

# Graphway: A\* Pathfinding Documentation

## Version 1.1

An extended and fully up to date version of this documentation can be found at:  
<https://zenythstudios.com/graphway/documentation>

To save disk space, the 'Demo' folder may be deleted without affecting Graphway's main functionality, however it is recommended to take a look at the demo scene before doing so.

### Quick Start Guide

#### 1. Create a New Graphway Object

Select *Tools* → *Graphway* in the **Menu Bar**.

#### 2. Create Nodes

Think of nodes as starting points, end points and junctions on a path. They should be placed at positions in your scene where paths will start, finish or overlap.

To place nodes select the Graphway object in the Hierarchy then click the '**Enable Node Placement**' button in the Inspector. You can now left click in the Scene Window to place new nodes. When finished click the '**Disable Node Placement**' button in the Inspector Panel.

If the scale of the node display is too small/large it can be adjusted in the Inspector Panel with the '**Node Size**' property.

If nodes are not being placed, check your target mesh has a collider attached.

#### 3. Create Connections

Connections join two nodes together to form a path. To create a connection **select two nodes in either the Scene View or Graphway Hierarchy**, then in the **Inspector Panel** select one of the '**Connect Nodes**' buttons explained below:

*Bidirectional* – The path can be followed in both directions.

*Unidirectional A to B* – The path can only be followed in one direction from the node with the lowest ID to the node with the highest ID. Node ID's are displayed above the node position in the Scene View.

*Unidirectional B to A* – The path can only be followed in one direction from the node with the highest ID to the node with the lowest ID.

When a connection is made a line will be drawn between both nodes.

If Unidirectional, an arrow will display the direction of allowed movement.

The 'C' key on your keyboard can be used to connect nodes bidirectionally to speed up workflow.

Connections can be broken by **selecting two connected nodes** and clicking the '**Disconnect Nodes**' button in the **Inspector Panel**.

## 4. Create Subnodes (Optional)

The use of subnodes greatly increases the efficiency of the pathfinding algorithm. Subnodes are used to set a path from one node to another where the path does not follow a straight line.

Subnodes are created similarly to nodes. To create subnodes find the connection between two nodes in the '**Connections**' hierarchy in the Graphway hierarchy (they are ordered from lowest node ID to highest node ID), then select the '**Enable Subnode Placement**' button in the **Inspector Panel**. Subnodes can now be placed by **left clicking in the Scene Window**. When finished select the '**Disable Subnode Placement**' button in the Inspector Panel.

**Important:** Subnodes are always ordered from the node with the lowest ID to the node with the highest ID regardless of the connection type. Subnodes can be reordered by adjusting their position in the hierarchy under the relevant connection.

## 5. Pathfinding Example (C#)

That's it, your all set up! Now it's possible to use Graphway in your scripts to find a path from A to B. At runtime Graphway creates a job system in the background so it can handle complicated graphs and find paths without severely impacting on your game's performance. A path request is sent to Graphway from your script, then when a path has or hasn't been found a callback method is invoked.

Assuming there is only one Graphway object in your scene you can call the following static method:

```
Graphway.FindPath(Vector3 origin,  
    Vector3 targetPosition,  
    Action<GwWaypoint[]> callback,  
    bool clampToEndNode = true,  
    bool debugMode = false  
);
```

### Parameters:

**origin** - path starting position.

**targetPosition** - path target end point.

**callback** - the method that will be called when a pathfinding request has completed. Returns waypoint objects if a path was found or null if not.

**clampToEndNode** - Fixes the end point to a node on the graph closest to the target end point.

**debugMode** – Set to true to see the pathfinding algorithm in action slowed down. Make sure Gizmos are turned on!

Please refer to the demo scene for a working example.

Note 1 – If there is more than one Graphway object in your scene you must set a reference in your script to the Graphway object you want to use, then use the non-static method 'PathFind' rather than '[Graphway](#).FindPath'. The method arguments remain the same however.

Note 2 – To increase/decrease the performance of the pathfinding algorithm the 'Pathfind Frame Limit' variable can be adjusted on the main Graphway object in the Inspector Panel. The higher the value the more operations will be allowed per frame at a cost of overall performance and frame rate.

## Advanced Features

### Connection Speed Weights (Optional)

Depending on your use case, some paths may be faster to travel than others, even if the nodes are a greater distance apart. To account for this when finding the fastest path you can assign a speed weight to a connection to make it relatively faster or slower than others.

To assign a speed weight **select a connection in the Graphway Hierarchy** and adjust the '**Speed Weight**' property in the **Inspector Panel**.

### Disabled Connections (Optional)

Connections can be disabled by default and then re-enabled at run time through scripting or vice versa. To disable a connection by default **select a connection in the Graphway Hierarchy** then check the '**Disabled**' property in the **Inspector Panel**. To enable a connection at runtime use the following snippet:

```
Graphway graphway = GameObject.Find("Graphway").GetComponent<Graphway>();

// For BIDIRECTIONAL connections
// We must enable/disable connections from both ends
graphway.nodes[nodeAID].EnableConnection(nodeBID);
graphway.nodes[nodeBID].EnableConnection(nodeAID);

// For UNIDIRECTIONAL connections
// Only associated with one node from the direction of travel
graphway.nodes[nodeFromID].EnableConnection(nodeToID);
```

## Extended Options

For further options, such as creating nodes and connections at runtime, please consult the extended documentation on our website available here:

<https://zenythstudios.com/graphway/documentation>

## Need More Help?

Should you need any additional help, have a question, or would like to suggest an improvement please do not hesitate to contact us at:

[zenythstudios@gmail.com](mailto:zenythstudios@gmail.com)

## Thank You

Thank you for supporting Graphway! We hope it has made your project's development that little bit easier and wish you every success with your project.

We kindly ask, if possible please add a review for Graphway on the Unity Asset Store to let others know what you think.