Table of Contents

An Introduction *2*
What are Nodes and Paths? *2*
How to create Nodes and Paths for 2D *3*
- Step 1: Create Pathfinder 3
- Step 2: Select Pathfinder 3
- Step 3: Create nodes 4
- Step 4: Create Paths (Connections) 4
- Step 5 (Optional): Enter costs for each path 5

How to create Nodes and Paths for 3D *5*
- Step 1: Create Pathfinder 5
- Step 2: Select Pathfinder 6
- Step 3,4,5: 6

How to fetch the shortest path: *6*
Example of how to fetch the shortest path: *8*
How to traverse through the Path (Optional): *9*
Caution with 3D node creation: *10*
FAQ *12*
1. I cannot see Nodes or Paths in the scene 12
2. To all the users using Free-Version 12
3. While trying to Create nodes, I am clicking on the Collider, but nodes are not created: 12
5. Logging and other debugging 13

(Most common questions are answered in the FAQ below)

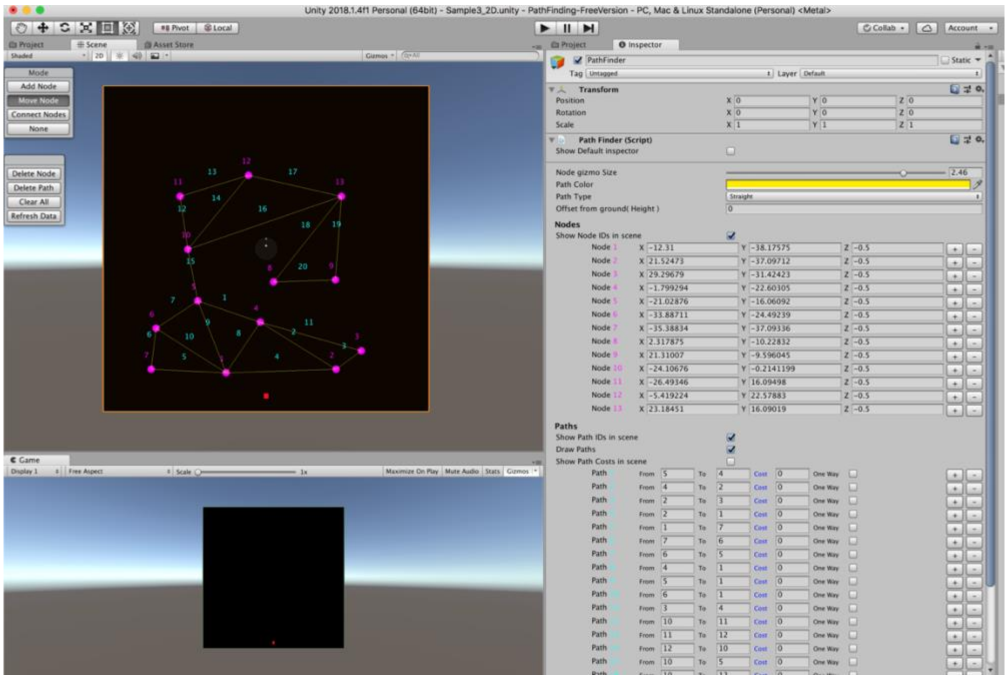
An Introduction

QPathFinder uses A* Path-finding algorithm. This algorithm is best known for finding the shortest path is minimal time.

Kindly Note, this plugin is not a grid-based pathfinder, but a node-based pathfinder.

You have to set up multiple *Nodes* and inter-connect them like in the image below. Once you have that set up properly, this algorithm will provide the shortest path from one point on the *Ground** to another.

Ground* - is a collider, on which we can add nodes.



PathFinder — Static
Tag: Untagged — Layer: Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

Path Finder (Script)
Show Default inspector ☐

Node gizmo Size: 2.46
Path Color: 🟨
Path Type: Straight
Offset from ground(Height): 0

Nodes
Show Node IDs in scene ✓

	X	Y	Z
Node 1	-12.31	-38.17575	-0.5
Node 2	21.52473	-37.09712	-0.5
Node 3	29.29679	-31.42423	-0.5
Node 4	-1.799294	-22.60305	-0.5
Node 5	-21.02876	-16.06092	-0.5
Node 6	-33.88711	-24.49239	-0.5
Node 7	-35.38834	-37.09336	-0.5
Node 8	2.317875	-10.22832	-0.5
Node 9	21.31007	-9.596045	-0.5
Node 10	-24.10676	-0.2141199	-0.5
Node 11	-26.49346	16.09498	-0.5
Node 12	-5.419224	22.57883	-0.5
Node 13	23.18451	16.09019	-0.5

Paths
Show Path IDs in scene ✓
Draw Paths ✓
Show Path Costs in scene ☐

	From	To	Cost	One Way
Path	5	4	0	☐
Path	4	2	0	☐
Path	2	3	0	☐
Path	2	1	0	☐
Path	1	7	0	☐
Path	7	6	0	☐
Path	6	5	0	☐
Path	4	1	0	☐
Path	5	1	0	☐
Path	6	1	0	☐
Path	3	4	0	☐
Path	10	11	0	☐
Path	11	12	0	☐
Path	12	10	0	☐
Path	10	5	0	☐

What are Nodes and Paths?

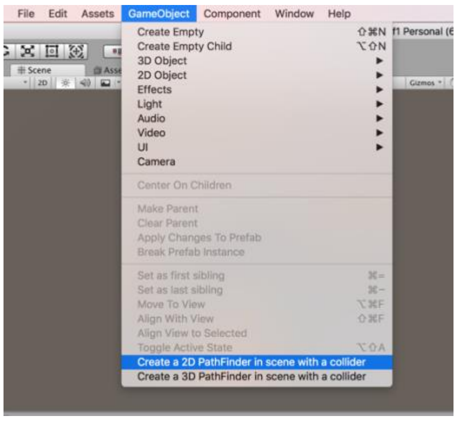
1. A *Node* is like Junctions in real life. Each Node has a specific ID which is auto-generated. You can find the IDs in Pink.
2. A *Path* is a connection between 2 Nodes. Each Path has a specific ID, which is auto-generated. You can find these IDs in Cyan.
   Note that Node ID & Path ID are different. They are also represented by different colors. Pink in the case of Nodes and Cyan for paths.

