

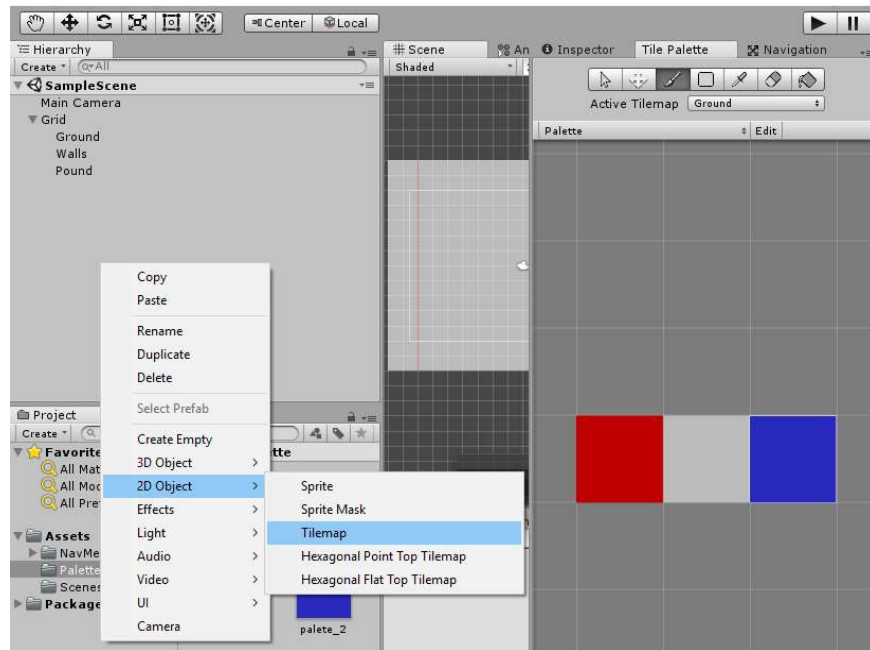
NAVMESH PLUS

HOW TO

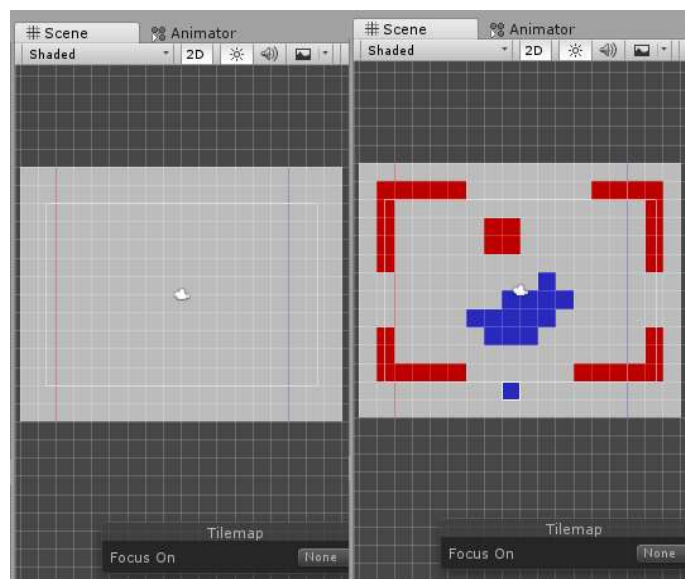
by h8man

Launch Unity and create new project from 2D Template. Now copy NavMeshComponents into your asset folder.

