HOW TO

h8man edited this page 30 minutes ago · 6 revisions

NAVMESH PLUS

by h8man

INTRO

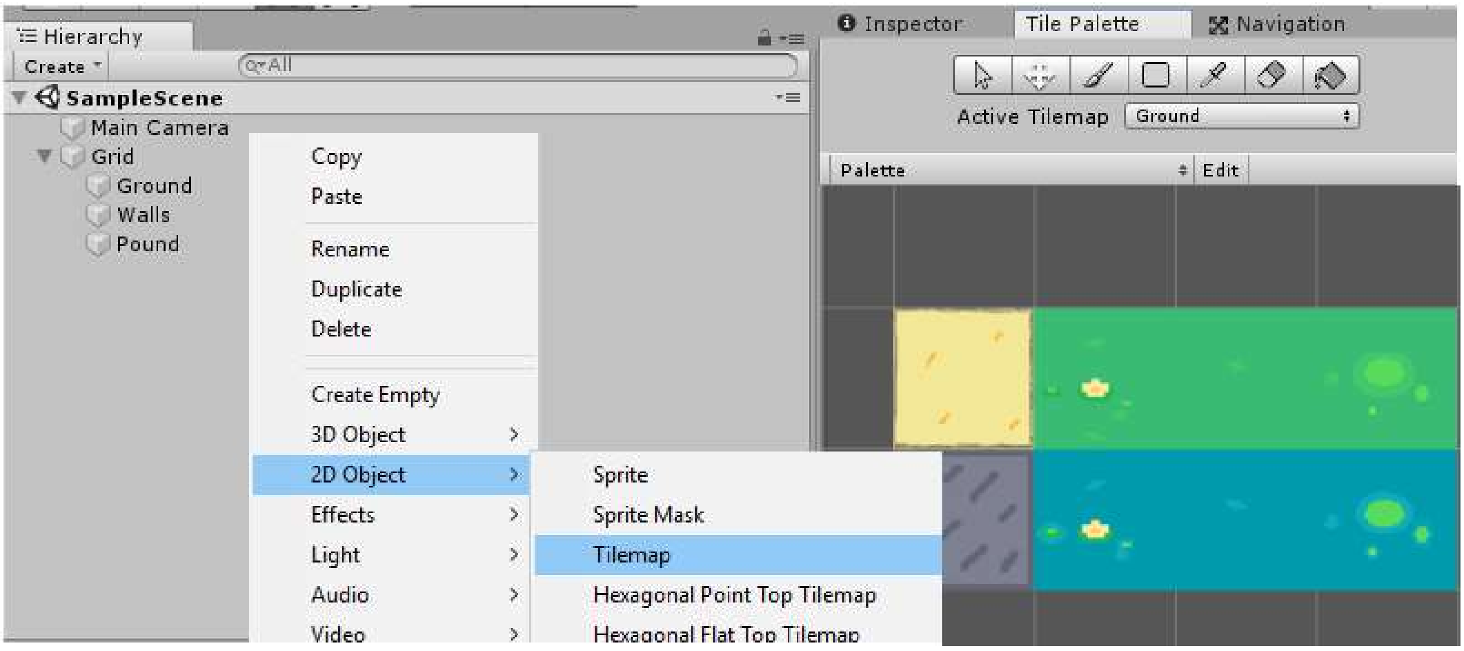
This short tutorial will focus on achieving the most basic implementation of NavMeshPlus.

To begin with, Create a new project from Unity's "2D Template".

Add a `Tilemap` GameObject to your scene (By right-clicking your Scene hierarchy and choosing the `2D Object` folder, and then `Tilemap` within). The first one you create should simultaneously create a `Grid` GameObject in the root of your scene and nest your new `Tilemap` underneath it. Add two more `Tilemap` s (for a total of 3) and and name them `Ground` , `Walls` and `Pond` .

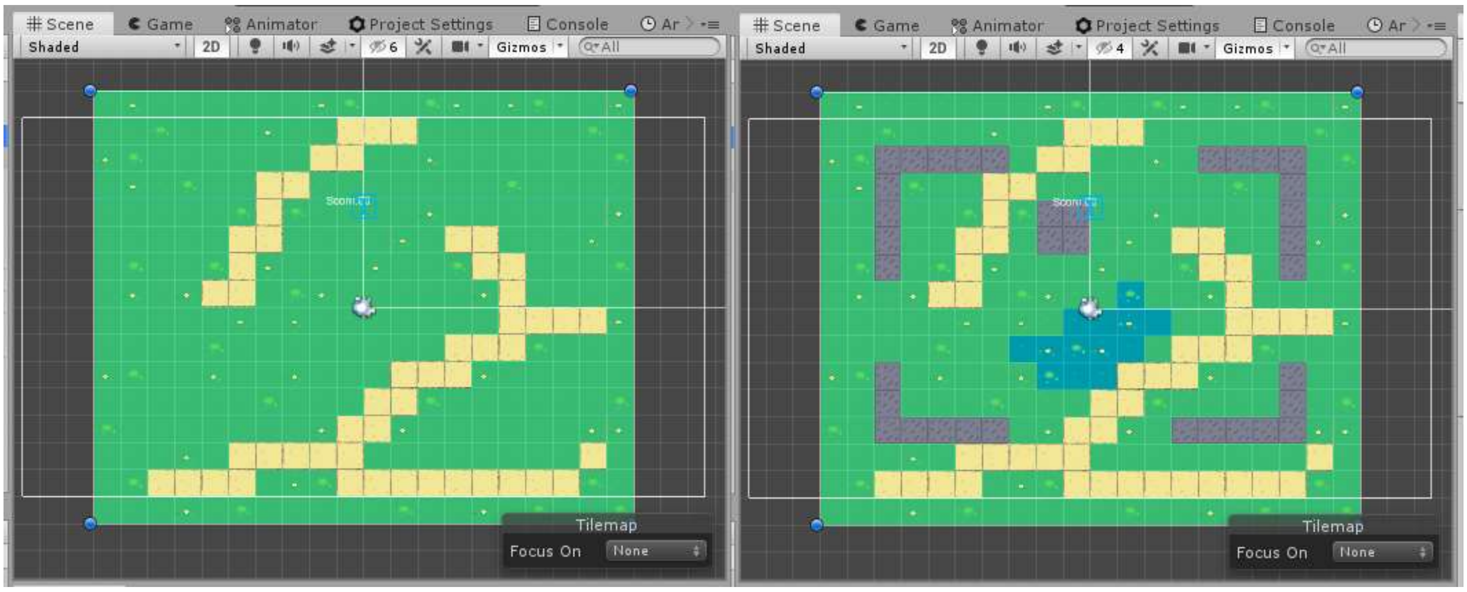
Import sprites that will be used as a palette. For example, you could just use 3 solid sold-colored sprites: a green color for ground, a black for walls, and blue for the pond. It's up to you.

Create a new palette and drag in your imported sprites.



Using the sprites imported into your palette, choose `Ground` as the "Active Tile Map", and then paint the ground into your seen any way you'd like. In our example, we're using the solid "green" brush.

Next, change "Active Tile Map" in your palette to `Walls`. Paint some obstacles. Then select `Pond` and do the same, drawing a body of water.



NAV MESH PLUS

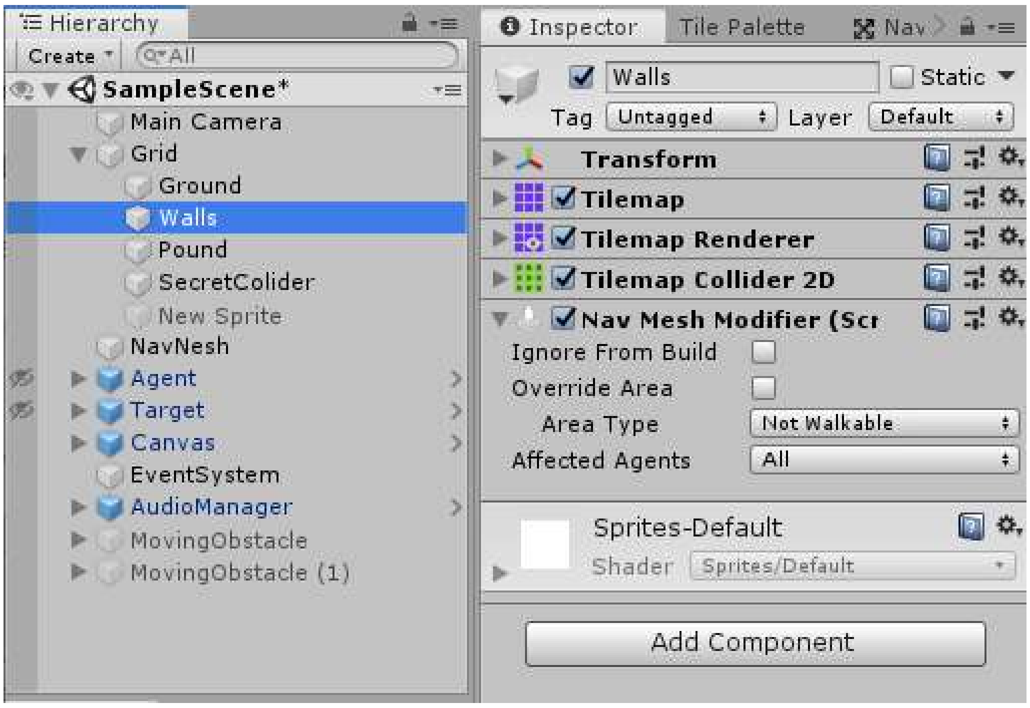
Download NavMeshPlus and copy NavMeshComponents into your Assets folder.

