GUI Node Editor

1.0.0

Generated by Doxygen 1.8.12

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 GUINodeEditor Namespace Reference

Classes

- class BezierConfig
- class Dock

  Holds node connection data. Its `DockWindow` is the little box on node sides.
- class DockInput

  Helper for clarification of the docks side, as only output-input can be connected.
- class DockOutput

  Helper for clarification of the docks side, as only output-input can be connected.
- class DockWindow
- class Node

  Holds node data, its `NodeWindow` renders that data in the editor.
- class NodeEditor

  This is the node editor engine, it runs all the core logic of window manipulation. Inherits from MonoBehaviour, all editor configs are serialized here.
- class NodeEditorConfig
- class NodeEditorMinimap
- class NodeEditorWindow
- class NodeLogic

  Non editor related serialization, this is what gets serialized to a text file.
- class NodeWindow
- class NodeWindow_Menu
- class NumberField
- class PopAnywhereStack
- class RuntimeNodeEditor

  It will render the nodes on screen if attached to the gameObject that has NodeEditor.
- class TypeHolder

  Serializes the type `Type` with FullSerializer because `Type` serialization does not work in Unity.

Chapter 5

Class Documentation

5.1 Bezier Class Reference

Static Public Member Functions

- static void DrawBezier (Vector2 start, Vector2 end, Color color, float opacity=1, float thickness=2, float precision=15)

  Draws the bezier curve.

5.2 GUINodeEditor.BezierConfig Class Reference

Public Attributes

- int precision = 10

  how many lines should each curve have. More means smoother.
- Color **normalColor** = Color.black
- Color triggeredColor = Color.green

  color used when node.isTriggered is true.
- Color **connectingColor** = Color.cyan

5.3 GUINodeEditor.Dock Class Reference

Holds node connection data. Its `DockWindow` is the little box on node sides.

Inherited by GUINodeEditor.DockInput, and GUINodeEditor.DockOutput.

Public Member Functions

- **Dock** (Node node, Type type, string name, object initial)

Public Attributes

- TypeHolder typeHolder

  Type for dock matching, only docks matched by type can be connected.
- string name

  Used as an identifier.
- object value

  Value a dock is carrying. It is defined by your editor logic, you can get and set it.
- Node node

  Parent node reference.
- DockWindow **dockWindow**
- List< Dock > targets

  The list of docks it is connected to. These targets are always connected back to this dock.

5.3.1 Detailed Description

Holds node connection data. Its `DockWindow` is the little box on node sides.

5.4 GUINodeEditor.DockInput Class Reference

Helper for clarification of the docks side, as only output-input can be connected.

Inherits GUINodeEditor.Dock.

Public Member Functions

- **DockInput** (Node node, Type type, string name, object initial)

Additional Inherited Members

5.4.1 Detailed Description

Helper for clarification of the docks side, as only output-input can be connected.

5.5 GUINodeEditor.DockOutput Class Reference

Helper for clarification of the docks side, as only output-input can be connected.

Inherits GUINodeEditor.Dock.

Public Member Functions

- **DockOutput** (Node node, Type type, string name, object initial)

Additional Inherited Members

5.5.1 Detailed Description

Helper for clarification of the docks side, as only output-input can be connected.

5.6 GUINodeEditor.DockWindow Class Reference

Inherits GUINodeEditor.NodeEditorWindow.

Public Attributes

- Dock **dock**

Additional Inherited Members

5.7 DrawGridOnScreen Class Reference

Public Member Functions

- void **OnGUI** ()

Public Attributes

- Texture2D **gridTexture**
- float **gridOpacity** = 1f
- float gridUnit = 25

  Edge length of the grid square in pixels.
- float gridMultiplyFactor = 20

  For performance reasons (until I generate the bigger texture from script), bigger texture with grid tiled should be provided. This is the number of times the grid has fit in the bigger texture
- Vector2 panningOffset

  Offset of the whole editor area.