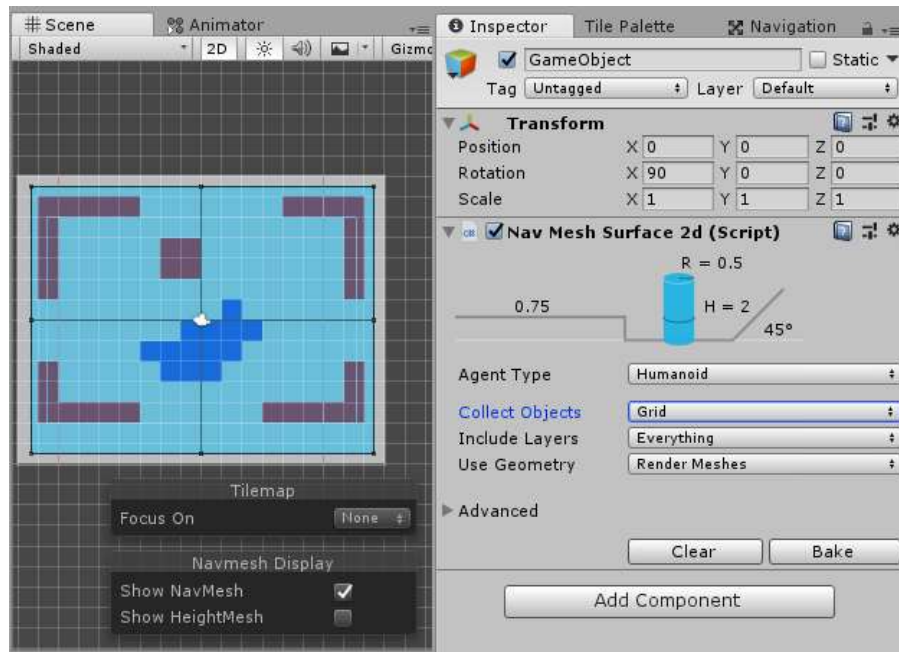
Add Tilemap to the scene, add three of them and call Ground, Walls and Pond. Import sprites that will be used as palette. For example 3 solid colored sprites. Create new palette and drag your imported sprites



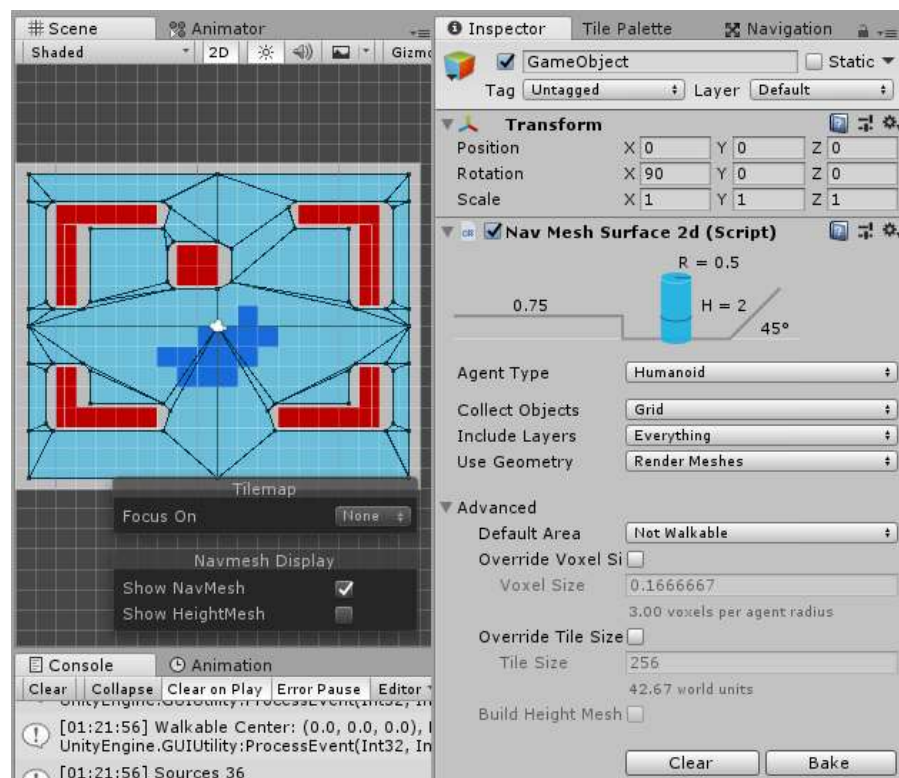
Paint your ground in any mean. For example by solid white brush. Change "Active Tile Map" in palette to "Walls", paint some obstacles, than select "Pond" and draw body of water