Now, in your scene, right click and choose `Create Empty` to add an empty GameObject into the root of your scene.

Name your new empty GameObject, name it "NavMesh", and rotate it `-90, 0, 0`.

With your `NavMesh` object selected, in the Inspector click `Add Component` and choose `Nav Mesh Surface 2d`.

Now select your `Walls` Tilemap GameObject (from within the `Grid` GameObject), and similarly, `Add Component` here, too. Choose a `Nav Mesh Modifier` component.



Hierarchy

SampleScene*
- Main Camera
- Grid
  - Ground
  - Walls
  - Pound
  - SecretColider
  - New Sprite
- NavMesh
- Agent
- Target
- Canvas
- EventSystem
- AudioManager
- MovingObstacle
- MovingObstacle (1)

Inspector

Walls — Static
Tag Untagged — Layer Default

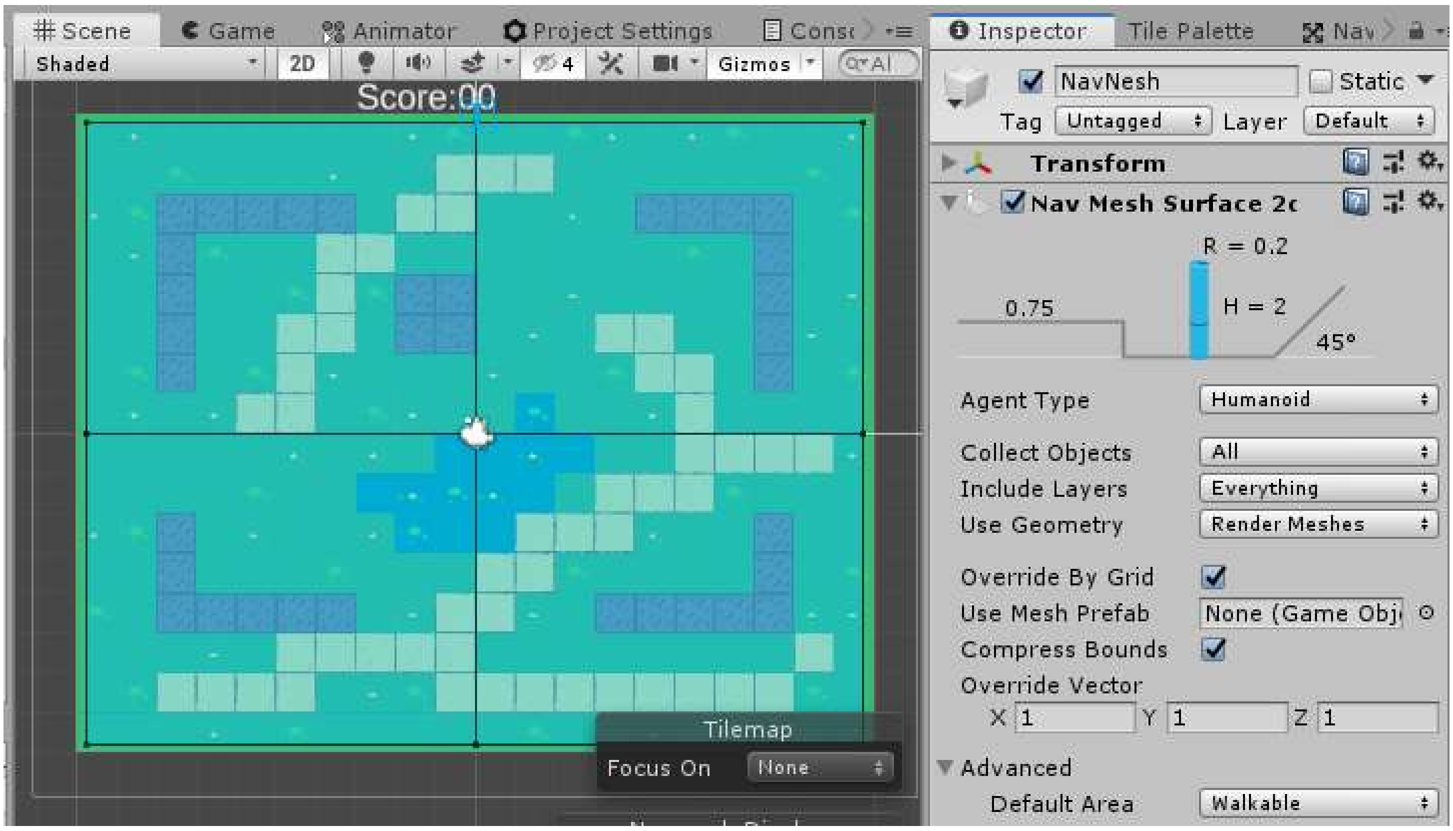
Transform
Tilemap
Tilemap Renderer
Tilemap Collider 2D
Nav Mesh Modifier (Scr

Ignore From Build	☐
Override Area	☐
Area Type	Not Walkable
Affected Agents	All

Sprites-Default
Shader Sprites/Default

Add Component

Go back to your `NavMesh` and, within the `Nav Mesh Surface 2d` component section of the inspector, hit "Bake".



Inspector

NavMesh — Static
Tag Untagged — Layer Default

Transform
Nav Mesh Surface 2c

R = 0.2
0.75
H = 2
45°

Agent Type	Humanoid
Collect Objects	All
Include Layers	Everything
Use Geometry	Render Meshes
Override By Grid	✓
Use Mesh Prefab	None (Game Obj
Compress Bounds	✓
Override Vector	X 1 Y 1 Z 1