5.7.1 Member Data Documentation

5.7.1.1 gridMultiplyFactor

```
float DrawGridOnScreen.gridMultiplyFactor = 20
```

For performance reasons (until I generate the bigger texture from script), bigger texture with grid tiled should be provided. This is the number of times the grid has fit in the bigger texture

5.8 Drawing Class Reference

Static Public Member Functions

- static Color MultOpacity (Color color, float opacity)

  Multiply opacity.
- static Rect **GetRightRectFromPoints** (Vector2 a, Vector2 b)

5.9 DrawTextureOnScreen Class Reference

Public Member Functions

- void **OnGUI** ()

Public Attributes

- Texture2D **backgroundTexture**
- float **backgroundOpacity** = 1

5.10 GUINodeEditor.Node Class Reference

Holds node data, its `NodeWindow` renders that data in the editor.

Public Member Functions

- virtual void Update ()

  Called externally from `NodeLogic.Update`. Place your logic here.
- virtual void **Init** (Vector2 position=default(Vector2))
- virtual void Init (Vector2 position, NodeWindow nodeWindow, string title="")

  Init with the specified position (usually position of the menuNode.rect), nodeWindow, parent node and title.
- T **GetFirstTargetValue**< **T** > (Dock dock, object returnIfNull=default(object))
- DockInput **AddInput** (Type type, string name="", object initial=null, int insertAtIndex=-1)
- DockOutput **AddOutput** (Type type, string name="", object initial=null, int insertAtIndex=-1)
- int **GetValidIndex** (int count, int index)
- object **CreateInstance** (Type type)
- DockInput **GetDockInputByName** (string name)
- DockOutput **GetDockOutputByName** (string name)

Public Attributes

- bool isTriggered

  If this is true, the connection will be drawn with bezierConfig.triggeredColor.
- List< DockInput > inputs

  List of left side docks.
- List< DockOutput > outputs

  List of right side docks.
- NodeWindow nodeWindow

  Renders the node data in `override OnGUI`.

5.10.1 Detailed Description

Holds node data, its `NodeWindow` renders that data in the editor.

5.10.2 Member Function Documentation

5.10.2.1 Init()

```
virtual void GUINodeEditor.Node.Init (
            Vector2 position,
            NodeWindow nodeWindow,
            string title = "" )  [virtual]
```

Init with the specified position (usually position of the menuNode.rect), nodeWindow, parent node and title.

Parameters

position	Position.
nodeWindow	Node window.
node	Parent node.
title	Title.

5.11 GUINodeEditor.NodeEditor Class Reference

This is the node editor engine, it runs all the core logic of window manipulation. Inherits from MonoBehaviour, all editor configs are serialized here.

Inherits MonoBehaviour.

Public Member Functions

- void Update ()

  Calls nodeLogic.Update.
- void DrawNodeWindows ()

  OnGUI of the editor, handles drawing of windows and all the functionality.
- bool ShouldRepaint ()

  For UnityEditor, if Repaint should be called, to prevent low fps.
- void DrawDebug ()

  Draws debug with some useful editor states.
- void **DrawBackground** ()
- void **DrawGrid** ()
- void **DrawMinimap** ()
- void MoveConnection (DockOutput moveTargetDock, DockInput fromDock, DockInput toDock)

  Moves the connection from one dock to another.
- void **MoveConnection** (DockInput moveTargetDock, DockOutput fromDock, DockOutput toDock)
- void DeconnectDocks (Dock a, Dock b)

  Deconnects the docks, remoces each other from their targets.