3. (Optional) Once you have the paths, you can provide cost per each path. Pathfinding prefers to use paths with lower costs. Also, you can set a few Paths as One-ways. The direction of the one way is ***From*** to ***To***, in the inspector.
   Note: Remember that Pathfinding takes two metrics into consideration before finding the shortest path.
   a. The costs you give for the *Path.*
   b. The physical distance between the nodes. (distance between vector3 of the source node and the destination node)

How to create Nodes and Paths for 2D

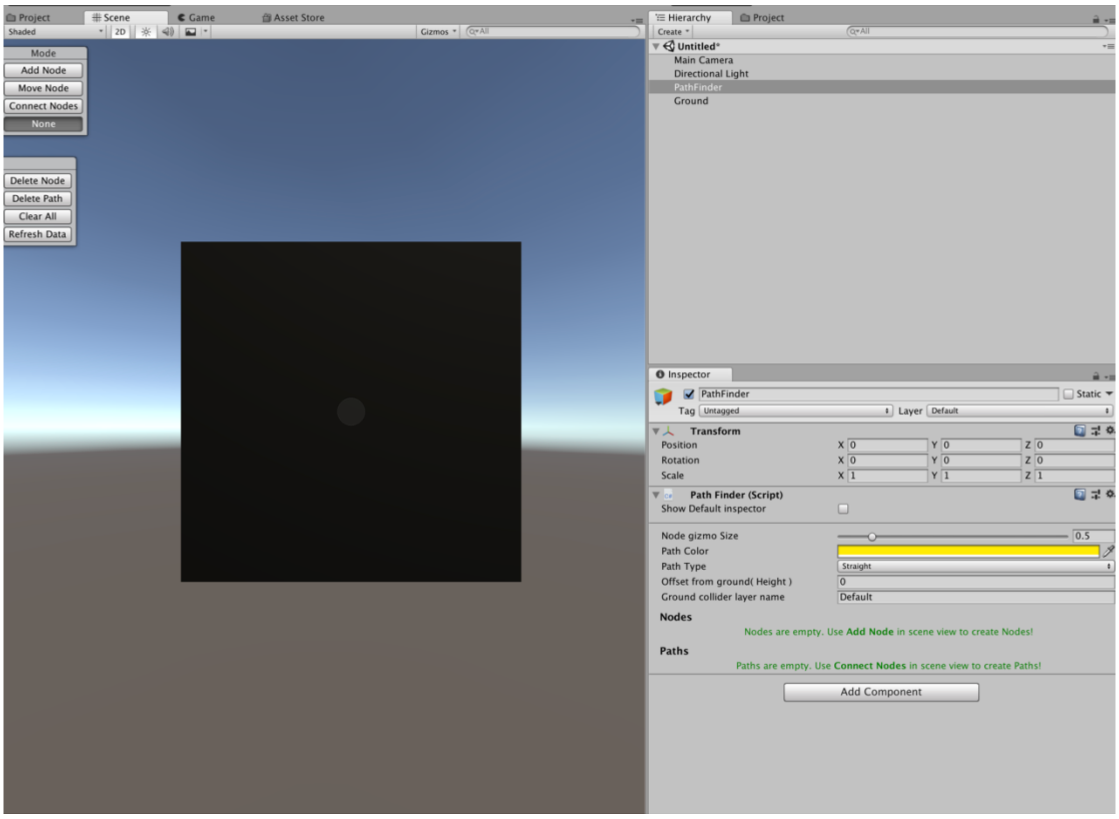
Step 1: Create Pathfinder

Create Pathfinder script and Ground colliders in the scene.



Step 2: Select Pathfinder

Select Pathfinder Gameobject in the hierarchy. This brings ups the Pathfinder GUI on the scene view.



PathFinder Static
Tag Untagged Layer Default

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Path Finder (Script)

Show Default inspector

Node gizmo Size	0.5
Path Color	🟨
Path Type	Straight
Offset from ground(Height)	0
Ground collider layer name	Default

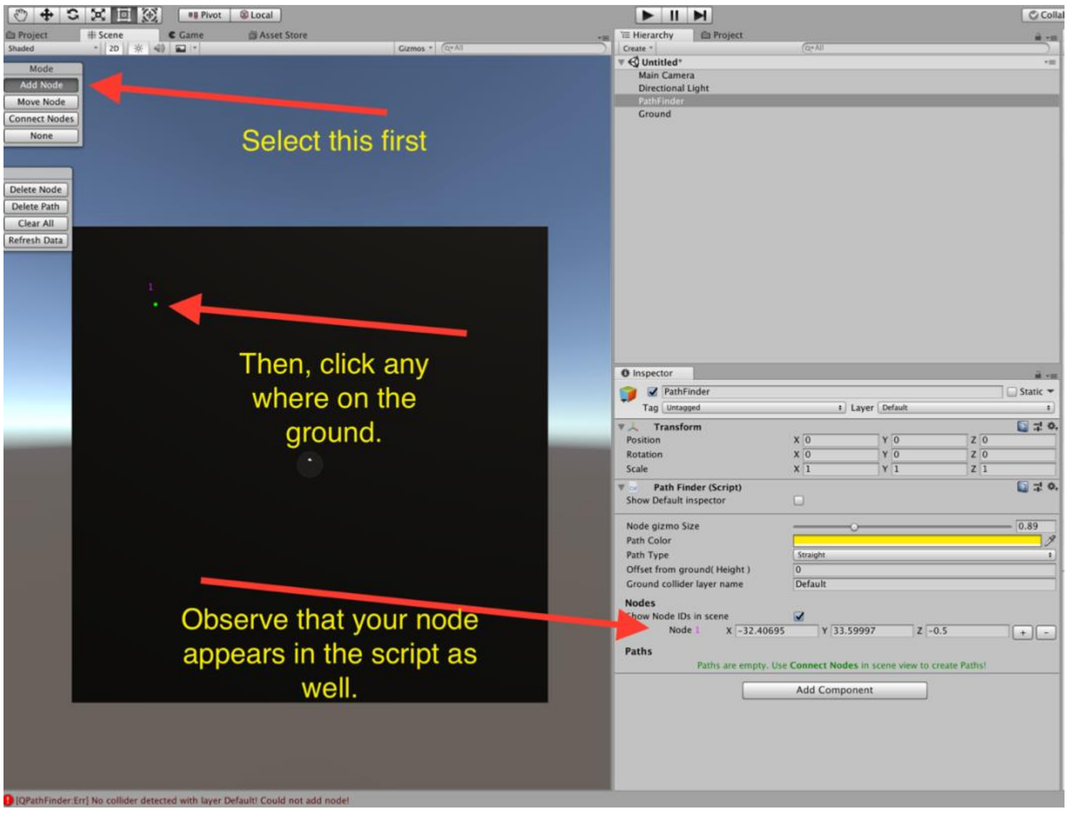
Nodes

Nodes are empty. Use **Add Node** in scene view to create Nodes!

Paths

Paths are empty. Use **Connect Nodes** in scene view to create Paths!