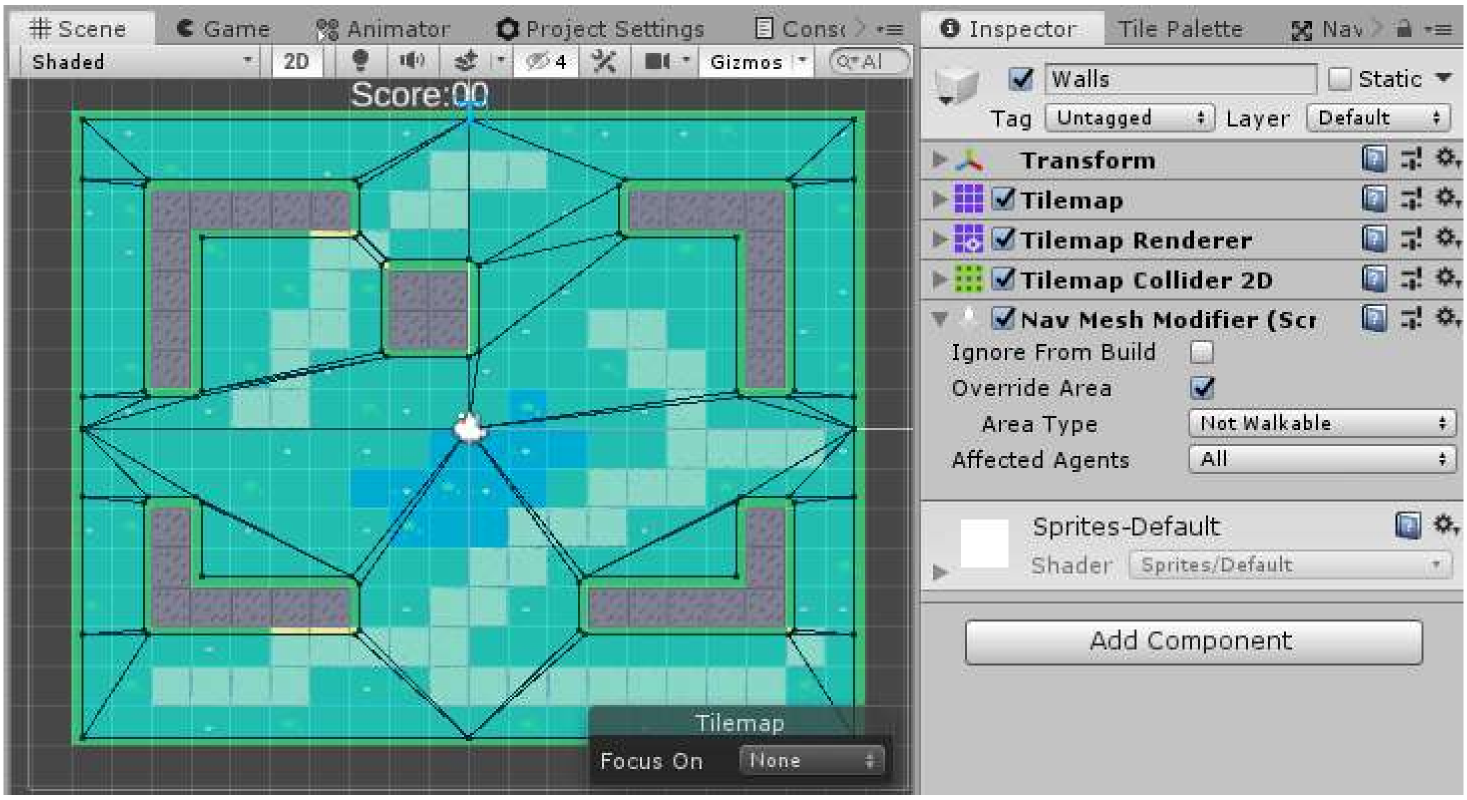
Advanced

Default Area	Walkable

As you may be able to see, clicking "Bake" caused the whole scene to be highlighted. Hence, the entire area is considered "walkable" at this point. Let's override it.

In the `Nav Mesh Modifier` component -- remember, that's on your `Walls` -- check "Override Area" and in the "Area Type" field, choose "Not Walkable".

Now hit "Bake" again (within the `NavMesh` object). You should see that the Walls you painted have been carved out of the highlighted areas. They are no longer considered "walkable".



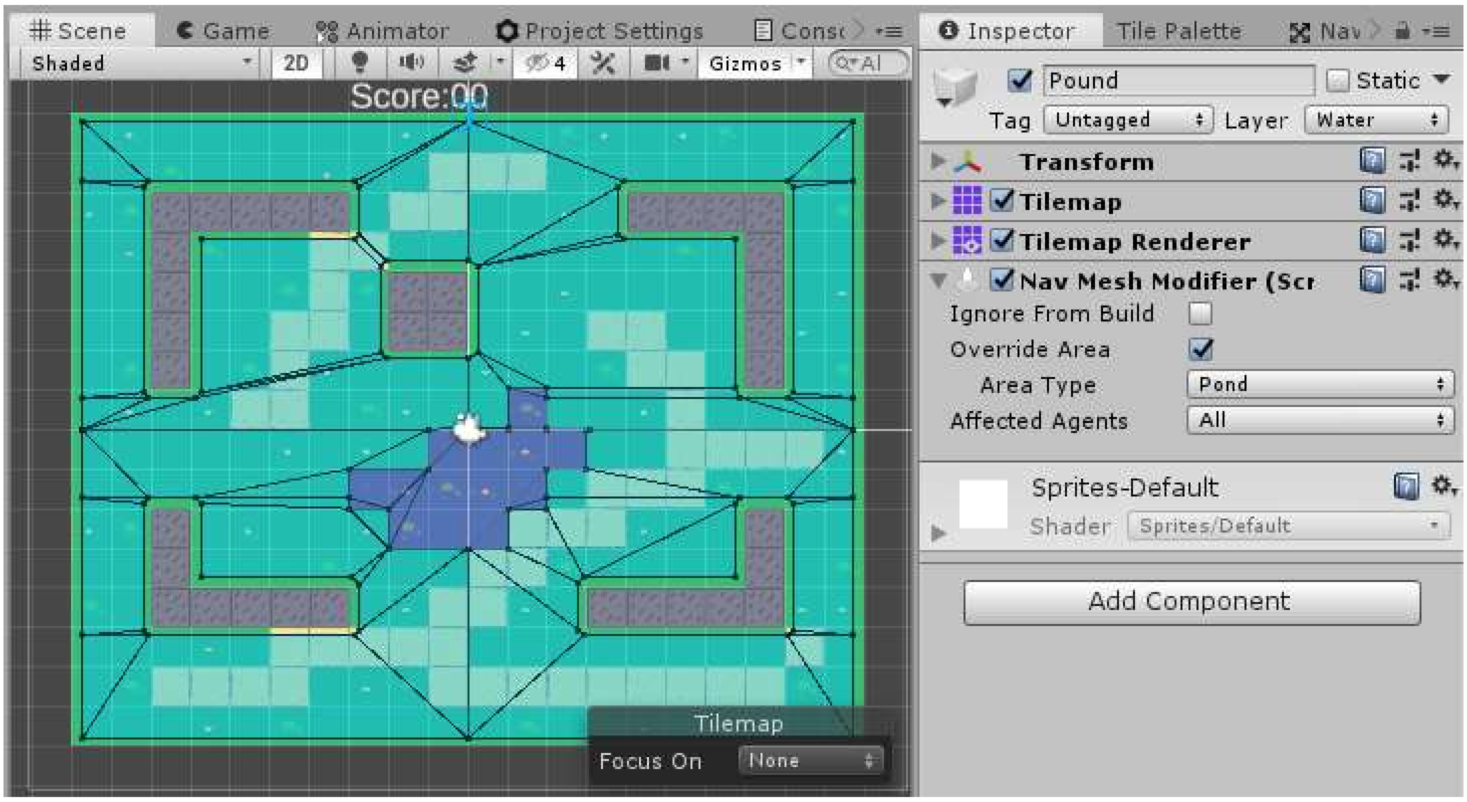
Walls ☐ Static
Tag Untagged Layer Default

Nav Mesh Modifier (Scr
Ignore From Build ☐
Override Area ✓
Area Type Not Walkable
Affected Agents All

Sprites-Default
Shader Sprites/Default

Now add a `Nav Mesh Modifier` Component to your `Pond` Tilemap. Override area like you did with Walls, and choose "Pond" for "Area Type".

Bake again, and see that the Pond is also carved out from your nav mesh.



Pound ☐ Static
Tag Untagged Layer Water

Nav Mesh Modifier (Scr
Ignore From Build ☐
Override Area ✓
Area Type Pond
Affected Agents All