- bool IsAllowedConnectionBetweenDocks (Dock startDock, Dock endDock)

  Returns if the two docks are of the matching type and not from the same parent node.
- void **ConnectDocks** (DockOutput output, DockInput input)
- void **ConnectDocks** (DockInput input, DockOutput output)
- void ClickSelect (NodeWindow nw)

  If <Shift> is held toggles selection, else deselects all and selects just it.
- void **UnselectWindow** (NodeWindow nw)
- void **SelectWindow** (NodeWindow nw)
- T CreateNewWindow< T > (Vector2 position=default(Vector2), bool isMenu=false)

  Call this from the menu node without parameters, it will get menu position.
- void DrawNodeConnections (Node n, Vector2 positionOffset=default(Vector2), float scale=1, bool is↩ Minimap=false)

  Also used in minimap.
- void DeselectAll ()

  Clears selectedWindows list.
- bool **IsNodeSelected** (Node node)
- void DeleteSelected ()

  Triggered on <Del>.
- string GetSaveLoadPath (string fileName)

  Gets Resources folder path with fileName.
- void Save (string fileName="")

  Saves the file with given fileName to save/load path in Resources folder. Overwrites if file with the same name exists (you have to check that separately). Sets the lastSaveLoadPath.
- void Load (string fileName="")

  Loads the file with given fileName from save/load path in Resources folder. If no file is found, creates a new save file with that name. Sets the lastSaveLoadPath.

Static Public Member Functions

- static NodeEditor **GetOrCreateNodeEditor** (string nodeEditorName, Type menuType=default(Type))

Public Attributes

- NodeEditorConfig **config** = new NodeEditorConfig ()
- string saveLoadName = "defaultSave"

  Currently loaded file name. Has to be set externally like from SaveLoadGUI.
- string saveLoadResourcesFolderName = ""

  Name of the folder where save/load files are kept (Resources folder exists in builds).
- string lastSaveLoadPath = ""

  last path that was either saved or loaded. This will be used to load after deserialize.
- TypeHolder menuTypeHolder = new TypeHolder ()

  Menu type, to specify which menu node will be used for this editor.
- NodeLogic nodeLogic = new NodeLogic ()

  Holds nodeEditor data like nodes, serialized with FullSerializer for polymorphism support.
- List< NodeWindow > **selectedWindows** = new List<NodeWindow> ()
- NodeWindow **hoveredWindow**
- bool **isDragging**
- Vector2 **startDraggingPosition**
- NodeWindow **mainDraggedWindow**
- bool **isPanning**
- Vector2 **oldPanningOffset**

- Vector2 **startPanningPosition**
- bool **isPrePanning**
- Vector2 **prePanningPosition**
- bool **isSelecting**
- Vector2 **startSelectPosition**
- Vector2 **startSelectPanningOffset**
- bool **isConnecting**
- Dock **startConnectDock**
- bool **shouldEndConnecting**
- bool **isDeconnecting**
- Dock **startDeconnectDock**
- Dock **endDeconnectDock**
- NodeWindow **renamingWindow**
- string **renamingName**
- Node **nodeMenu**
- bool **shouldSpawnMenu**
- Dictionary< NodeWindow, bool > **windowVisibilities** = new Dictionary<NodeWindow, bool> ()

5.11.1 Detailed Description

This is the node editor engine, it runs all the core logic of window manipulation. Inherits from MonoBehaviour, all editor configs are serialized here.

5.11.2 Member Function Documentation

5.11.2.1 Load()

```
void GUINodeEditor.NodeEditor.Load (
            string fileName = "" )
```

Loads the file with given fileName from save/load path in Resources folder. If no file is found, creates a new save file with that name. Sets the lastSaveLoadPath.

5.11.2.2 Save()

```
void GUINodeEditor.NodeEditor.Save (
            string fileName = "" )
```

Saves the file with given fileName to save/load path in Resources folder. Overwrites if file with the same name exists (you have to check that separately). Sets the lastSaveLoadPath.

5.12 GUINodeEditor.NodeEditorConfig Class Reference

Public Member Functions

- Vector2 GetWindowOverflow ()

  To be able to draw docks outside the the node window, another rect is wrapped around it that draws both node window and docks. This is for how much the docks overflow that inner node window.