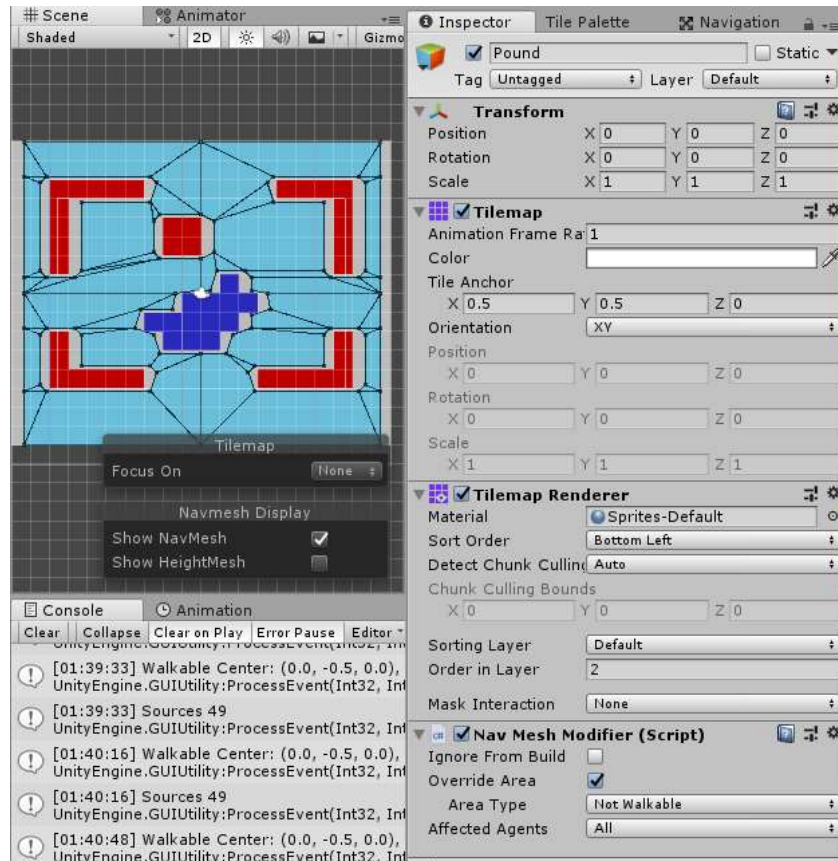
Now add EmptyObject in the root, rotate (-90;0;0). Add NavMeshSurface2d Component, select CollectionSource to Grid and hit bake.



As you see all area is walkable. Lets override it, add TilemapCollider2D component to "Walls" Tilemap, select EmptyObject and unfold Advanced, select there Default Area - Not Walkable and hit bake. Collidable areas will be carved out



Now add NamMeshModifier Component to "Pond" Tilemap, override area as Not Walkable, and bake again. Now pond is also carved out form mesh.



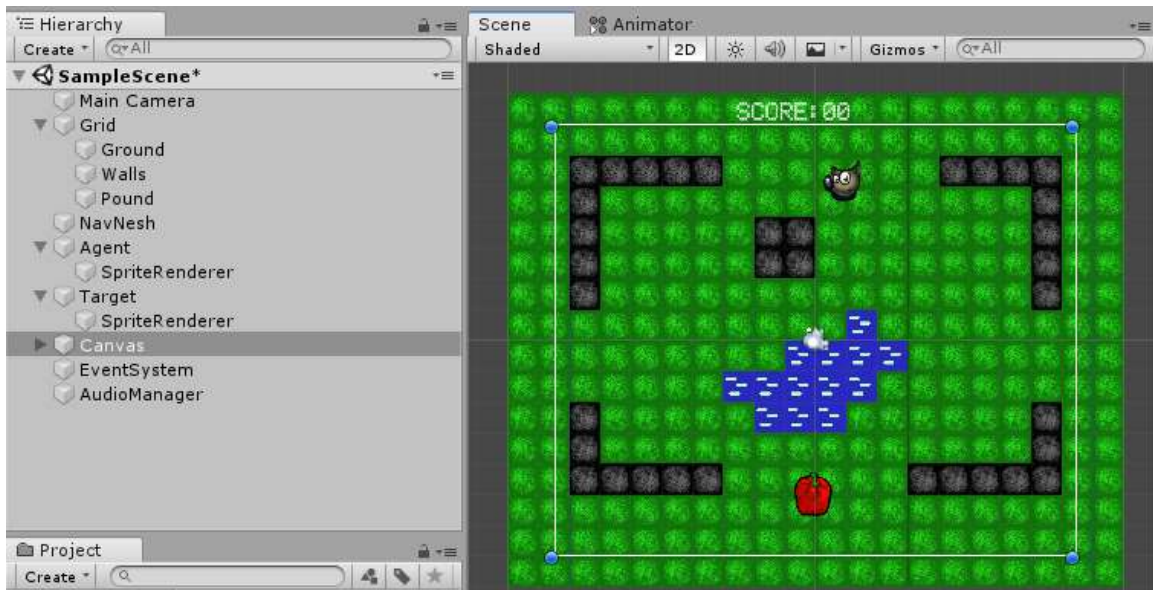
Add NavMeshAgents and details, be aware that agent tend to rotate game object, that can be undesirable, so you should fix its rotation:

Code (CSharp):

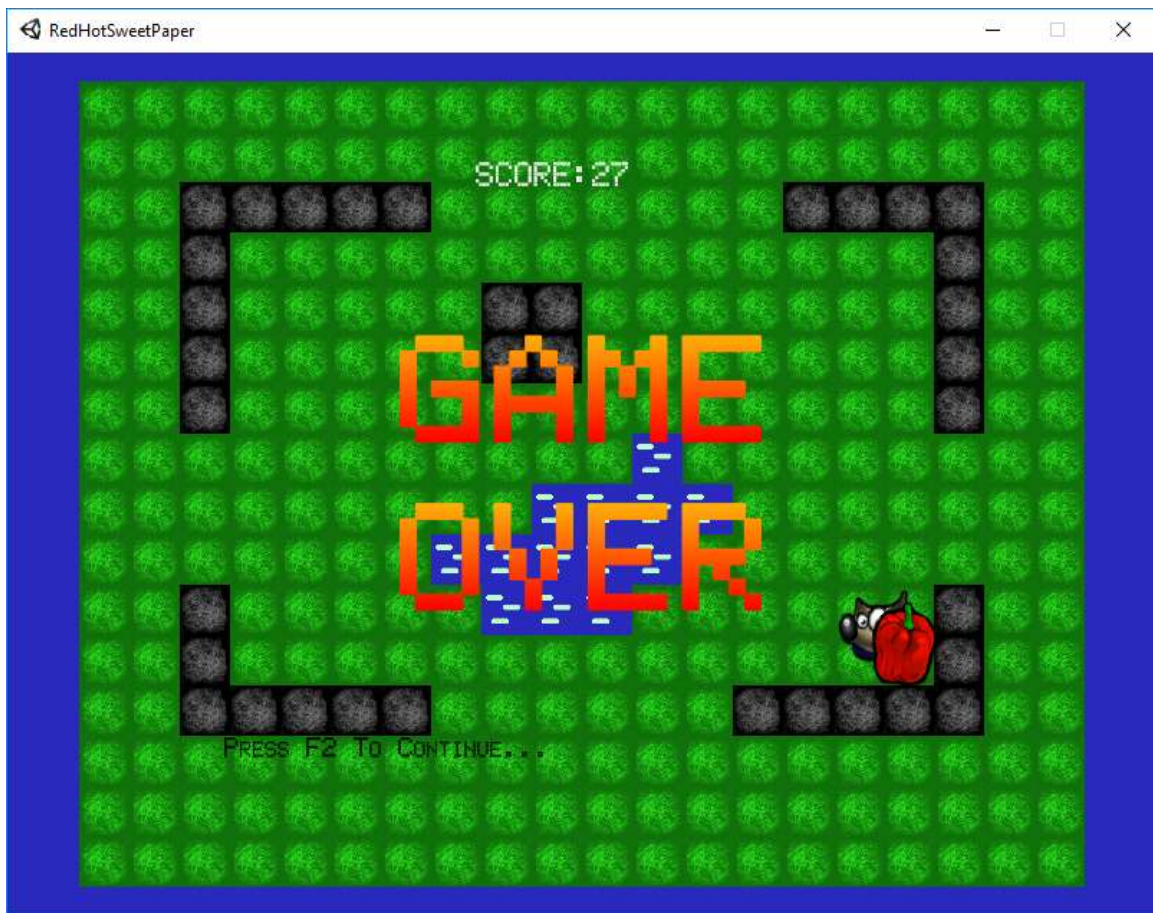
```
private NavMeshAgent agent;

// Start is called before the first frame update
void Start()
{
    agent = GetComponent<NavMeshAgent>();
    agent.updateRotation = false;
    agent.updateUpAxis = false;
}
```


Add this on Start for every of agents, and you ready to chase with obstacle avoidance



Enjoy "fully featured" 2D run and chase game "RedHotSweetPeper"