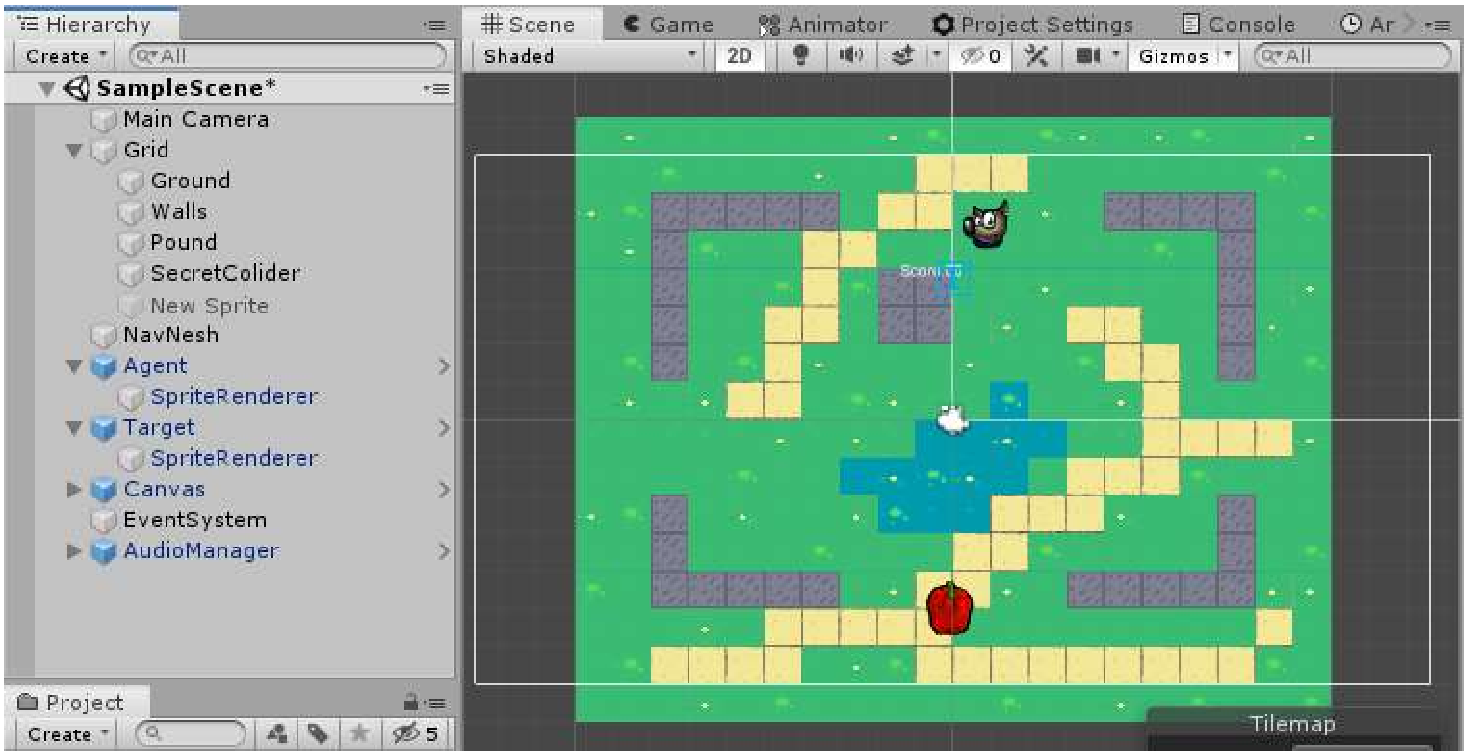
Sprites-Default
Shader Sprites/Default

At this point, you can add GameObjects with a `Nav Mesh Agent` component, which are capable of interacting with (i.e., walking on) your NavMesh.

Important note: When a "Nav Mesh Agent" starts to interact with your scene (e.g., when they start to move in-game) the system tends to rotate the GameObject upon which the component is placed. This can be undesirable (as in, your object may be rotate away from your plane of view and hence become invisible), so you should fix its rotation by adding something like the following to a new or existing script:

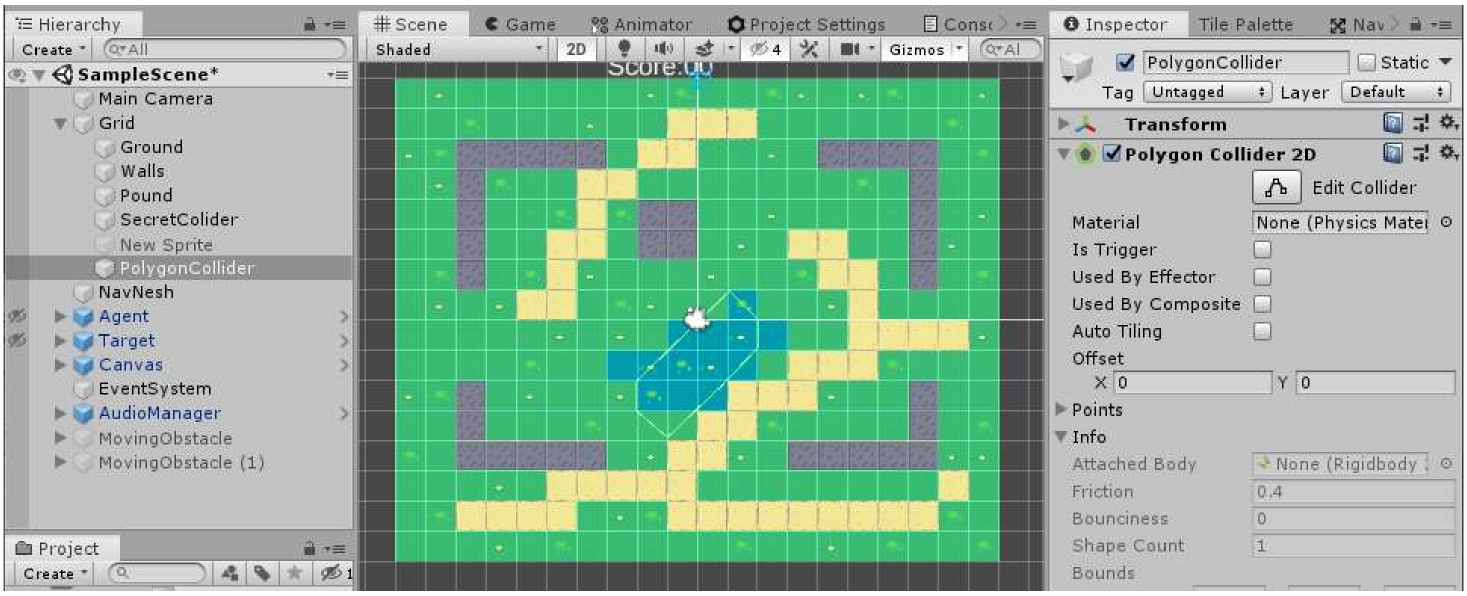
```
    void Start()    {
            var agent = GetComponent<NavMeshAgent>();
            agent.updateRotation = false;
            agent.updateUpAxis = false;
    }
```

Once this is added to the `Start` method for all of your agents, you are ready for action.

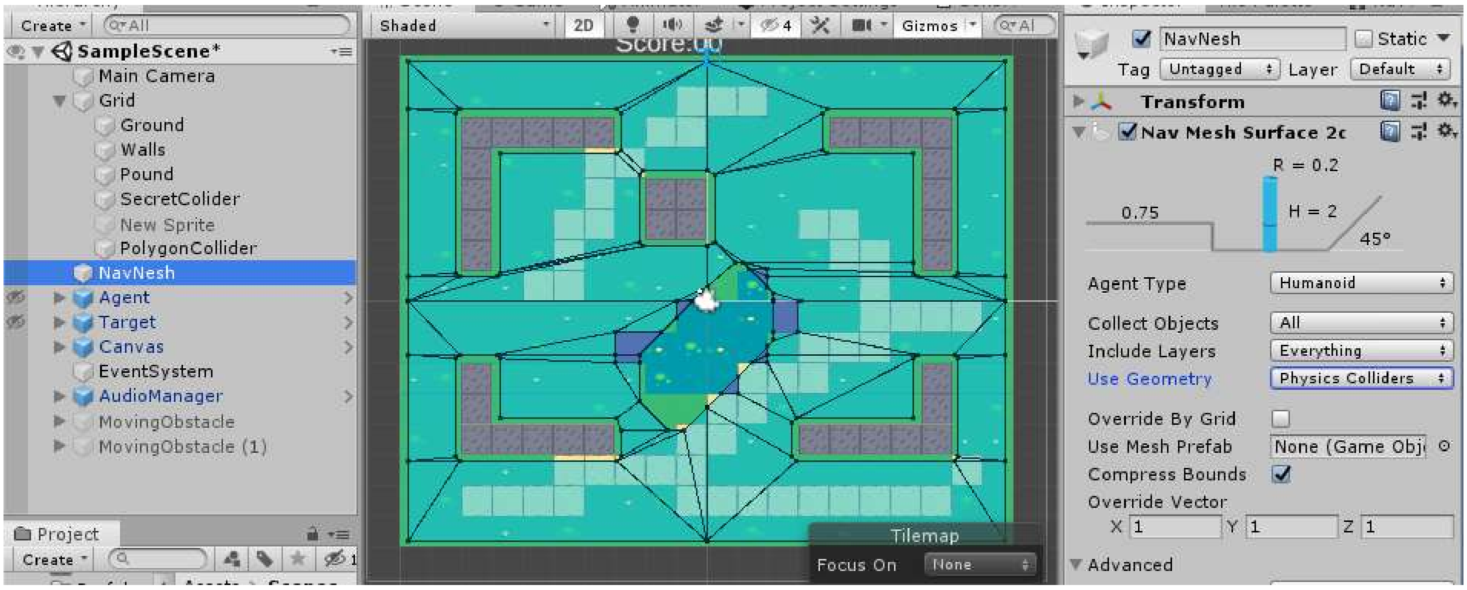


USED GEOMETRY

With new API introduced in Unity 2019.3 there is an ablitiy to choose geometry source. By default NavMeshSurface2d uses sprite Mesh to buld its navigation surface, so surface will stick to sprite "outline". But if you change "Use Geometry" option from "Render Meshes" to "Phisics Colliders", `NavMeshSurface2d` would build its surface form 2D colliders. Lets create new `PoundCollider` inside `Grid` object. Add `Polygon Collider 2d` and `NavMeshModifier` to it, override its area to "Pound". Now click "Edit Collider" to make desired pound's shape.