Public Attributes

- bool runUpdateInEditMode = true

  > Update is not called out of play mode. If true, nodeEditor will call nodeLogic.Update that calls Update for each node.
- GUISkin guiSkin

  Standard Unity GUISkin that changes the appearance of GUI elements.
- bool drawMinimap = true

  Minimap is drawn while panning or moving.
- NodeEditorMinimap **nodeEditorMinimap** = new NodeEditorMinimap ()
- DrawTextureOnScreen **drawTextureOnScreen** = new DrawTextureOnScreen ()
- bool snapToGrid = true

  Node window position is snapped to grid when dragging ends.
- DrawGridOnScreen **drawGridOnScreen** = new DrawGridOnScreen ()
- Rect dockRect = new Rect (0, 5, 14, 10)

  If using a custom dock Texture, adjust this to get pixel perfect placement.
- Texture2D **dockTexture**
- Vector2 tooltipOffset = new Vector2 (10, 10)

  Offset of the tooltip from mousePosition.
- bool drawDockTypeTooltip = true

  If tooltip with dock type should be drawn.
- BezierConfig **bezierConfig** = new BezierConfig ()

5.12.1 Member Function Documentation

5.12.1.1 GetWindowOverflow()

```
Vector2 GUINodeEditor.NodeEditorConfig.GetWindowOverflow ( )
```

To be able to draw docks outside the the node window, another rect is wrapped around it that draws both node window and docks. This is for how much the docks overflow that inner node window.

5.12.2 Member Data Documentation

5.12.2.1 runUpdateInEditMode

```
bool GUINodeEditor.NodeEditorConfig.runUpdateInEditMode = true
```

> Update is not called out of play mode. If true, nodeEditor will call nodeLogic.Update that calls Update for each node.

5.13 GUINodeEditor.NodeEditorMinimap Class Reference

Public Member Functions

- void **DrawMinimap** ()

Public Attributes

- List< Node > **nodes**
- float **opacity** = 0.6f
- Vector2 **panningOffset**
- Vector2 **dockRectSize**
- Rect **rect**
- bool **drawScreenRect** = true
- float **screenRectOpacity** = 0.4f
- float **scale** = 0.35f

5.14 GUINodeEditor.NodeEditorWindow Class Reference

Inherited by GUINodeEditor.DockWindow, and GUINodeEditor.NodeWindow.

Public Member Functions

- virtual void **OnGUI** ()

Public Attributes

- Rect **rect** = new Rect ()
- Color **backgroundColor**
- string **title** = ""
- NodeEditor **nodeEditor**
- Vector2 **dragStartOffset**

5.15 GUINodeEditor.NodeLogic Class Reference

Non editor related serialization, this is what gets serialized to a text file.

Public Member Functions

- void Update ()

  Calls `Update` of each node.

Public Attributes

- List< Node > **nodes** = new List<Node>()
- Vector2 **panningOffset** = Vector2.zero

5.15.1 Detailed Description

Non editor related serialization, this is what gets serialized to a text file.

5.16 GUINodeEditor.NodeWindow Class Reference

Inherits GUINodeEditor.NodeEditorWindow.

Inherited by GUINodeEditor.NodeWindow_Menu.

Public Member Functions

- virtual float **GetWindowWidth** ()
- virtual float **GetWindowHeight** ()
- virtual void **SetWindowSize** (Vector2 size)
- Color **SetOpacity** (Color c, float opacity)
- void **DrawTooltip** (string tooltip)
- void **DrawDock** (Dock dock, bool isTitleRow=false)

Public Attributes

- Node **node** = new Node ()
- Popup **popup** = new Popup()

Properties

- Vector2 **cachedSize** `[get, set]`

5.17 GUINodeEditor.NodeWindow_Menu Class Reference

Inherits GUINodeEditor.NodeWindow.