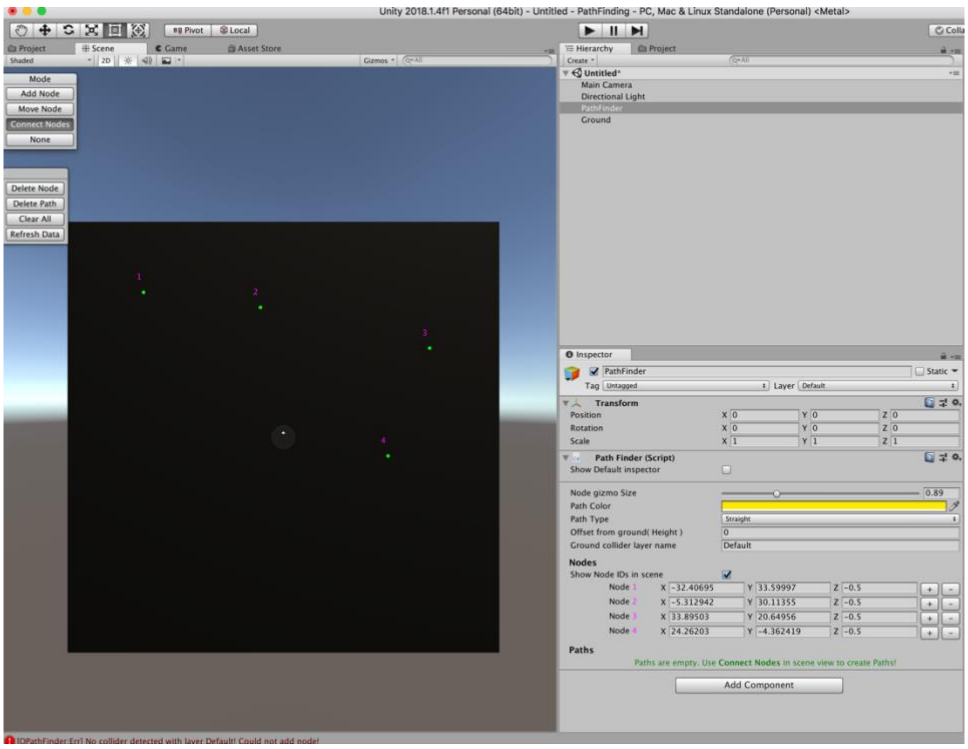
Step 3: Create nodes



```
Inspector
PathFinder                                    Static
Tag  Untagged          Layer  Default
Transform
Position    X 0    Y 0    Z 0
Rotation    X 0    Y 0    Z 0
Scale       X 1    Y 1    Z 1
Path Finder (Script)
Show Default inspector
Node gizmo Size                0.89
Path Color
Path Type                      Straight
Offset from ground( Height )   0
Ground collider layer name     Default
Nodes
Show Node IDs in scene
    Node 1   X -32.40695   Y 33.59997   Z -0.5
Paths
Paths are empty. Use Connect Nodes in scene view to create Paths!
```

```
[QPathFinder:Err] No collider detected with layer Default! Could not add node!
```

Repeat the process multiple times. Then you would have something like this

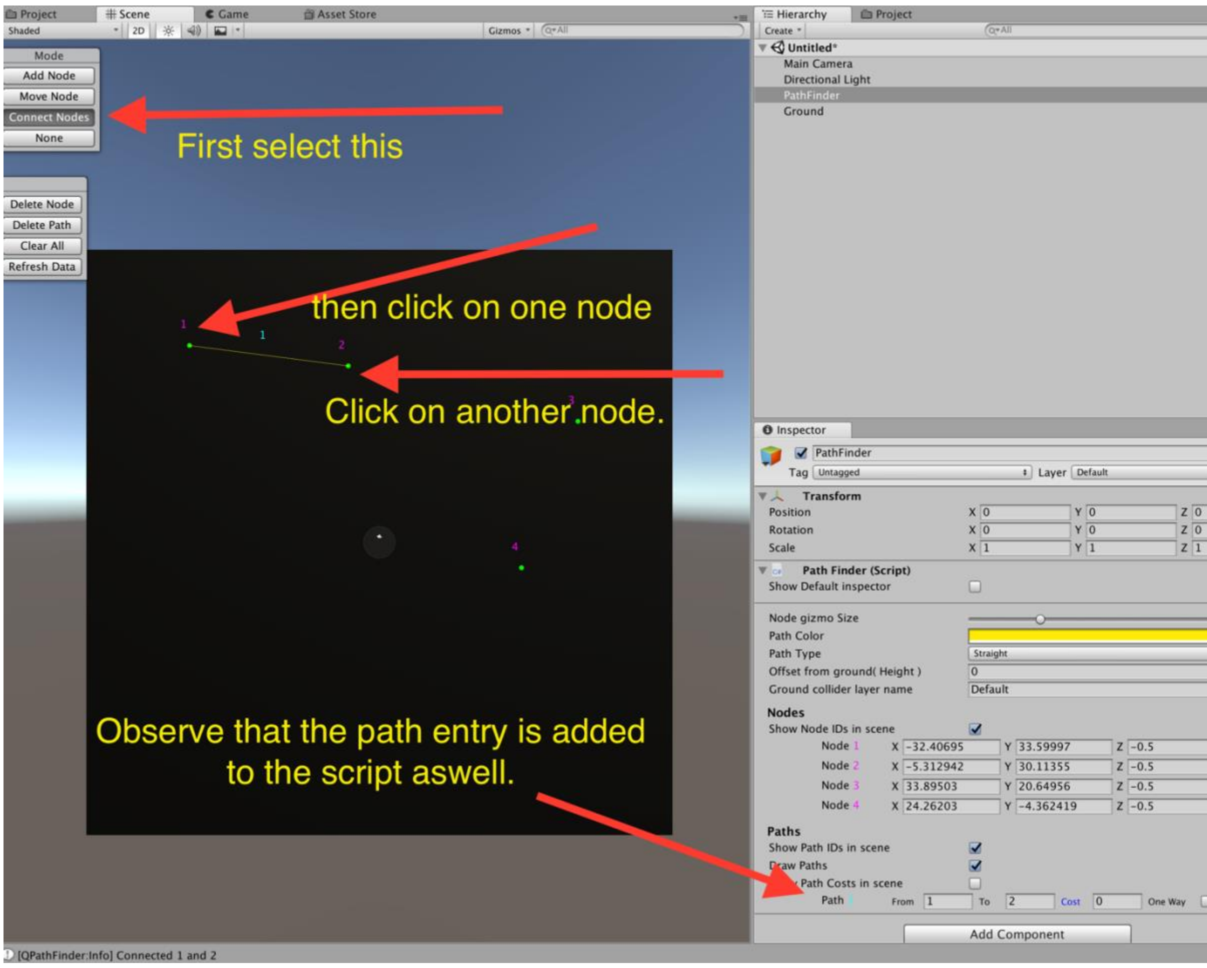


```
Inspector
PathFinder                                    Static
Tag  Untagged          Layer  Default
Transform
Position    X 0    Y 0    Z 0
Rotation    X 0    Y 0    Z 0
Scale       X 1    Y 1    Z 1
Path Finder (Script)
Show Default inspector
Node gizmo Size                0.89
Path Color
Path Type                      Straight
Offset from ground( Height )   0
Ground collider layer name     Default
Nodes
Show Node IDs in scene
    Node 1   X -32.40695   Y 33.59997   Z -0.5
    Node 2   X -5.312942   Y 30.11355   Z -0.5
    Node 3   X 33.89503    Y 20.64956   Z -0.5
    Node 4   X 24.26203    Y -4.362419  Z -0.5
Paths
Paths are empty. Use Connect Nodes in scene view to create Paths!
```

```
[QPathFinder:Err] No collider detected with layer Default! Could not add node!
```

Step 4: Create Paths (Connections)

This is a critical step, where u create paths between nodes.