Select `NavMesh` object and change "Use Geometry" option from "Render Meshes" to "Phisics Colliders", click "Bake" and new Mesh will appear.

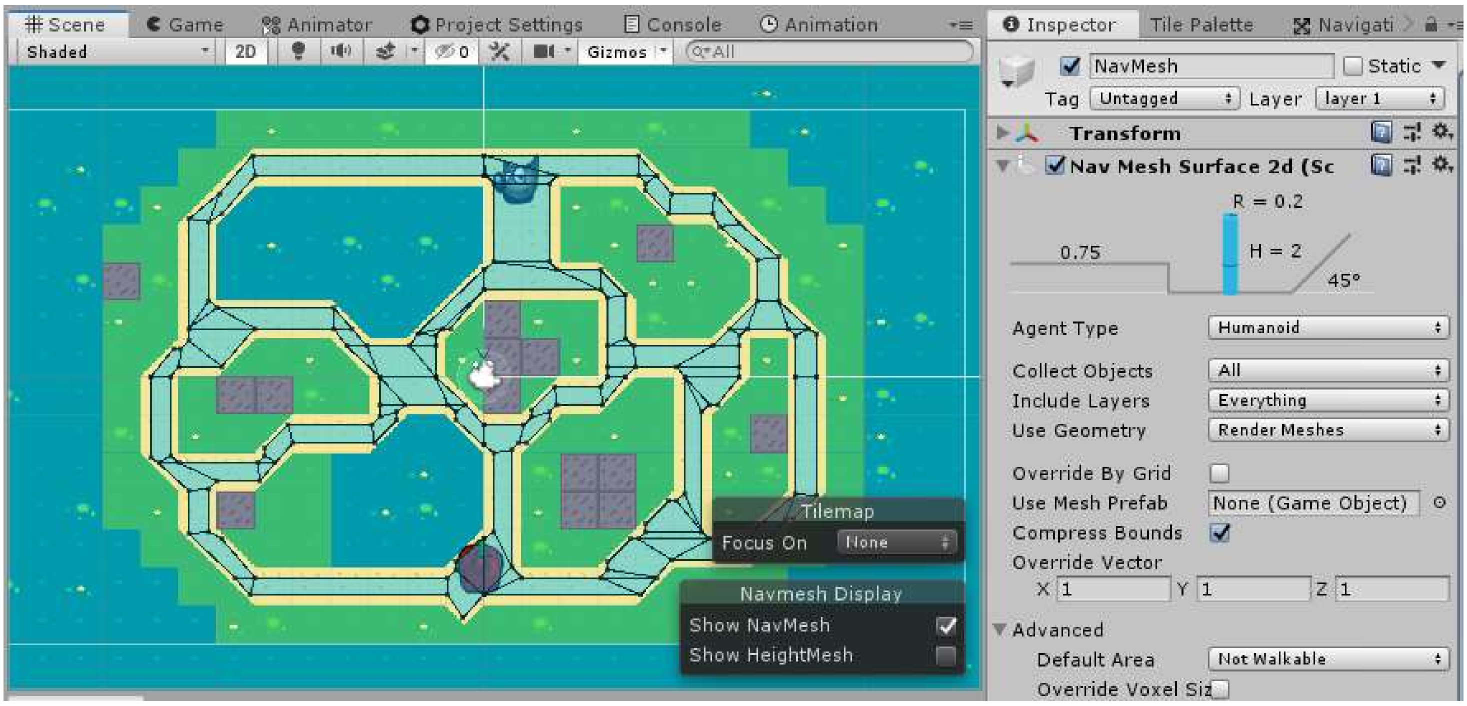


`Walls` stay the same, becase `Walls` object had composite collider with exact same box shape as sprite mesh. But pound now shaped not from tilemap sprites, but from `Plygon Collider 2D`.

BAKING A MAZE

Another common task is not to carve out obstacles, but to navigate through a maze, wherein our agents can only follow a strict path.

To do that, Bake your `NavMesh` with "Default Area" selected to "Not Walkable", and the `Nav Mesh Modifier` component set to be "Walkable", and you will get this:



NavMesh Static
Tag Untagged Layer layer 1
Transform
Nav Mesh Surface 2d (Sc
R = 0.2
0.75
H = 2
45°

Agent Type	Humanoid
Collect Objects	All
Include Layers	Everything
Use Geometry	Render Meshes
Override By Grid	☐
Use Mesh Prefab	None (Game Object)
Compress Bounds	✓
Override Vector	X 1 Y 1 Z 1

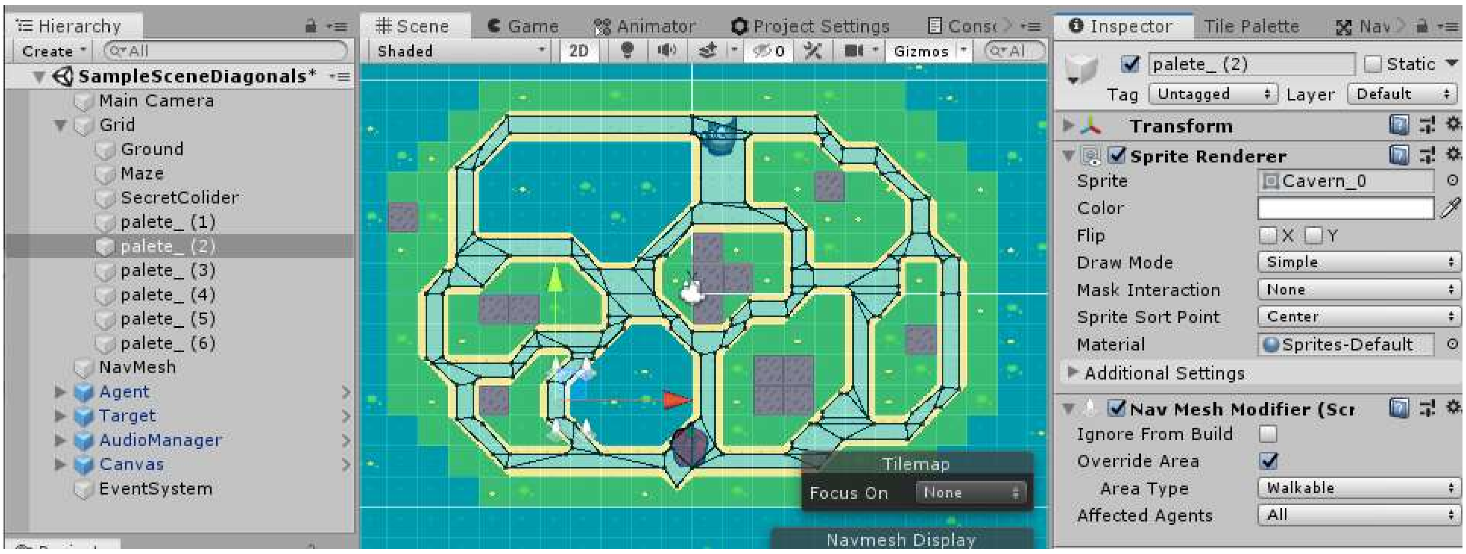
Advanced

Default Area	Not Walkable
Override Voxel Siz	☐

Tilemap
Focus On None

Navmesh Display
Show NavMesh ✓
Show HeightMesh ☐

Any sprite within the `Grid` object with a `Nav Mesh Modifier 2d` will be added into Bake process. (Sprite must have a mesh). You can build bridges or barricades, and these objects will be added into your `NavMesh` as static data.