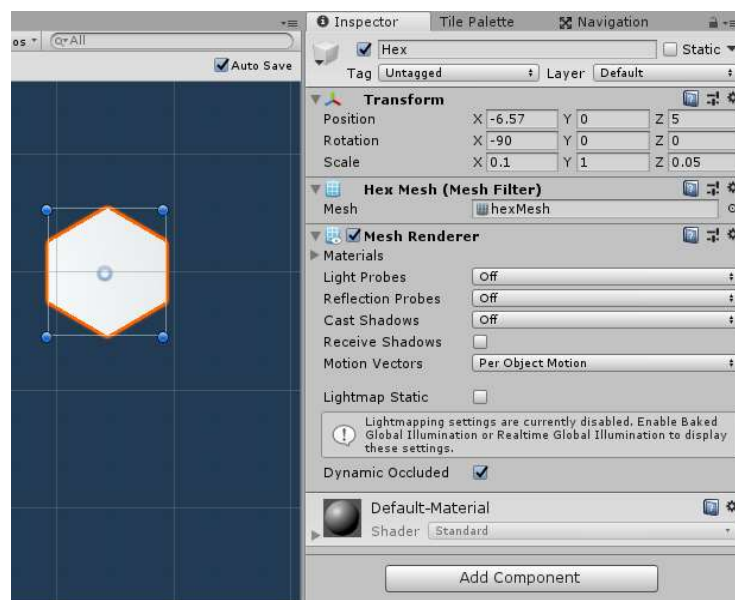
Custom Grid

Some games want to use hexagonal or isometric grid, or even diamond grid. NamMesh can be done to other types of grid as well, but with some work. To do that we need to have a mesh that has a form of a tile.

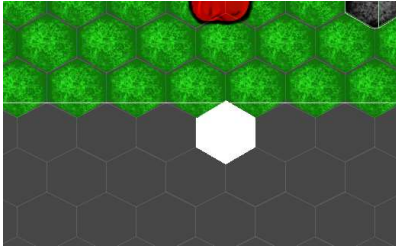
Lets create a hexagonal tile map, create new scene and do all previous steps, but now with hexagons:



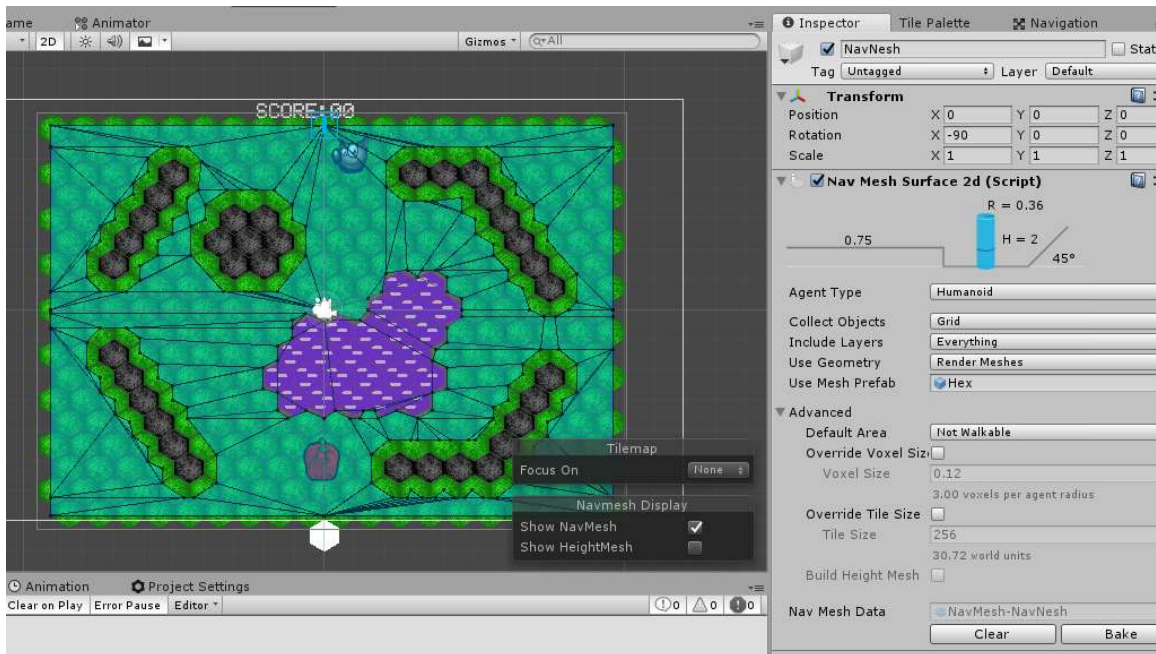
Now import mesh for hexagon tile (use blender or any other 3D software):



Add your hexagon to the scene and rotate it to face, and scale down to cell size:



The last step is to inform what shape tiles do have, feed our hex object to NavMeshSurface2d filed "Use Mesh Prefab" and hit bake.



THE END

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/>