PathFinder
Tag Untagged Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Path Finder (Script)
Show Default inspector ☐
Node gizmo Size
Path Color
Path Type Straight
Offset from ground(Height) 0
Ground collider layer name Default
Nodes
Show Node IDs in scene ✓

	X	Y	Z
Node 1	-32.40695	33.59997	-0.5
Node 2	-5.312942	30.11355	-0.5
Node 3	33.89503	20.64956	-0.5
Node 4	24.26203	-4.362419	-0.5

Paths
Show Path IDs in scene ✓
Draw Paths ✓
Path Costs in scene ☐
Path From 1 To 2 Cost 0 One Way

[QPathFinder:Info] Connected 1 and 2

Step 5 (Optional): Enter costs for each path

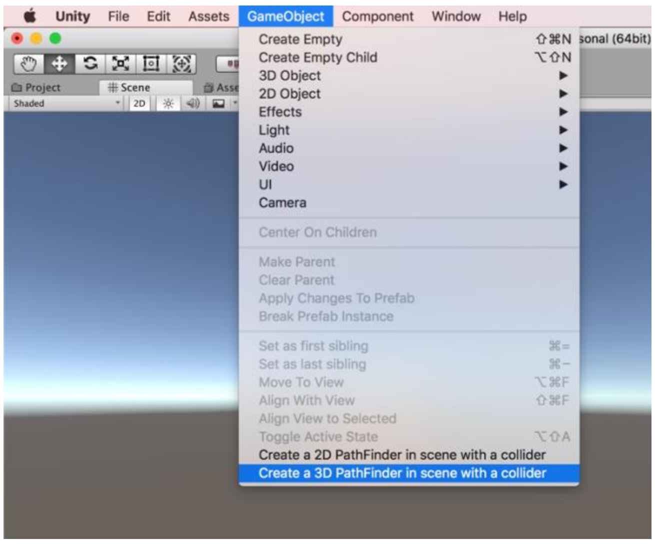
When u select Pathfinder GameObject, you can Enter the cost for each path.

Or set a few Paths are one-way. One-Ways go from ***From*** to ***To*** Nodes (in inspector).

How to create Nodes and Paths for 3D

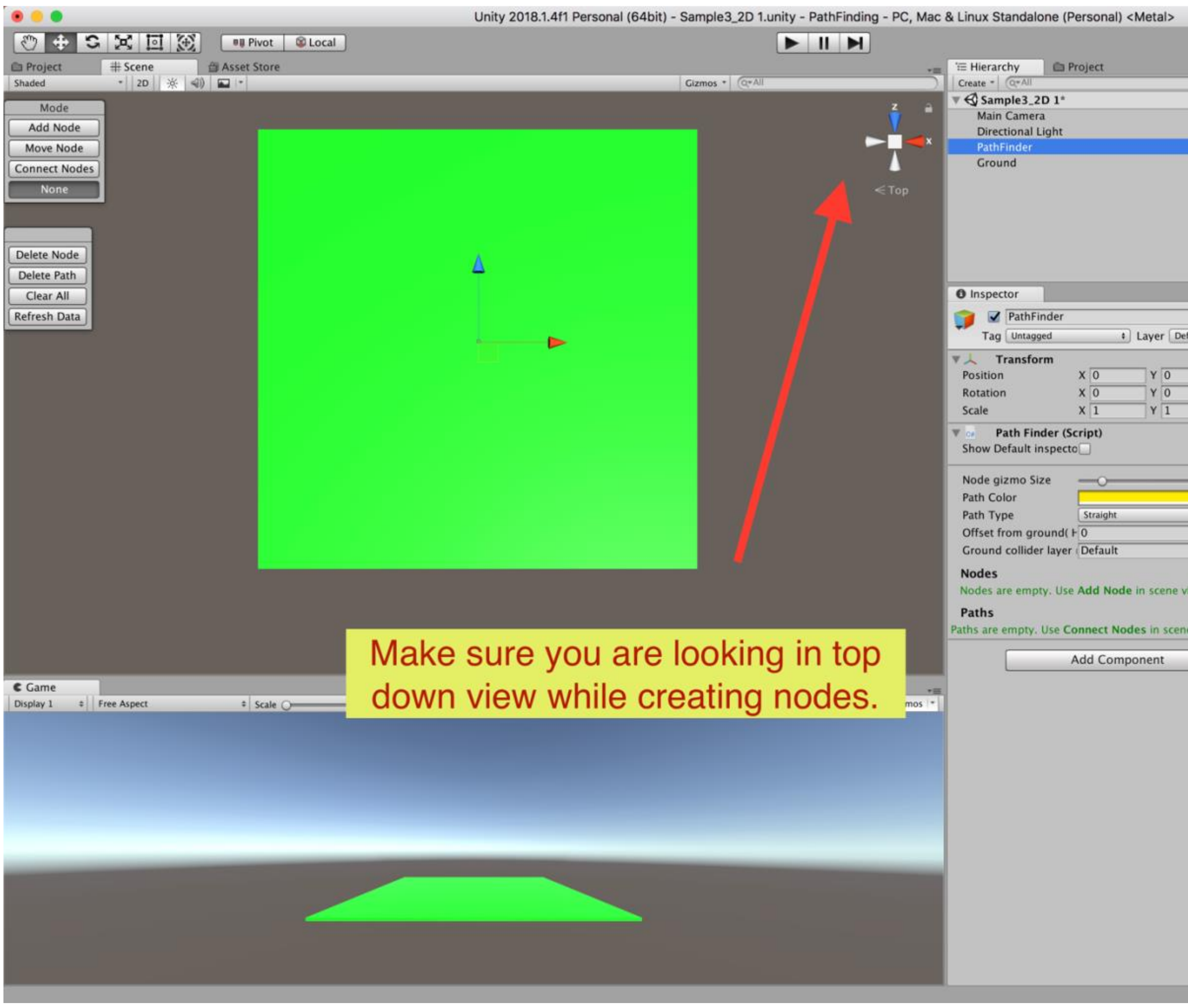
Step 1: Create Pathfinder

Create Pathfinder GameObject and ground collider.



Step 2: Select Pathfinder

Select *Pathfinder* Game object from the Hierarchy. This will show the Pathfinder GUI on the scene view.



Make sure you are looking in top down view while creating nodes.

PathFinder
Tag Untagged Layer
Transform
Position X 0 Y 0
Rotation X 0 Y 0
Scale X 1 Y 1
Path Finder (Script)
Show Default inspecto
Node gizmo Size
Path Color
Path Type Straight
Offset from ground(0
Ground collider layer Default
Nodes
Nodes are empty. Use **Add Node** in scene v
Paths
Paths are empty. Use **Connect Nodes** in scen

Step 3,4,5:

these steps are same as in 2D. Please refer to that. Go there.

