# Unit 4 – Lesson 9. Implementing Nav Mesh

**Aim:**

* What is navigation mesh and how do we create it in Unity?

**Objectives:** After the lesson, students should be able to:

* Obtain understanding of navigation mesh
* Implement Navigation Mesh in Unity games

**CLASS PROCEDURE:**

***Do Now:*** Go to the public drive and lesson notes folder, and copy the zip file named Standard Assets Example Projects to your USB flash drive. Unzip the file. Open Unity, and then open the scene named “CharacterThirdPersonAI”. Run the game project. Click on anywhere and watch Ethan automatically running towards the given position.

***Discussion / Presentation:***

1. Let’s take a look at the AICharacterControl.cs. How does the program make Ethan finding the right path?

***AICharacterControl.cs:***

using System;

using UnityEngine;

namespace UnityStandardAssets.Characters.ThirdPerson

{

[RequireComponent(typeof (NavMeshAgent))]

[RequireComponent(typeof (ThirdPersonCharacter))]

public class AICharacterControl : MonoBehaviour

{

public NavMeshAgent agent { get; private set; } // the navmesh agent required for the path finding

public ThirdPersonCharacter character { get; private set; } // the character we are controlling

public Transform target; // target to aim for

private void Start()

{

// get the components on the object we need ( should not be null due to require component so no need to check )

agent = GetComponentInChildren<NavMeshAgent>();

character = GetComponent<ThirdPersonCharacter>();

agent.updateRotation = false;

agent.updatePosition = true;

}

private void Update()

{

if (target != null)

agent.SetDestination(target.position);

if (agent.remainingDistance > agent.stoppingDistance)

character.Move(agent.desiredVelocity, false, false);

else

character.Move(Vector3.zero, false, false);

}

public void SetTarget(Transform target)

{

this.target = target;

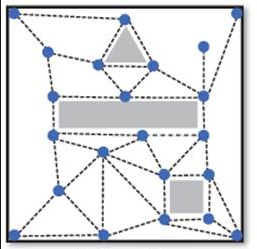
}

}

}

1. What is NavMesh and what does it do?

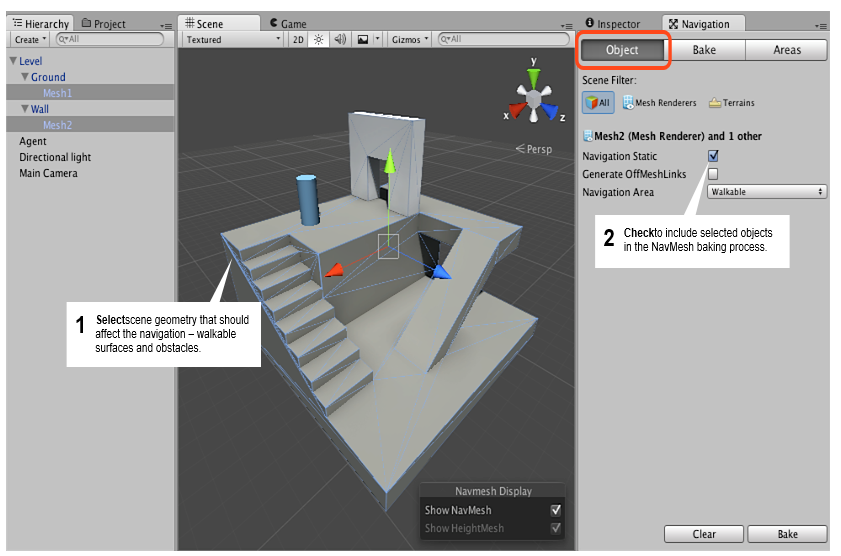
* A **navigation mesh**, or **navmesh**, is an abstract data structure used in artificial intelligence applications to aid agents in path finding through complicated spaces. This approach has been known since at least the mid-1980s in robotics, where it has been called a meadow map, and was popularized in video game AI in 2000. [wiki]

