

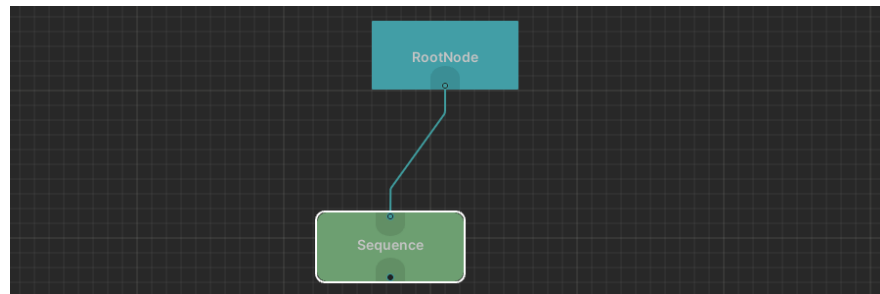
## Behaviour Tree quick start guide

- **Creating behaviour tree**

- In asset folder - Create > Behaviour Tree - will generate scriptable object of BT
- Behaviour Tree > Tree Editor will open editor window

- **Editing behaviour tree**

- Root node
  - First node will always be root node, to create it right click on the grid inside Tree Editor window > Root Node
- Adding more nodes
  - Right click > Select nodes from 3 categories
- Deleting nodes
  - Select nodes to delete, click delete button on your keyboard
- Connecting nodes
  - You can connect nodes by dragging from output handle to input



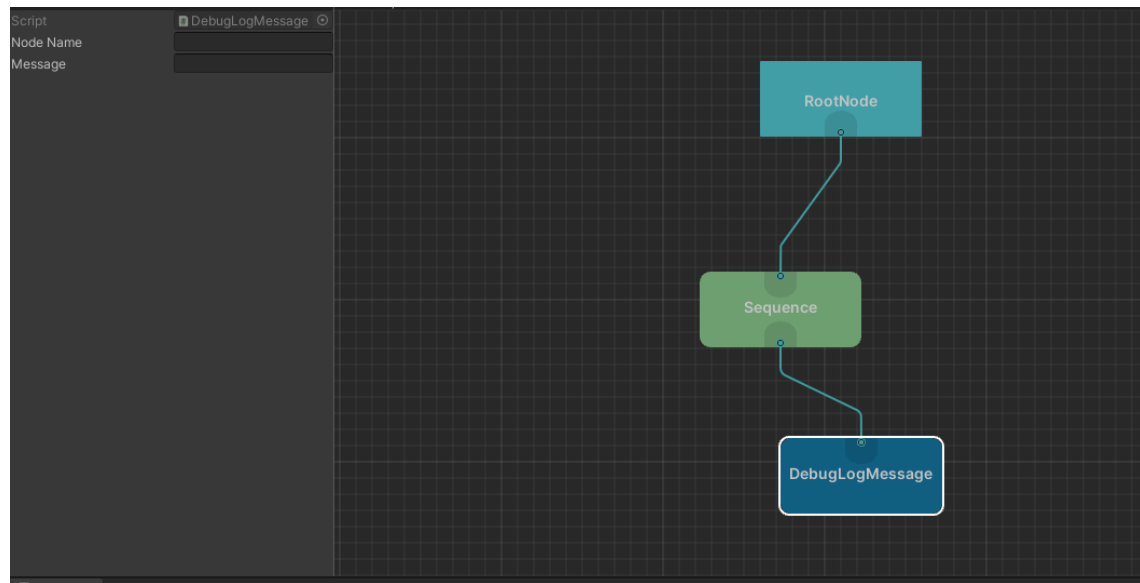
- Removing connection
  - Select connection and press delete button on your keyboard

- **Creating custom nodes**

- Create custom `c#` script
- There are 3 major types of nodes
  - `ActionNode` - performs specific task and returns state
  - `CompositeNode` - Performs actions on list of children nodes
  - `DecoratorNode` - performs actions on single child node
- Each new custom node will have to inherit from one of these 3 abstract types
- After implementing interface you'll have 3 methods
  - `OnStart()` will be called once when tree reaches node for the first
  - `OnStop()` will be called when node exists with success or failure state
  - `OnTick()` will be called when node is being evaluated by the tree
- Node now will appear in right click context menu on grid in Tree Editor window and can be created

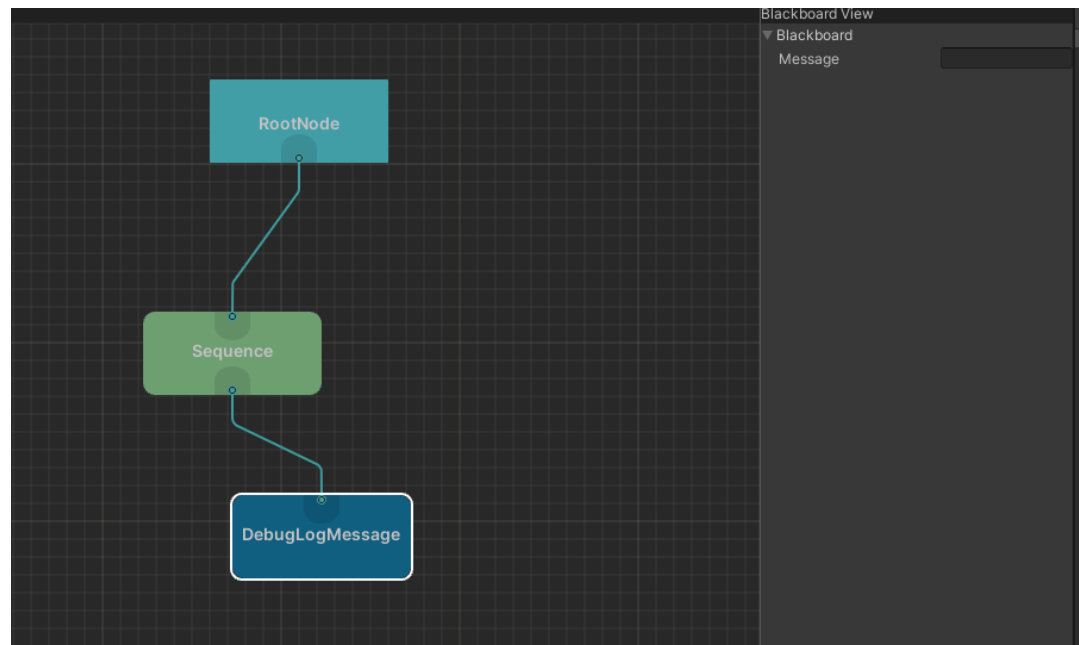
- **Node properties**

- All public fields on node will be exposed on node inspector



- Node Name - is universal to all nodes and changing the field will change visible node name on the graph after Right click > Refresh is clicked
- Message - in this case its public string field on DebugLogMessage node
- Node properties are serialized and saved to a tree

- **Blackboard**



- 
- Blackboard stores all required data that will be shared between nodes