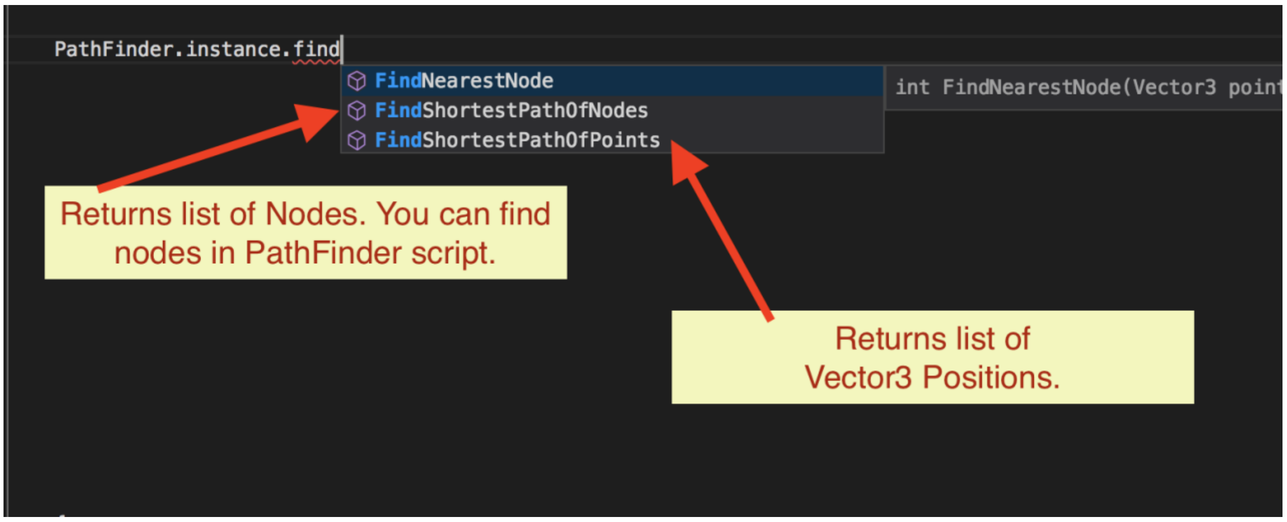
How to fetch the shortest path:

Once you have created all Nodes and Paths. Use Pathfinder class to access all the methods necessary.

include namespace QPathFinder wherever needed.

```
using QPathFinder;
```

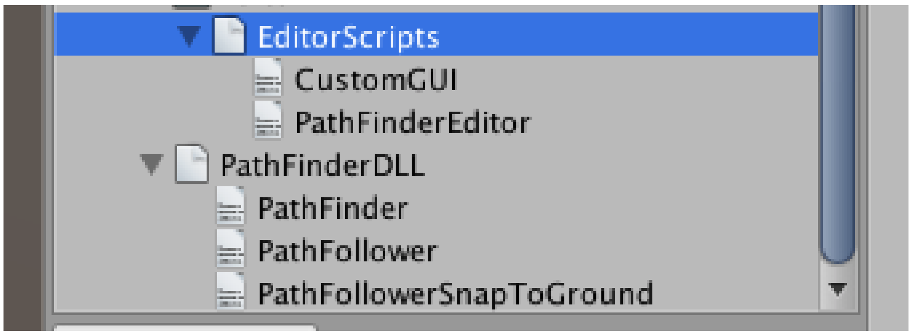
As long as you have Pathfinder script in the **scene**, you can access PathFinder.Instance directly from the code.



```
PathFinder.instance.find
FindNearestNode
FindShortestPathOfNodes
FindShortestPathOfPoints
int FindNearestNode(Vector3 point
```

Returns list of Nodes. You can find nodes in PathFinder script.
Returns list of Vector3 Positions.

Note for Free-Version users: As I had included DLLs instead of script files, you can expand the pathfinder DLL in unity and find the scripts there. You can attach them to gameobjects just like any other scripts.



There are 3 Methods to fetch the paths from PathFinder Class

1.

```
/// Finds shortest path between Nodes.
/// Once the path if found, it will return the path as List of nodes (not positions, but nodes. If you need positions, use FindShortestPathOfPoints).
/// <returns> Returns list of **Nodes**</returns>
/// <param name="fromNodeID">Find the path from this node</param>
/// <param name="toNodeID">Find the path to this node</param>
/// <param name="executionType">Synchronous is immediate & locks the control till path is found and returns the path.
/// Asynchronous type runs in coroutines without locking the control. If you have more than 50 Nodes, Asynchronous is recommended</param>
/// <param name="callback">Callback once the path is found</param>


public void FindShortestPathOfNodes ( int fromNodeID, int toNodeID, Execution executionType, System.Action<List<Node>>  callback );
```

2.

```
/// Finds shortest path between Nodes.
/// Once the path is found, it will return the path as List of Positions (not Nodes, but vector3. If you need Nodes, use FindShortestPathOfNodes).
/// <returns> Returns list of **Positions**</returns>
/// <param name="startNodeID">Find the path from this node</param>
/// <param name="endNodeID">Find the path to this node</param>
/// <param name="pathType">Path type. It can be a straight line or curved path</param>
/// <param name="executionType">Synchronous is immediate & locks the control till path is found and returns the path.
/// Asynchronous type runs in coroutines without locking the control. If you have more than 50 Nodes, Asynchronous is recommended</param>
/// <param name="OnPathFound">Callback once the path is found</param>


public static void FindShortestPathOfPoints ( this PathFinder manager, int startNodeID, int endNodeID, PathLineType pathType, Execution executionType, System.Action<List<Vector3>> OnPathFound );
```

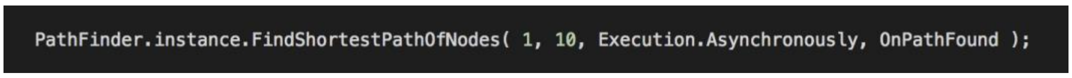
2. (overloaded)

```
/// Finds shortest path between Nodes.
/// Once the path is found, it will return the path as List of Positions ( not Nodes, but vector3. If you need Nodes, use FindShortestPathOfNodes).
/// <returns> Returns list of **Positions**</returns>
/// <param name="startNodeID">Find the path from this node</param>
/// <param name="endNodeID">Find the path to this node</param>
/// <param name="pathType">Path type. It can be a straight line or curved path</param>
/// <param name="executionType">Synchronous is immediate & locks the control till path is found and returns the path.
/// Asynchronous type runs in coroutines with out locking the control. If you have more than 50 Nodes, Asynchronous is recommended</param>
/// <param name="searchMode"> This is still WIP. For now, Intermediate and Complex does a tad bit more calculations to make the path even shorter</param>
/// <param name="OnPathFound">Callback once the path is found</param>


public static void FindShortestPathOfPoints ( this PathFinder manager, Vector3 startPoint, Vector3 endPoint, PathLineType pathType, Execution executionType, SearchMode searchMode, System.Action<List<Vector3>> OnPathFound );
```

