Step 1: Create Project

Create a new unity 3D project and name it Click to Move

Step 2: add a floor

Once you created a scene go to the gameobject panel and add a 3D plane, extend it a bit and attach a new material to the floor, make the material a light grey so the floor stands out from the game objects

Step 3: Add Player and Obstacles

Add 4 3D gameobjects, the first one being a capsule with a material attached to it, make that material Green; afterwards rename it “Player”. Next add 3 cubes and space them out around the plane, these will be the obstacles in your level, rename them to “Obstacle”

Step 4: Nav Mesh

Add a Nav mesh onto the player object, make sure that the radius and height of the nav mesh are set correctly to the size of the object.

Then after that click the floor and open the navigation window, it will be in the window panel, under AI.

Once you open it, tick the Navigation Static box and set the floor to Walkable, after that go to bake and just click the bake button.

Once it's done you will see a blue area on the floor which indicates the area the player can move on.

Step 5: Set Obstacles

Select all the obstacles on your scene and go to the navigation tab once more.

Select the Navigation Option again and turn it on, then change the Navigation area to Not Walkable.

Afterwards go back into the Bake function and click bake, it will be done when you see grey spaces under the blocks.

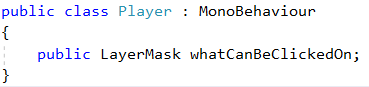
What this does is it makes it so the player cannot walk on those spaces.

Step 6: Player Movement

Create a new script called player and drag it onto the player object.

Open the script and after its opened delete the start and update function so you have a clean canvas.

Afterwards create a public Layer Mask and call it “whatCanBeClickedOn” what this will do is it will give us specific layers to our player that we can click on in order to move them



Step 7: Nav Mesh control

In order to access the NavMesh on our player we need to specify it on our script

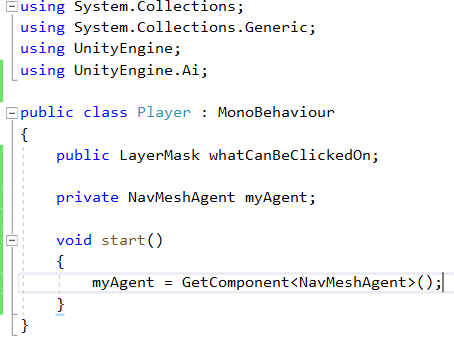
At the top of the scrips add a UnityEngine.AI.

After that it will give us access to use the NavMesh, create a new private variable type NavMeshAgent and call it “myAgent”

After that create a new start function and set my agent equal to GetComponent of type NavMeshAgent.

What this will do is it will access the NavMeshAgent that was added to our player at the start.

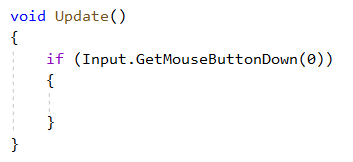
The script should look like this after the changes.



Step 8: Button Input

To check if you’ve pressed down the right mouse button, we need to create underneath everything a new update function.

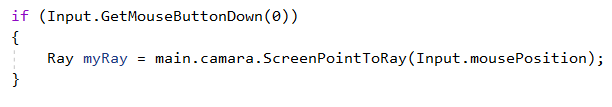
Inside the function set an If statement that checks if the Input of GetMouseButtonDown is passing 0, this means that if your mouse button is pressed then it will work.



Step 9: Ray Casting

Inside the If statement create a variable of the type Ray and call it myRay, set that equal to the main camera and set the ray with a ScreenPointToRay statement and in brackets set an input onto the mouse position.

What all this means is, it's making an invisible line from the view of the camera directly onto where your mouse is.

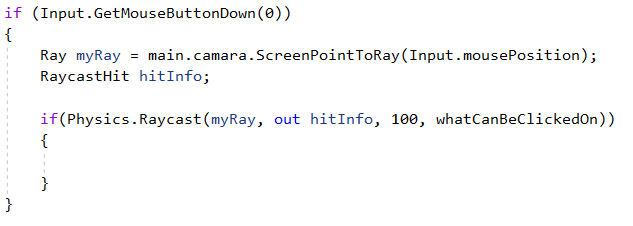


After that create a RayCastHit variable called hitInfo, this will contain all the information of what the ray hits.

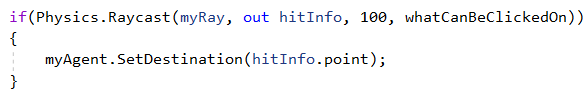
Now to actually cast the ray we have to create a new variable starting with an If statement, inside the statement set a Physics.Raycast variable and then open parentheses nest to it and write “myRay” so the beam from the camara to the position on the screen, the ray info variable (hitInfo) but make sure to put the word out before it so the function will feed the variable with the correct information.

Then set a large number for the ray size so for example 100 and lastly pass the layer mask variable (whatCanBeClickedOn) so that it only detects objects from said layer.

An important factor to note is that in order to specify all these together they must be separated by a coma.

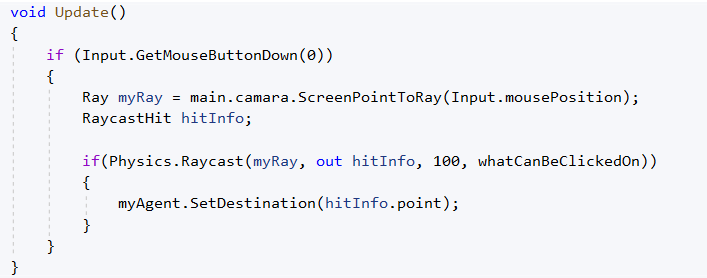


And Lastly inside that if statement add myAgent.SetDestination, which will make the player move to set destination and in parenthesis next to it add hitInfo.point which will specify the point itself where you hit it.



This last line is very important as it is the one that will make your player move.

In the end the entire section of code will look like this.



Step 10: Unity Changes

Return to Unity to apply the options that will make the script work.

First go onto the layer panel on the right after you click the floor, select the layer button and click on add new layer.

Set the 8th layer to be called “Ground” then apply that layer to the floor.

Then lastly select your player and scroll down to the script, and in the area that said What can be clicked, select the ground layer.

Completed Tutorial

If you want to modify the speed of the player, you can do that in the Nav Mesh component.

This tutorial was taken from BlackThornProd’s Tutorial on click to move

Link: <https://www.youtube.com/watch?v=KU2CKBlCAxQ&t=131s>