SampleSceneDiagonals*
- Main Camera
- Grid
  - Ground
  - Maze
  - SecretColider
  - palete_ (1)
  - palete_ (2)
  - palete_ (3)
  - palete_ (4)
  - palete_ (5)
  - palete_ (6)
- NavMesh
- Agent
- Target
- AudioManager
- Canvas
- EventSystem

palete_ (2) Static
Tag Untagged Layer Default
Transform
Sprite Renderer

Sprite	Cavern_0
Color	
Flip	☐ X ☐ Y
Draw Mode	Simple
Mask Interaction	None
Sprite Sort Point	Center
Material	Sprites-Default

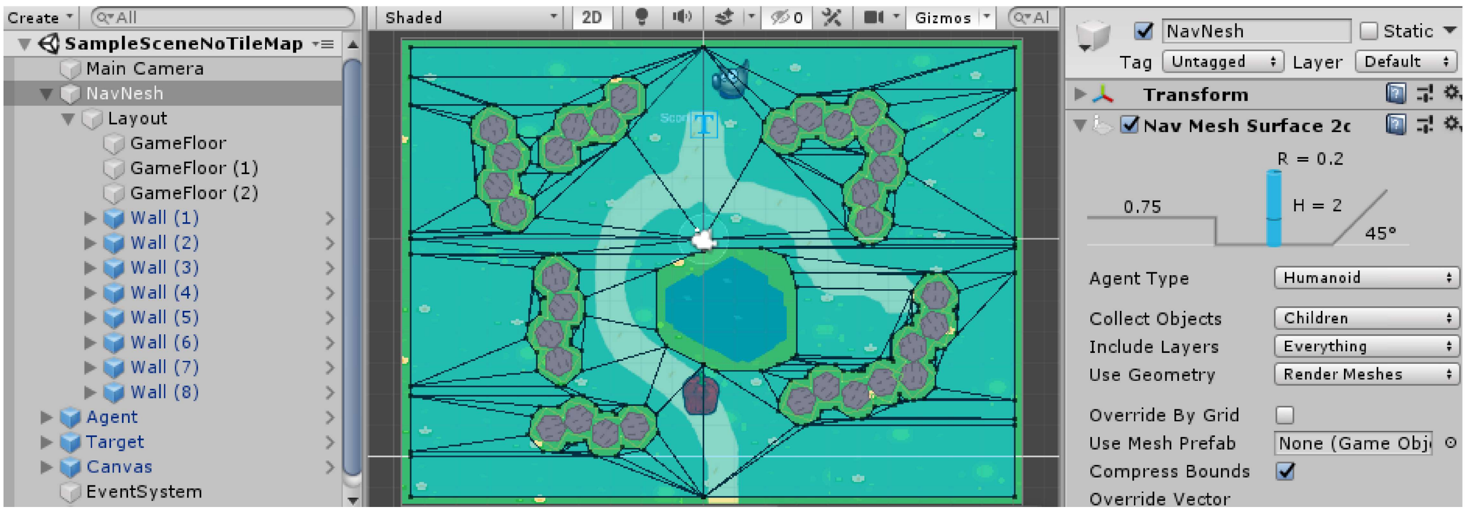
Additional Settings
Nav Mesh Modifier (Scr

Ignore From Build	☐
Override Area	✓
Area Type	Walkable
Affected Agents	All

Tilemap
Focus On None
Navmesh Display

NON-TILED

`NavMeshSurface2d` can bake objects that are not a part of `Tilemap`, or even if there is not `Tilemap`. To do that place all `GameObject` s with `NavMeshModifier` that define your layout under object with `NavMeshSurface2d`, and select appropriate option `Collect Objects - Children`



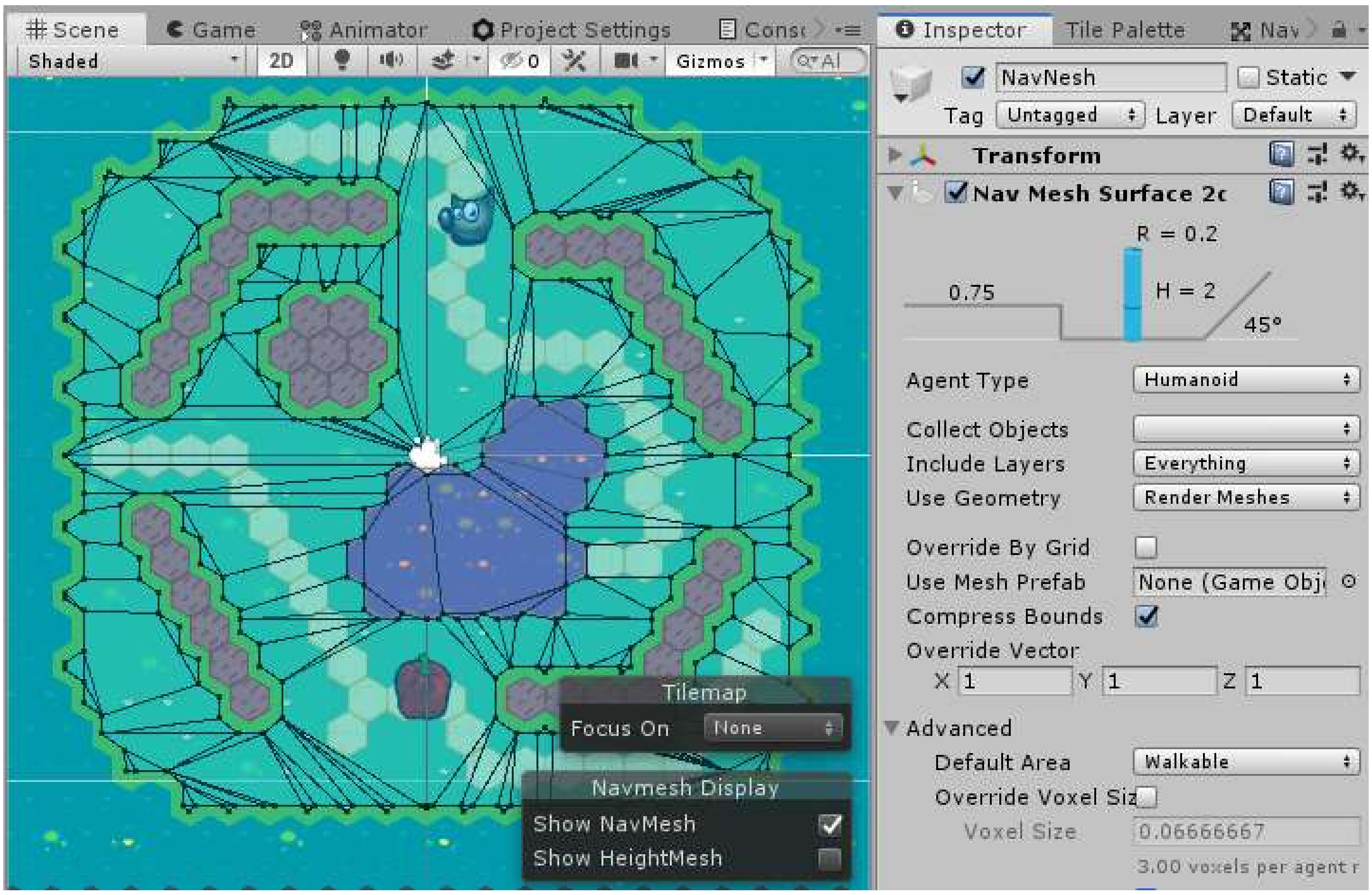
SampleSceneNoTileMap
- Main Camera
- NavNesh
  - Layout
    - GameFloor
    - GameFloor (1)
    - GameFloor (2)
  - Wall (1)
  - Wall (2)
  - Wall (3)
  - Wall (4)
  - Wall (5)
  - Wall (6)
  - Wall (7)
  - Wall (8)
- Agent
- Target
- Canvas
- EventSystem

NavNesh Static
Tag Untagged Layer Default
Nav Mesh Surface 2c
R = 0.2
0.75 H = 2 45°

Agent Type	Humanoid
Collect Objects	Children
Include Layers	Everything
Use Geometry	Render Meshes
Override By Grid	☐
Use Mesh Prefab	None (Game Obj
Compress Bounds	✓
Override Vector	

OTHER GRIDS

Some games want to use hexagonal, isometric, or diamond-shaped grids. NavMeshPlus can be done in other types of grid as well as long as the spites share the shape of your grid.