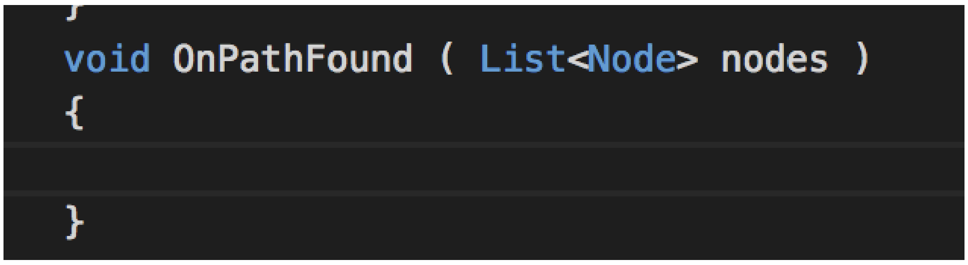
Note that some return list of Nodes and others return list of positions. Choose as you deem fit for your project.

Example of how to fetch the shortest path:

If I need to get the shortest path from node 1 to 10, you can get it like this,



```
PathFinder.instance.FindShortestPathOfNodes( 1, 10, Execution.Asynchronously, OnPathFound );
```



```
void OnPathFound ( List<Node> nodes )
{

}
```

Once the result is found, the callback will be triggered with list of nodes or positions (based on the method you use).

You can find more examples in the plugin under QPathFinder/Samples/Scripts.

How to traverse through the Path (Optional):

Once we get the shortest path, we can choose to move a character across the path.

This is Optional. You can write your own path follower classes to move your character based on what your game needs. But I have included a sample PathFollower to do just the basics.

How to Trigger PathFollower from the code

1. To move an object along the list of positions.

```
/// <Summary>
/// This will move the game object through the positions specified.
/// </Summary>
/// <param name="transform">The object you want to move along the path</param>
/// <param name="points">List of positions along which the object is moved.</param>
/// <param name="moveSpeed">Movement speed</param>

public static PathFollower FollowPath( Transform transform, List<Vector3> points, float moveSpeed );
```

2. To move an object along a list of positions on uneven terrains, you can use this.

```
/// <Summary>
/// This will move the game object through the points specified, Also, it will keep the gameobject snapped to the ground.
/// So if your Nodes are a little above the ground, your target will still move on the ground.
/// We are doing this by raycasting from above the player to the ground. At the ray cast hit position, we are snapping the player.
/// </Summary>
/// <param name="transform">The object you want to move along the path</param>
/// <param name="points">List of positions along which the object is moved.</param>
/// <param name="moveSpeed">Movement speed</param>
/// <param name="directionOfRayCast"> We use raycasting to find the ground
position. If your ground is down, the ray has to go down, so use Vector3.down.
</param>
/// <param name="offsetDistanceToFloatFromGround"> If you want your
character to float a little above the ground, give the offset value here. More you
give, character floats higher from the ground. </param>
/// <param name="groundGameObjectLayer">This is the ground Gameobject's
layer. When we use raycast we target to hit this layer</param>
/// <param name="offsetDistanceFromPoint">this is to calculate the raycast
origin, from where we shoot rays. raycast origin is generally above the player,
casting rays towards ground. For most cases, you can leave this as
default.</param>
/// <param name="maxDistanceForRayCast">this is the distance of ray from the
raycast origin. For most cases you can let this be default value. </param>


public static PathFollower FollowPathWithGroundSnap( Transform
transform, List<Vector3> points, float moveSpeed, Vector3
directionOfRayCast, float offsetDistanceToFloatFromGround, int
groundGameObjectLayer, float offsetDistanceFromPoint = 10, int
maxDistanceForRayCast = 40 )
```

3. To halt a moving object.

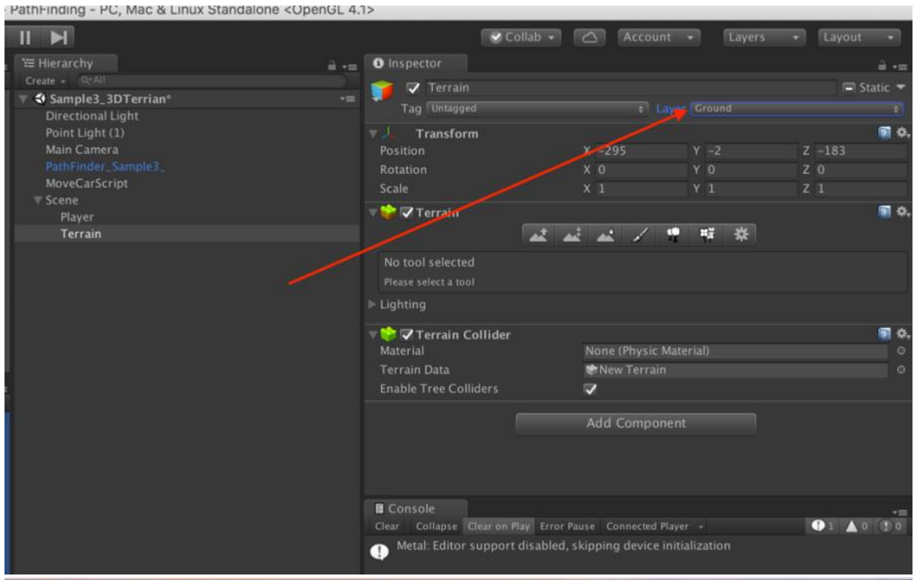
```
/// <summary>
/// Stops the gameobject while moving along the path.
/// </summary>
/// <param name="transform">The gameobject which needs to stop
moving</param>

public static void StopFollowing( Transform transform )
```

Note: All these Pathfollower methods are in PathFollowerUtility class. You can find more examples under QPathFinder/Samples/Scripts.

You can find more examples in the plugin under QPathFinder/Samples/Scripts.

Caution with 3D node creation:

1. Make sure you are in top-down view while adding nodes. This will make ensure that are placed where u tap.
2. The *Layer* of the ground collider should not be same as the *Layer* of the movable objects. Also, the *layer* of the ground collider should be entered in Pathfinder script.



