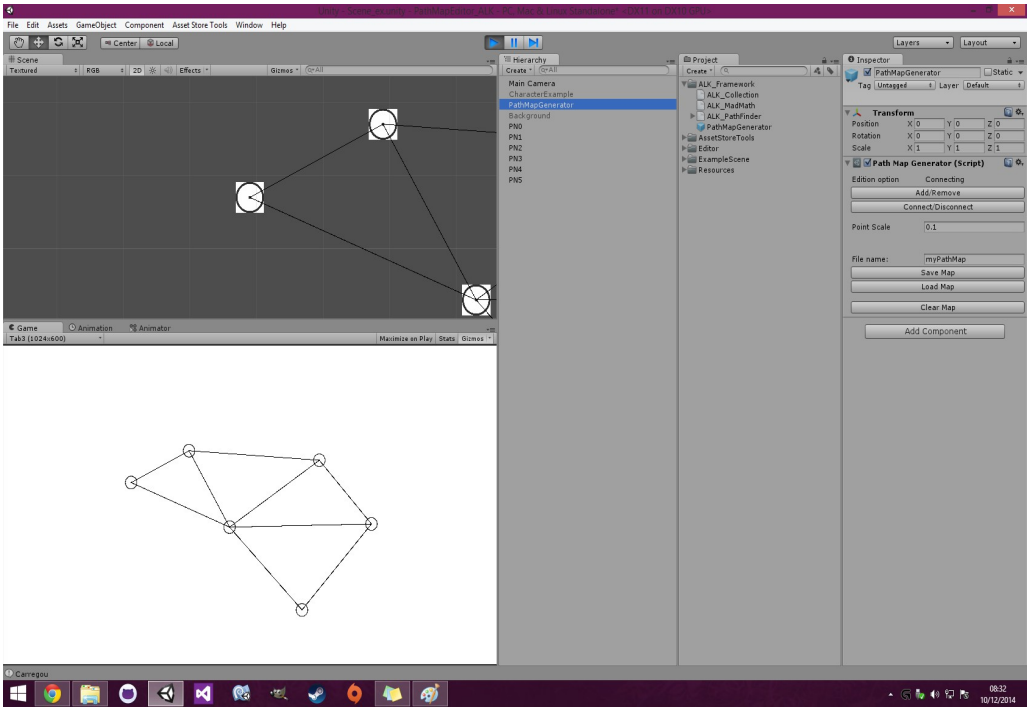
> PathMap Editor (From ALK_Framework) <

Created by: Adriano L. Kerber - adrianomaddog@hotmail.com –
www.adrianomaddog.webuda.com
A C# framework!

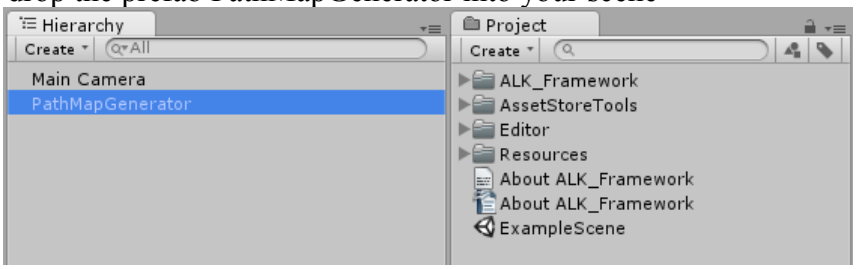


About: Path Map Editor is an editor which allows you create path maps through Unity editor. It creates nodes and connect them to use this map of nodes as you wish. It's important to know that the nodes are only visible during edition. So when you wish to use the map as path for a character it will load only the nodes not the GameObjects, since visual representation is not needed for the nodes.

Setting the editor

First, to create the map, you will need to drag and drop the prefab **PathMapGenerator** from the folder ALK_Framework in the scene, after that you have to run the scene because this path map editor was built to work at run time. So just to be clear, follow these steps:

- Drag and drop the prefab PathMapGenerator into your scene

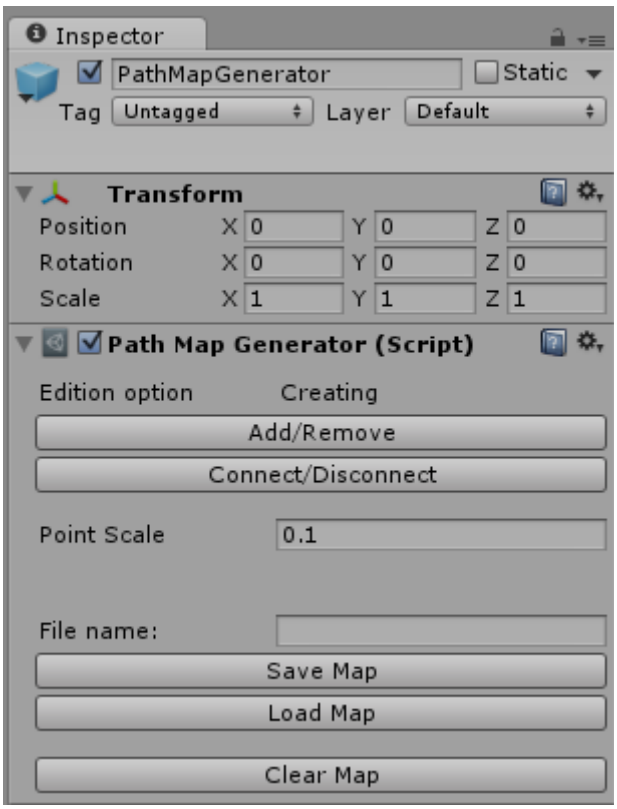


- Play/run scene

- Now you are ready to create/edit your path map!!! :)