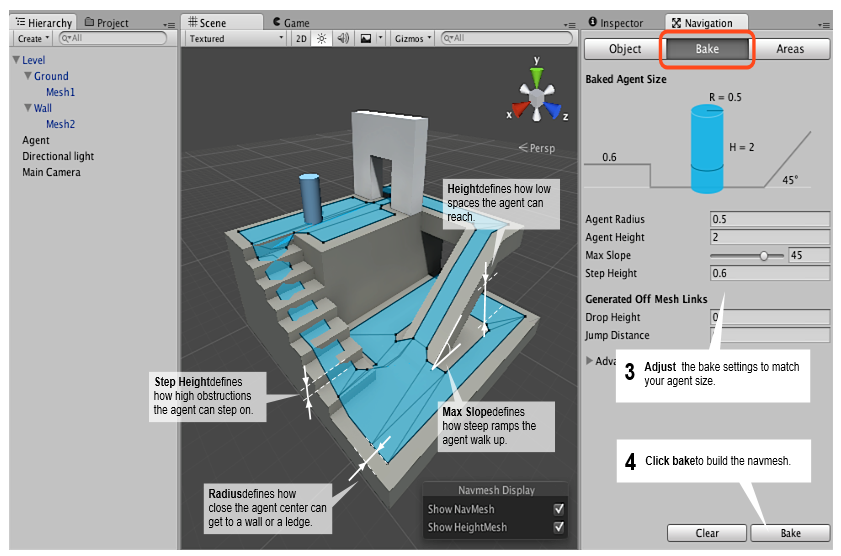


* A navigation mesh is a collection of two-dimensional [convex polygons](https://en.wikipedia.org/wiki/Convex_polygon) (a [polygon mesh](https://en.wikipedia.org/wiki/Polygon_mesh)) that define which areas of an environment are traversable by agents. In other words, a character in a game could freely walk around within these areas unobstructed by trees, lava, or other barriers that are part of the environment. Adjacent polygons are connected to each other in a [graph](https://en.wikipedia.org/wiki/Graph_data_structure). [wiki]
* Pathfinding within one of these polygons can be done trivially in a straight line because the polygon is convex and traversable. Pathfinding between polygons in the mesh can be done with one of the large number of [graph search](https://en.wikipedia.org/wiki/Graph_search) algorithms, such as [A\*](https://en.wikipedia.org/wiki/A*).[[1]](https://en.wikipedia.org/wiki/Navigation_mesh#cite_note-FOOTNOTESnook2000294.E2.80.93295-1) Agents on a navmesh can thus avoid computationally expensive [collision detection](https://en.wikipedia.org/wiki/Collision_detection) checks with obstacles that are part of the environment. [wiki]
* Representing traversable areas in a 2D-like form simplifies calculations that would otherwise need to be done in the “true” 3D environment, yet unlike a 2D grid it allows traversable areas that overlap above and below at different heights.[[2]](https://en.wikipedia.org/wiki/Navigation_mesh#cite_note-FOOTNOTESnook2000289-2) The polygons of various sizes and shapes in navigation meshes can represent arbitrary environments with greater accuracy than regular grids can.[[3]](https://en.wikipedia.org/wiki/Navigation_mesh#cite_note-FOOTNOTEBrand20094-3)

1. How do we set up Navigation Mesh in Unity?

<https://docs.unity3d.com/2018.3/Documentation/Manual/nav-BuildingNavMesh.html>





1. **Select** scene geometry that should affect the navigation – walkable surfaces and obstacles.
2. **Check Navigation Static** on to include selected objects in the NavMesh baking process.
3. **Adjust** the bake settings to match your agent size.
   * *Agent Radius* defines how close the agent center can get to a wall or a ledge.
   * *Agent Height* defines how low the spaces are that the agent can reach.
   * *Max Slope* defines how steep the ramps are that the agent walk up.
   * *Step Height* defines how high obstructions are that the agent can step on.
4. **Click bake** to build the NavMesh.

Refer to <https://docs.unity3d.com/Manual/nav-BuildingNavMesh.html> for more details.

***Pair – sharing Activity – Mesh basics:***

<https://learn.unity.com/tutorial/navigation-basics>

Continue working on your midyear project.