Sample3_3DTerrian*
Directional Light
Point Light (1)
Main Camera
PathFinder_Sample3_
MoveCarScript
Scene
Player
Terrain

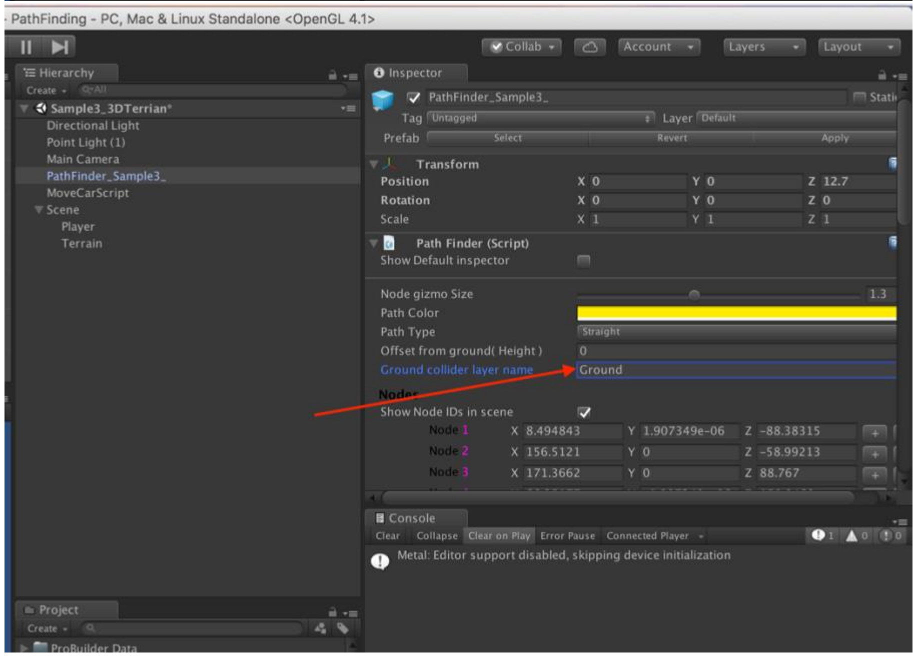
Terrain
Static
Tag Untagged
Layer Ground

Transform
Position X -295 Y -2 Z -183
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Terrain
No tool selected
Please select a tool
Lighting

Terrain Collider
Material None (Physic Material)
Terrain Data New Terrain
Enable Tree Colliders ✓

Metal: Editor support disabled, skipping device initialization



Sample3_3DTerrian*
Directional Light
Point Light (1)
Main Camera
PathFinder_Sample3_
MoveCarScript
Scene
Player
Terrain

PathFinder_Sample3_
Static
Tag Untagged
Layer Default
Prefab

Transform
Position X 0 Y 0 Z 12.7
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Path Finder (Script)
Show Default inspector

Node gizmo Size 1.3
Path Color 🟨
Path Type Straight
Offset from ground(Height) 0
Ground collider layer name Ground

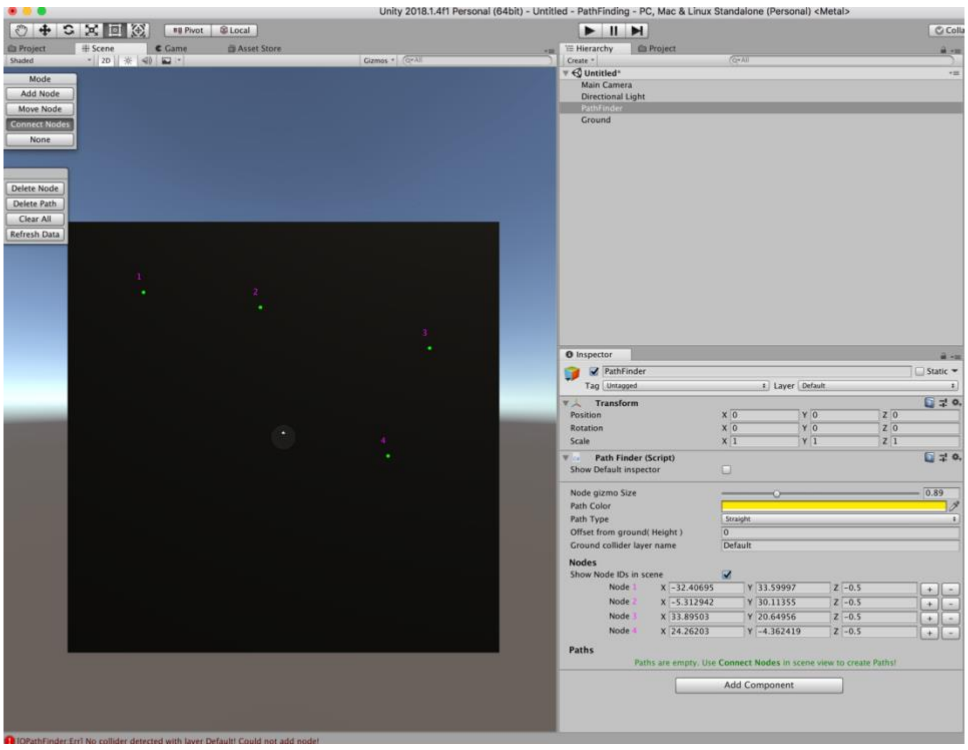
Nodes
Show Node IDs in scene ✓

Node 1	X 8.494843	Y 1.907349e-06	Z -88.38315
Node 2	X 156.5121	Y 0	Z -58.99213
Node 3	X 171.3662	Y 0	Z 88.767

Metal: Editor support disabled, skipping device initialization

FAQ

1. I cannot see Nodes or Paths in the scene



Unity 2018.1.4f1 Personal (64bit) - Untitled - PathFinding - PC, Mac & Linux Standalone (Personal) <Metal>

Untitled*
- Main Camera
- Directional Light
- PathFinder
- Ground

PathFinder — Static — Tag: Untagged — Layer: Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

Path Finder (Script)

Show Default inspector ☐

Node gizmo Size	0.89
Path Color	🟨
Path Type	Straight
Offset from ground(Height)	0
Ground collider layer name	Default

Nodes

Show Node IDs in scene ✓

	X	Y	Z
Node 1	-32.40695	33.59997	-0.5
Node 2	-5.312942	30.11355	-0.5
Node 3	33.89503	20.64956	-0.5
Node 4	24.26203	-4.362419	-0.5

Paths

Paths are empty. Use **Connect Nodes** in scene view to create Paths!

[OPathFinder Err] No collider detected with layer Default! Could not add node!

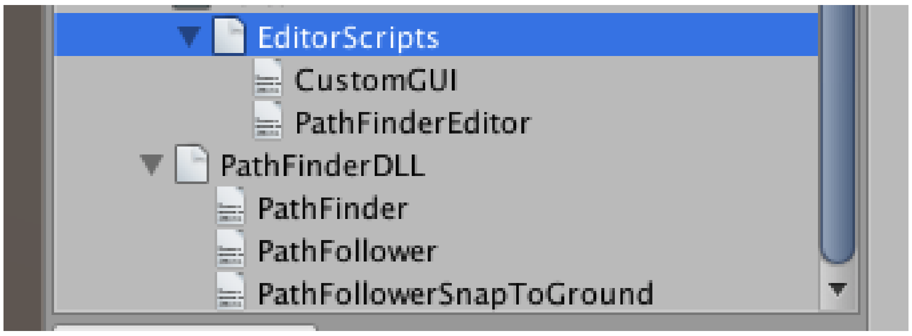
If you cannot see the above GUI in your scene window, make sure you select the pathfinder GameObject in your hierarchy.

