You need to specify the available templates in `Editor/MbtTemplates.xml` (currently Helicopter nodes are not specified in there yet, and thus the MBT editor will assert when trying to open that tree).

When you open the editor you will see the following panels:

Signal View

Signals are strings, they can either be sent from the MBT itself or from the engine/code. A signal will set a tree variable to true or false when it is triggered. These variables can then be used to make decisions in the tree.

Examples of signals you can react to are:

- OnActionDone
- OnAvortAction
- OnNewAttentionTarget
- OnUseSmartObject
- OnTargetDead
- OnGroupMemberMutilated
- MovementPlanProduces
- OnFormationPointReached
- OnCoverCompromised
- OnTargetFleeing
- OnAttentionTargetThreatChanged
- OnNoTargetAwareness
- OnMovingInCover
- OnMovingToCover
- OnEnterCover
- OnLeaveCover
- OnTargetTooClose
- OnTargetTooFar

It is also possible to send signals from the behavior tree using the Signal node.

Timestamp View

Timestamps are set when a signal comes in, and can be used to check how long ago something happened.

e.g Timestamp TimeSinceTargetSeen can be set on the signal OnNewAttentionTarget, then certain nodes in the tree can check how long ago it was that this timestamp was set.

Variables View

Here, the variables for the tree are defined. These are just string values that can be set to true or false by signals. The following nodes can interact, react or query based on these variables, and in return on the signals.

- Case
- Ifcondition

Hyper Graph

This shows the MBT nodes, they are color coded as following:

- Yellow: Statemachine nodes
- Blue: State nodes
- Green: Tree nodes

The tree should be read from top to bottom, and then left to right.

There are certain rules to building the tree which are mentioned in the documentation for Modular behavior tree nodes, some important ones are:

[List important design example]

The tree automatically aligns all the nodes for you to give a clear view of the tree.

You can drag nodes on top of each other to link them as parent/child.

You can drag nodes around to change the execution order (left to right).

Property Panel

When you click on a node in the hypergraph it will populate the propertypanel with the associated properties for this MBT node.

Only changed properties are saved to the .xml, this also goes for newly added nodes.

Possible Future Improvements

- Create a decorator class to color nodes and add comments to the hypergraph
- Add a visual debugger in the style of flow graph
- Improve the validation of a tree by helping out on checking design patterns in the tree
- Enforcing correct childcount (1 or multiple).