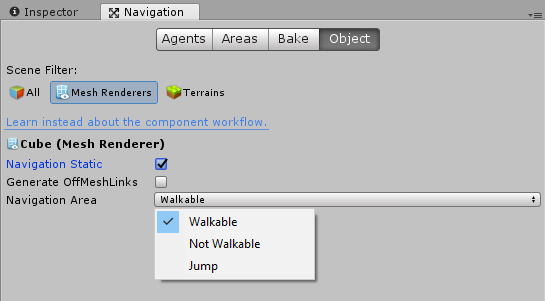
# Nav Agent and Nav Mesh

# Building a Nav Mesh

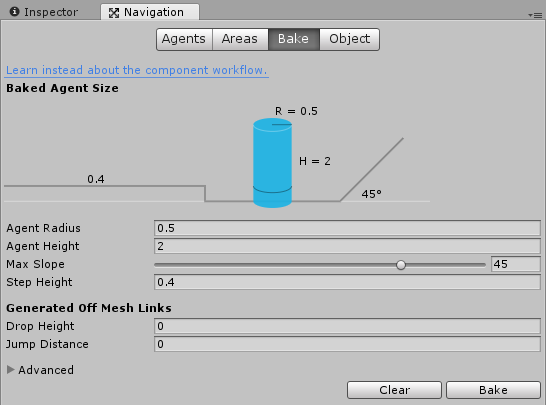
## Defining walkable Areas

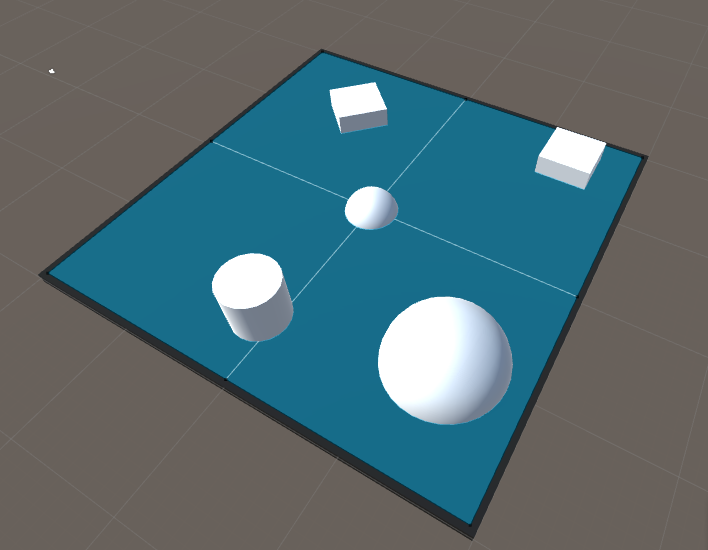




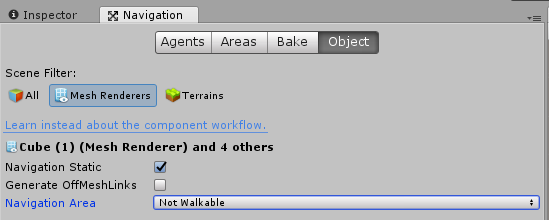


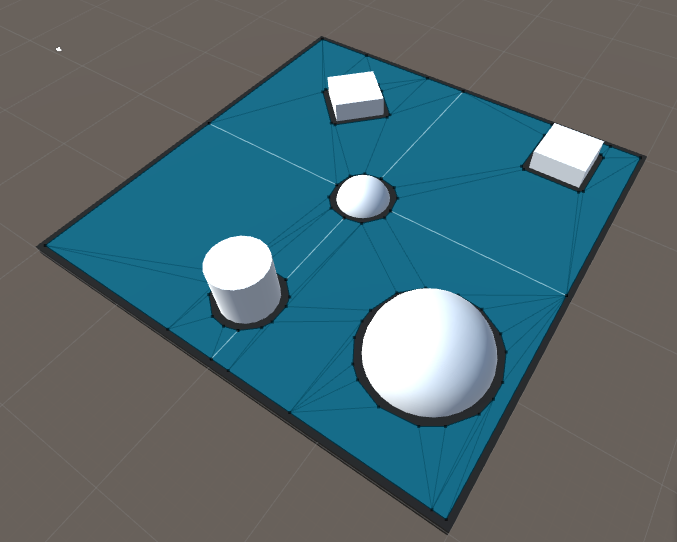
## Bake Nav Mesh



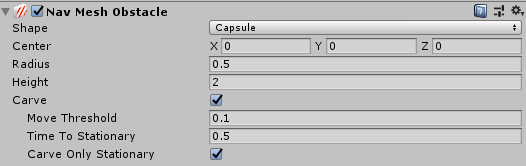


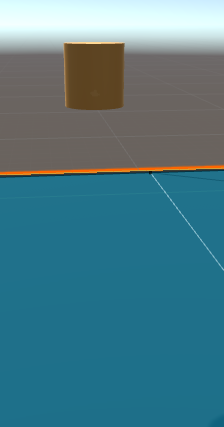
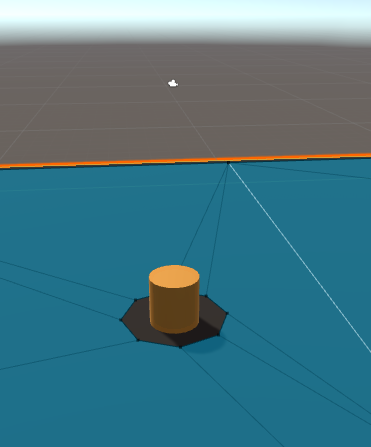
## Defining Non-Walkbale Areas



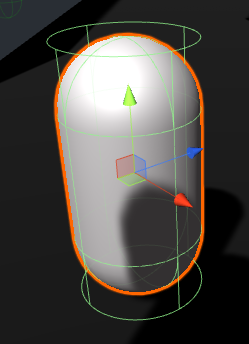
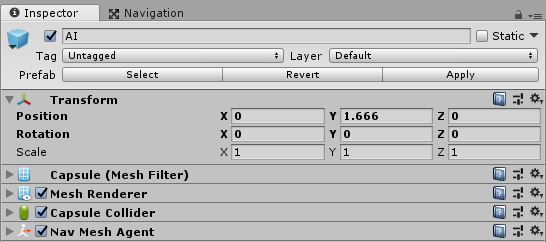


# Dynamic Objects



# Nav Agent

## Important Properties



### Destination



The target location where you want the AI to navigate to. Changing this value will cause a new path to be generated.

### Corners



All the Vector3 positions along the calculated path are stored here.

### Status



The current status of the nav agent. Possible values

* PathComplete
* PathPartial
* PathInvalid

### Remaining Distance



A float with the remaining distance on the path. Will be zero when the destination is reached.

## Ray from Mouse Cursor



Returns a ray from the cursor in the direction of camera. This projects the 2D cursor position to 3D world space.

# Exercises

## Exercise 1: Follow Cursor

Create a script called AiFollowCursor.

When the left mouse button is clicked a ray should be cast from the mouse cursor (Use the ray from mouse cursor above). The destination of the AI should be set to the impact point of the ray hit result.

## Exercise 2: Follow Path

Create an empty GameObject called PathMarker,

* Create a new Tag called PathMarker,
* Assign this Tag to the object.
* Add a SphereCollider to the object. Ensure this is set to be a trigger.
* Create a PreFab from this object.

Create a Script called PathMarker

* Add one public variable called NextMarker of type PathMarker
* Add this script to the PathMarker PreFab

Create a script called AiFollowPath.

* This script will have a public variable of PathMarker called Target
* The location of this marker will be the starting destination for the nav agent.
* Add the OnCollisionEnter method
  + When the AI collides with an object tagged as PathMarker
  + Get the PathMarker component from the object the AI hit
  + Set your Target variable to be the PathMarker component you pulled from the hit object
  + Move the AI to the location of the new Target
* Add a few PathMarkers to the scene. Link the PathMarkers together, ensure the first and last path marker are linked.
  + Path Markers hold the next marker to be moved to
  + Linking markers is filling in the NextMarker variable of each path marker object