If you do not have pathfinder in your hierarchy, refer here to create one.

2. To all the users using **Free-Version**

There is absolutely no difference between free-version and full-version except that I have provided DLLs instead of source code. Other than that, you can do everything a full-version can do. There are no restrictions.

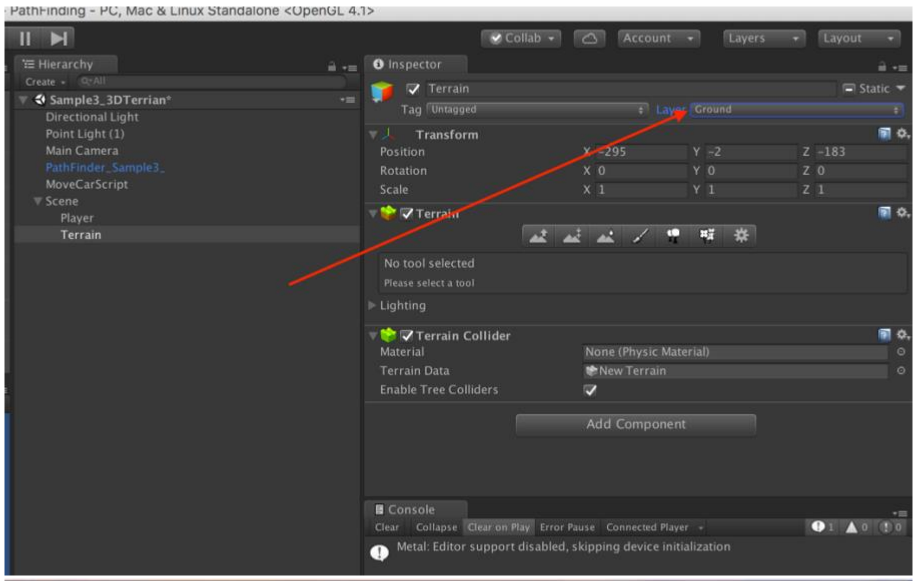
When you expand your DLL, you can find the class declarations there. You can drag these into the scene just like any other script.



- EditorScripts
  - CustomGUI
  - PathFinderEditor
- PathFinderDLL
  - PathFinder
  - PathFollower
  - PathFollowerSnapToGround

3. While trying to Create nodes, I am clicking on the Collider, but nodes are not created:

Make sure the collider name in PathFinder script and your collider layer name are the same.


PathFinding - PC, Mac & Linux Standalone <OpenGL 4.1>

Hierarchy: Sample3_3DTerrian*, Directional Light, Point Light (1), Main Camera, PathFinder_Sample3_, MoveCarScript, Scene, Player, Terrain

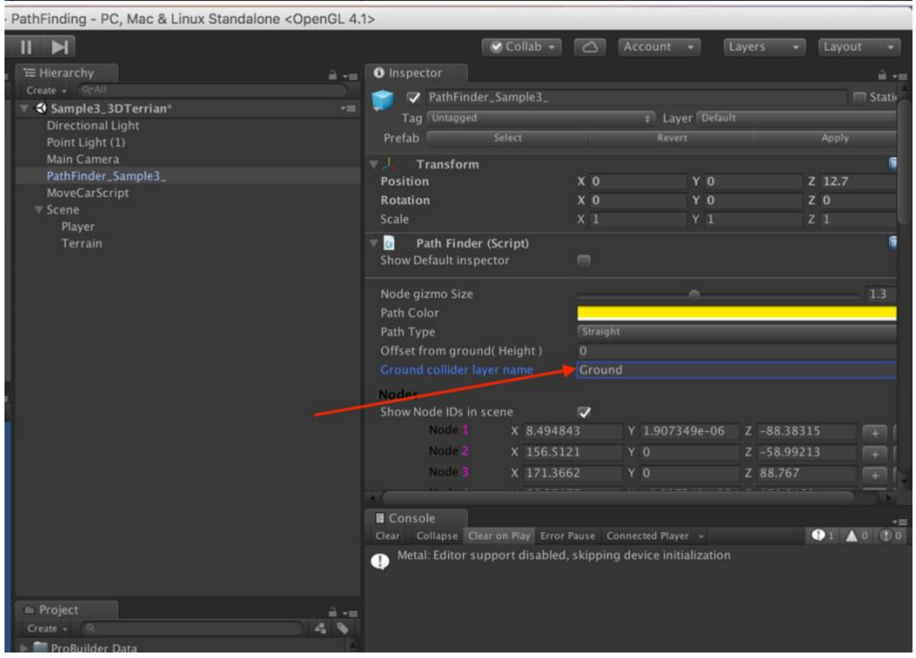
Inspector: Terrain — Tag Untagged — Layer Ground

Transform: Position X -295 Y -2 Z -183; Rotation X 0 Y 0 Z 0; Scale X 1 Y 1 Z 1

Terrain: No tool selected. Please select a tool

Terrain Collider: Material None (Physic Material); Terrain Data New Terrain; Enable Tree Colliders ✓

Console: Metal: Editor support disabled, skipping device initialization


PathFinding - PC, Mac & Linux Standalone <OpenGL 4.1>

Inspector: PathFinder_Sample3_ — Tag Untagged — Layer Default

Transform: Position X 0 Y 0 Z 12.7; Rotation X 0 Y 0 Z 0; Scale X 1 Y 1 Z 1

Path Finder (Script): Show Default inspector

Node gizmo Size 1.3; Path Color; Path Type Straight; Offset from ground(Height) 0; Ground collider layer name Ground

Nodes: Show Node IDs in scene ✓

Node 1 X 8.494843 Y 1.907349e-06 Z -88.38315
Node 2 X 156.5121 Y 0 Z -58.99213
Node 3 X 171.3662 Y 0 Z 88.767

Console: Metal: Editor support disabled, skipping device initialization

4. My object is not moving properly in 3D Terrain

Make sure the collider name in PathFinder script and your collider layer name are the same. And it's best if your ground collider has a different layer than your player. E.g. Ground can have a layer "Ground" and characters can have something else. The reason behind this is that we are doing ray casts to find the height of the ground and if there is any other game object with the same layer, we will have trouble finding proper height.

5. Logging and other debugging

You can enable multiple levels of logging, using these.

```
QPathFinder.Logger.SetLoggingLevel( debugLogLevel );

QPathFinder.Logger.SetDebugDrawLineDuration ( debugDrawLineDuration );
```

Note: When log level is set to "Info", you can also see Debug Lines drawn, which shows the shortest path chosen and other information using debug lines in scene view.

Make sure you have Gizmos turned on.

If you have any other questions, please visit my website Veluri.in and drop a message with your email. I will get back to you.

Thanks for using this plugin and hope it helps in whatever you are trying to build :)

Vijay Veluri