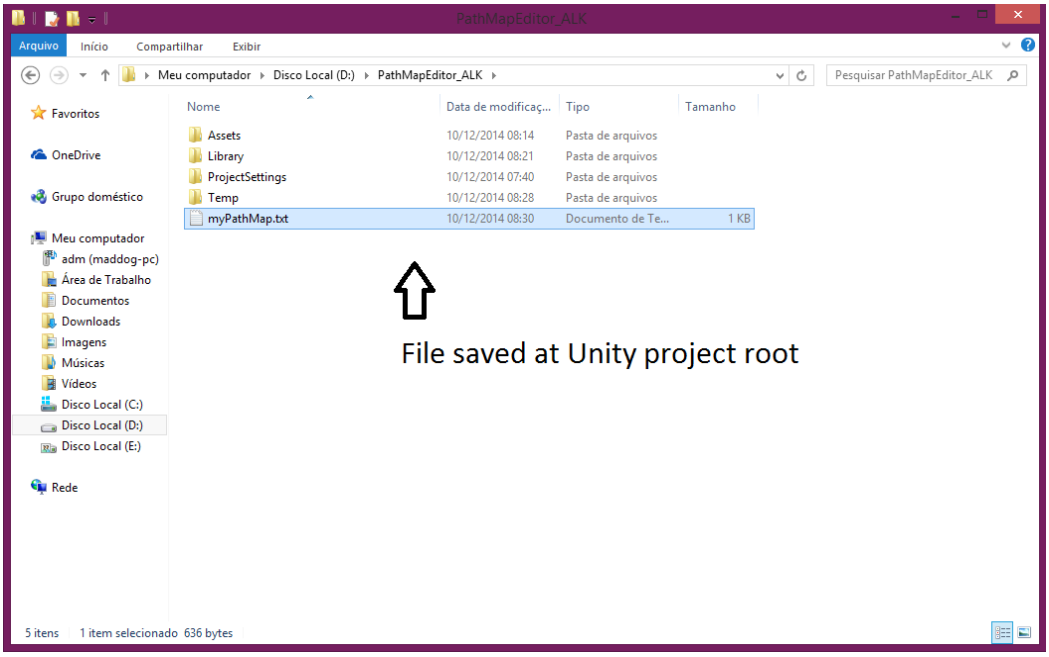
Using the editor (Creating a map)

Once you have set all the project, you can start using the editor of maps. The information about the path map is shown in the inspector of the object **PathMapGenerator**. Is importat to note that each node will become a GameObject on scene (Just while editing in PathMapGenerator).



Following the image above, you have:

1. Edition Buttons: These buttons change among the edition modes. There are four: Add (Add a note), Remove (Delete a node), Connect (Connect nodes) and Disconnect (Disconnect nodes). Each button:
    > Add/Remove: Toogle between the add option and the remove option.
    > Connect/Disconnect: Toogle between the connect option and the disconnect option.
2. Point Scale: Is the scale of the node presented on screen. IMPORTANT: Changing this value will only affect the nexts nodes that you create.
3. File name: Keeps the name of the map that you are creating. IMPORTANT: Just use the name of the file, DO NOT put the path of the file.
4. Save/Load:



> Save Map: Save the map with the specified name. IMPORTANT: The file is saved on the root directory of the current Unity project.

> Load Map: Load the map with the specified name. IMPORTANT: It replaces a map that is already loaded.

5. Clear Map: Delete the loaded map from memory. IMPORTANT: It doesn't delete a saved map from disk.

When connecting/disconnecting a node from another, you will first have to select a node (It doesn't gives you a visual feedback if there is a selected node or not, so be sure of what you selected) and then click on the other node that you want to connect/disconnect. To deselect a node you have to click outside any node (Inside the Game window).

Shortcuts: Use the key numbers (1 [ADD], 2 [REMOVE], 3 [CONNECT] and 4 [DISCONNECT]) to change among edition modes. IMPORTANT: Sometimes it may not change the field that shows the name of the selected mode on the inspector of PathMapGenerator, but it work exactly the same as changing the mode by the buttons on the inspector.

Using the created map in game

You may want to use the classes **PathMap** (Where the map actually takes place) as attribute of a character which needs to move throught the map, and **PositionNode** which is the class that represents the smallest particle of the PathMap and is the node of the map. To use this class you will just need to instanciate it and load the map. After that you can use some methods that the class has to work with the nodes and find for example the closest node of the children from the current node. For instance, you can use the classes through this:

```
using ALK_PathFinder; // To use, call the namespace of the PathMap;
```

The main methods that you may want to use directly of PathMap are:

```
// Finds the closest node to the given X and Y:
public PositionNode FindMyNodeInAllNodes(float X, float Y);
// Returns the closest node from targetX and targetY in the currentNode's children:
public PositionNode FindClosestNode(float targetX, float targetY, PositionNode currentNode)
```

One example of use can be found in the Assets folder of the project. But it's important to note that you need to put the file **myPathMap.txt** on solution level (One above Assets folder).

EXTRAS: You may want to use the extra classes from the framework as well, to mesure distances, and other functionalities. They can be accessed by the namespaces: `ALK_MadMath` end `ALK_Generic`. Have fun!