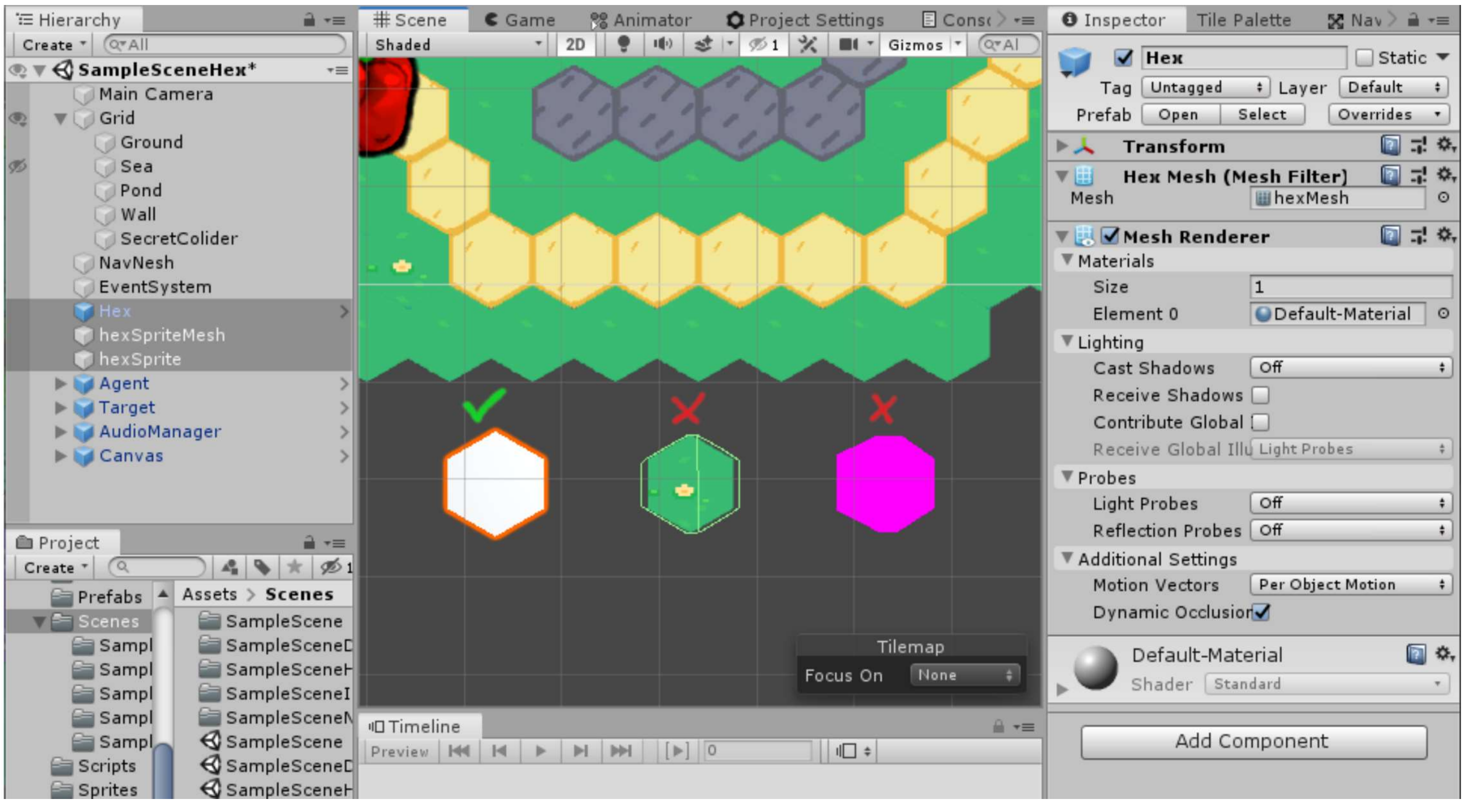
Tilemap
Focus On None

Navmesh Display
Show NavMesh ✓
Show HeightMesh ☐

NavMesh Static
Tag Untagged Layer Default
Nav Mesh Surface 2c
R = 0.2
0.75 H = 2 45°

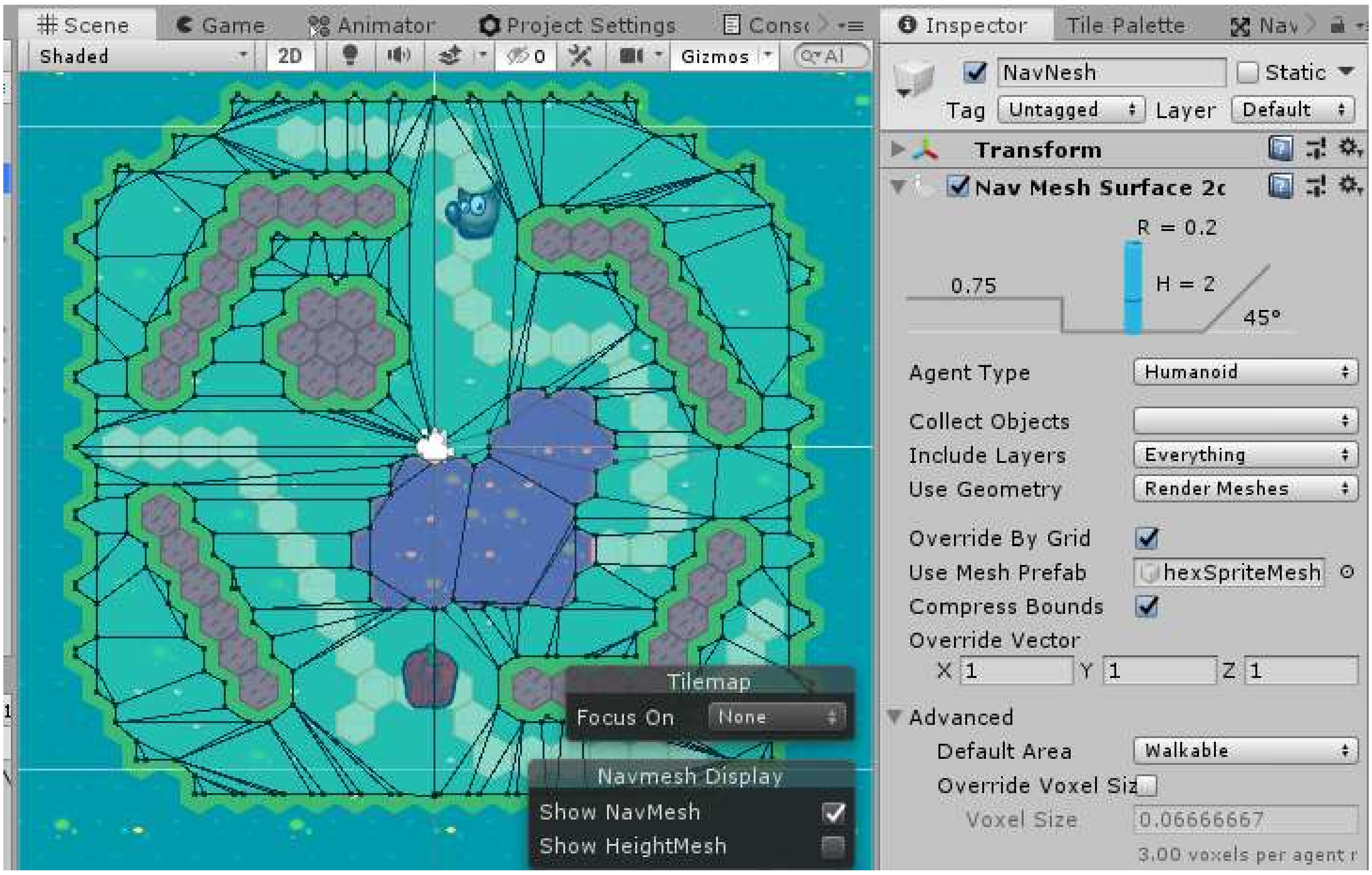
Agent Type	Humanoid
Collect Objects	
Include Layers	Everything
Use Geometry	Render Meshes
Override By Grid	☐
Use Mesh Prefab	None (Game Obj
Compress Bounds	✓
Override Vector	X 1 Y 1 Z 1
Advanced	
Default Area	Walkable
Override Voxel Siz	☐
Voxel Size	0.06666667
	3.00 voxels per agent r

But in some cases, the sprite can differ from your grid shape, but we still want clean navigation though the tilemap.



Hex Static
Tag Untagged Layer Default
Prefab Open Select Overrides
Transform
Hex Mesh (Mesh Filter)
Mesh hexMesh
Mesh Renderer
Materials
Size 1
Element 0 Default-Material
Lighting
Cast Shadows Off
Receive Shadows ☐
Contribute Global ☐
Receive Global Illu Light Probes
Probes
Light Probes Off
Reflection Probes Off
Additional Settings
Motion Vectors Per Object Motion
Dynamic Occlusion ✓
Default-Material
Shader Standard

