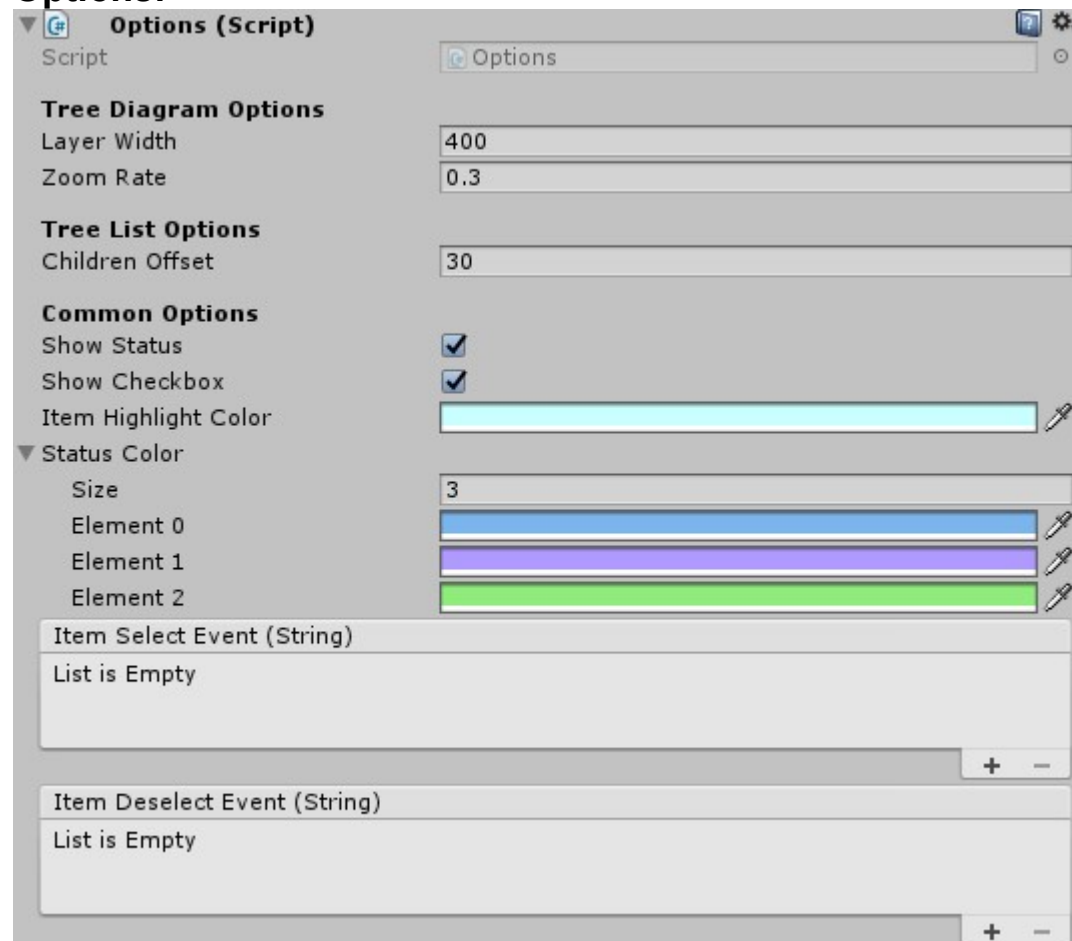


Basic usage:

1. Drag and drop TreeList/TreeDiagram prefab into the canvas
2. Adjust "Options" component of that prefab
3. Fill in "Series Data" component, this is content of the TreeList/TreeDiagram
4. Start the game

Component explanation:

Options:

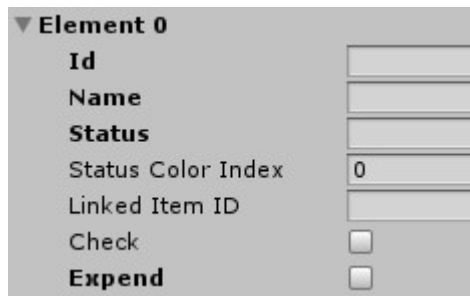


- Layer Width (float): width of each vertical layer of tree diagram
- Zoom Rate (float): tree diagram zoom rate
- Children Offset (float): offset distance between parent node and its children
- Show Status (bool): show/hide status of each node
- Show Checkbox (bool): show/hide checkbox of each node

- Item Highlight Color (color): color of selected item
- Status Color (color array): colors for status text
- Item Select/Deselect Event: invoked when an item is selected, pass in the item's "Linked Item ID" as parameter. Example can be found in "Demo_ItemSelectEvent" scene.

Series Data: Contain a list of "Series". "Data" and "Series" are defined below.

- Data: basic structure of each node.



▼ Element 0	
Id	<input type="text"/>
Name	<input type="text"/>
Status	<input type="text"/>
Status Color Index	0
Linked Item ID	<input type="text"/>
Check	<input type="checkbox"/>
Expend	<input type="checkbox"/>

- Id (string): id of the node
- Name (string): node name to be displayed
- Status (string): node status to be displayed
- Status Color Index (int): status color index, refer to Option->Status Color.
- Linked Item ID (string): string to be passed when "Item Select/Deselect Event" is invoked. Example can be found in "Demo_ItemSelectEvent" scene.
- Check (bool): item is checked/unchecked.
- Expend (bool): item is expended/collapsed.

- Series:



▼ Main	
Id	Main
▼ Data List	
Size	5
▶ s1	
▶ s2	
▶ s3	
▶ s4	
▶ s5	

- Id (string): id of the series. If series id matched any node id, this series will be attached to the node, meaning all nodes inside this series will be children of that node.
- Data List (Data array): nodes of this series.

Scripting :

Manipulate treeList.data / treeDiagram.data with normal c# list operations. When the data is ready, just call Refresh() function and the view will be updated.

The following codes will update the treeList with one node only.

```
//create a new node
Data exampleData = new Data();
exampleData.id = "000";
exampleData.name = "example";
exampleData.status = "test";
exampleData.statusColorIndex = 0;
exampleData.check = false;
exampleData.expend = false;

//create a main series and add one node
Series exampleSeries = new Series();
exampleSeries.id = "Main";
exampleSeries.dataList.Add(exampleData);

//clear the data, and add the example series
treeList.data.series.Clear();
treeList.data.series.Add(exampleSeries);

//refresh tree list
treeList.Refresh();
```

Note:

- * there must be a series with id "Main"
- * Use GetSelectedItems() to get multiple items selection, Example can be found in "Demo_ItemSelectEvent" scene.
- * Items can be customized, several item prefabs are provided, replace "Item Prefab" in "Tree List / Tree Diagram" component.