Public Attributes

- NodeWindow **clickedWindow**

Additional Inherited Members

5.18 GUINodeEditor.NumberField Class Reference

Public Member Functions

- int **Int** (int val)
- float **Float** (float val)

5.19 GUINodeEditor.PopAnywhereStack Class Reference

Public Member Functions

- object **Head** (object toReturnIfNull)
- void **HandleInsertRemove** (object obj, bool active, object instance)

Public Attributes

- List< object > **stack** = new List<object> ()

5.20 Popup Class Reference

First drawn as a button, then you have to draw it externally to appear outside the buttons area rect.

Public Member Functions

- Rect GetListRect ()

  Gets the popup list rect.
- void DrawList ()

  Draws GUI.SelectionGrid of 1 column.
- Enum EnumPopup (Enum currentEnum)

  Enum popup.

Public Attributes

- object identifier

  If this is not null, popup is opened and should be drawn externally.

5.20.1 Detailed Description

First drawn as a button, then you have to draw it externally to appear outside the buttons area rect.

```
// draw popup
if (p.identifier != null) {
    // get local rect
    Rect popupRect = p.GetListRect();

    // add position of your parent element
    popupRect.position += n.nodeWindow.rect.position;

    // draw a window because overlapping elements will consume click events
    GUI.Window (-i, popupRect, (id) => p.DrawList (), "", GUI.skin.box);

    // you might have to check for event on more places to close properly
    if (Event.current.type == EventType.mouseDown)
        p.identifier = null;
}
```

5.20.2 Member Function Documentation

5.20.2.1 GetListRect()

```
Rect Popup.GetListRect ( )
```

Gets the popup list rect.

Returns

The popup list rect.

5.21 GUINodeEditor.RuntimeNodeEditor Class Reference

It will render the nodes on screen if attached to the gameObject that has NodeEditor.

Inherits MonoBehaviour.

Public Attributes

- NodeEditor **nodeEditor**

5.21.1 Detailed Description

It will render the nodes on screen if attached to the gameObject that has NodeEditor.

5.22 SaveLoadGUI Class Reference

Save/Load GUI handler, set the fields externally.

Public Member Functions

- delegate void **OnSave** (string fileName)
- delegate void **OnLoad** (string fileName)
- void **OnGUI** ()

Public Attributes

- List< string > fileNames = new List<string> ()

  Displayed in the dropdown when _isLoadDialogue is set to true. When fileName is clicked, OnLoad is invoked with that fileName.
- OnSave onSave

  callback when Save button is pressed
- OnSave onLoad

  callback when Load button is pressed
- Vector2 scrollPosition = Vector2.zero

  scroll position of the area, if items exceed Screen.height
- string saveLoadName = ""

  String that the user can change.
- string currentName = ""

  Set this to the name that is currently loaded to highlight it.

5.22.1 Detailed Description

Save/Load GUI handler, set the fields externally.

5.22.2 Member Data Documentation

5.22.2.1 fileNames

```
List<string> SaveLoadGUI.fileNames = new List<string> ()
```

Displayed in the dropdown when _isLoadDialogue is set to true. When fileName is clicked, OnLoad is invoked with that fileName.

5.23 Serialization Class Reference

Static Public Member Functions

- static void **Save< T >** (string path, T obj)
- static T Load< T > (string resourcesLocalPath)

  Returns a generic object that is serialized to a Resources folder path.
- static string GetFullResourcesPath (string path)

  Returns the Application.dataPath + Resources folder.

5.24 StringSerializationAPI Class Reference

Static Public Member Functions

- static string **Serialize** (Type type, object value)
- static object **Deserialize** (Type type, string serializedState)

5.25 GUINodeEditor.TypeHolder Class Reference

Serializes the type `Type` with FullSerializer because `Type` serialization does not work in Unity.

Inherits ISerializationCallbackReceiver.

Public Member Functions

- void **OnBeforeSerialize** ()
- void **OnAfterDeserialize** ()

Public Attributes

- string **serializedType**
- Type **type**

5.25.1 Detailed Description

Serializes the type `Type` with FullSerializer because `Type` serialization does not work in Unity.

Index