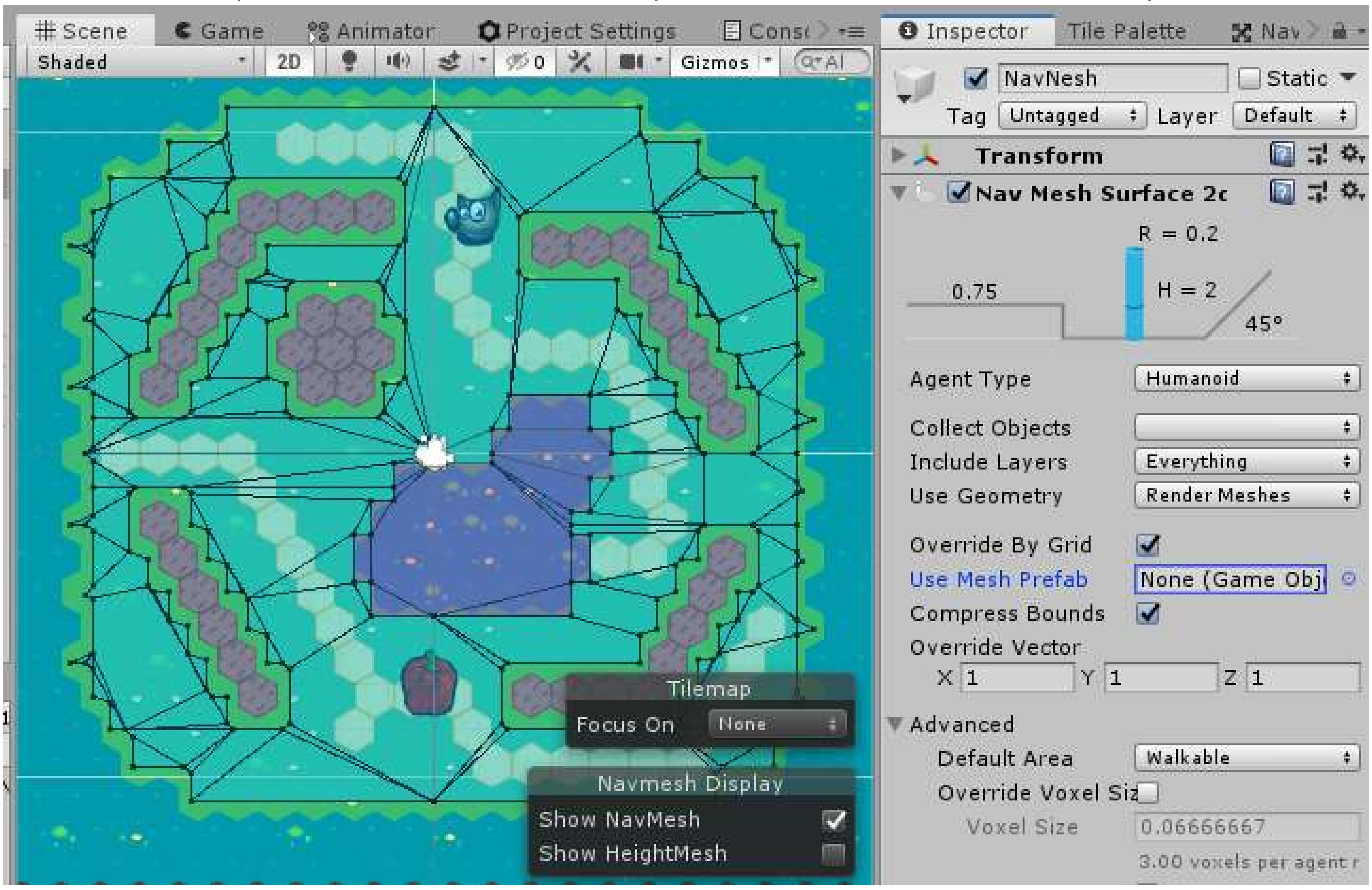
To do that, we will override all tiles within a grid. First we need to have a mesh that has the form of a tile. For this hexagonal tile example, we'll import a mesh shaped like a hexagon. (Use Blender or any other 3D software) Add your hexagon to the scene and rotate it to face the camera, and scale it down to match the size of your cells. In the `Nav Mesh Surface 2d` component, check "Override By Grid" and all tiles will have a shape of used prefab. Even if this option disabled, all sprites without a proper mesh will be substituted with used one.



NavNesh Static
Tag Untagged Layer Default
Transform
✓ Nav Mesh Surface 2d

```
Agent Type            Humanoid
Collect Objects
Include Layers        Everything
Use Geometry          Render Meshes
Override By Grid      ✓
Use Mesh Prefab       hexSpriteMesh
Compress Bounds       ✓
Override Vector
  X 1   Y 1   Z 1
Advanced
  Default Area        Walkable
  Override Voxel Siz
  Voxel Size          0.06666667
                      3.00 voxels per agent r
```

If we remove the prefab and leave the override option, all tiles will be considered as squares.

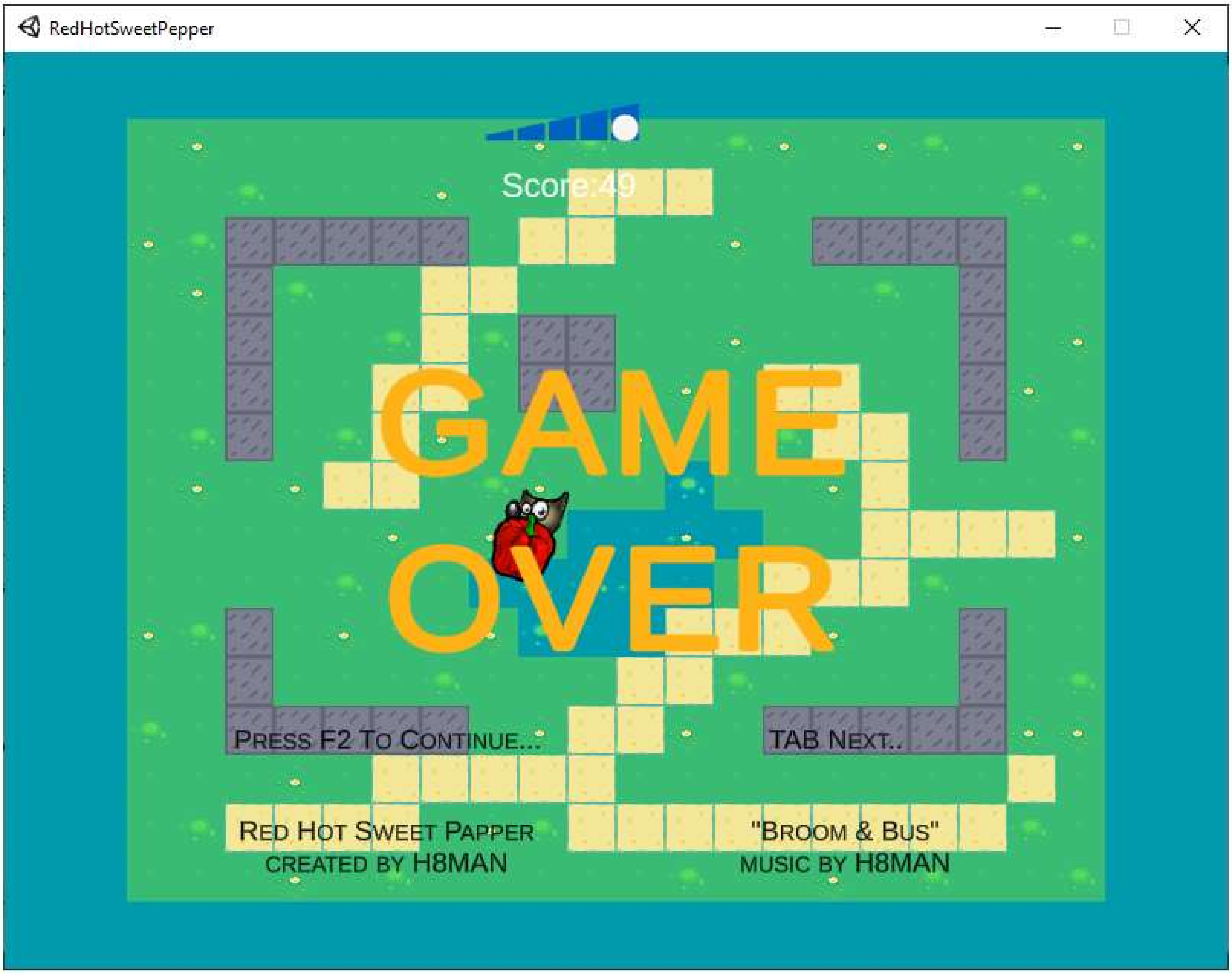


NavNesh Static
Tag Untagged Layer Default
Transform
✓ Nav Mesh Surface 2d

```
Agent Type            Humanoid
Collect Objects
Include Layers        Everything
Use Geometry          Render Meshes
Override By Grid      ✓
Use Mesh Prefab       None (Game Obj
Compress Bounds       ✓
Override Vector
  X 1   Y 1   Z 1
Advanced
  Default Area        Walkable
  Override Voxel Siz
  Voxel Size          0.06666667
                      3.00 voxels per agent r
```

THE END

Try out a fully-featured 2D "run-and-chase" game I made to demo NavMesh2d: https://github.com/h8man/RedHotSweetPepper.



References:

1. NavMeshSurface - https://docs.unity3d.com/Manual/class-NavMeshSurface.html
2. NavMeshSurface2d - https://github.com/h8man/NavMeshPlus
3. The Game Demo - https://github.com/h8man/RedHotSweetPepper
4. Forum to discuss - https://forum.unity.com/threads/2d-navmesh-pathfinding.503596/