Feedback Manager Documentation

(by Almanzo McConkey)

Background

Introduction

The *Feedback Manager* is an Asset for *Unity*, made for Virtual Reality-based research. The purpose is to display graphical feedback to the participant in real time, showing their achievement in a given task, and related variables. This is very important in research which involves finding out how people learn skills over time by repeated use of those skills. Currently, the only visual feedback on achievement which can be easily given to participants in Virtual Reality-based experiments is text-based feedback. Therefore, we need a richer, more communicative form of visual feedback which can be displayed to the participant to clearly show their achievement, their improvement over time, and any other variables which may be related.

When you use the Feedback Manager, you will be setting up a virtual tablet, on which can be placed Feedback Tabs, each holding one Graph and, optionally, a list of the most recent values added to it. Data can be added to a Tab with a single function, which will then update all components of that Tab. Either one graph can be displayed, or multiple graphs to be cycled between. Currently, the available Tabs are: a progressive Tab, displaying one variable added over time; a scattergraph Tab, displaying two variables with an optional line of best fit; and a categorical data Tab, displaying a Pie Graph which updates as values are added.

Rationale

The easiest way, in Unity, to display a graph is to get a graphing asset. However, to get from the frameworks provided in a graphing asset to a functional graph displaying feedback in real time is a matter of hours of coding. The Feedback Manager drastically reduces the amount of time and effort required to create, customise and add data to a graph to display real-time feedback. It does this by sitting between the Graphing Asset (currently *Graph Maker*) and the Developer, abstracting away unnecessary complexities, including the vast majority of the programming.

This makes it far easier to give visual data to a player in a Virtual Reality application without a huge expenditure of time and effort.

Implementation

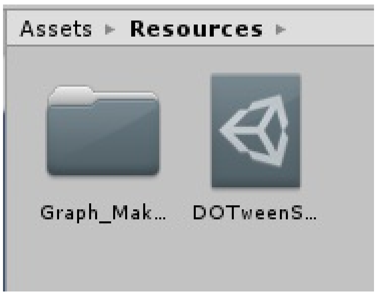
Initial Setup

1) Download Graphing Asset

Download the latest version of the graphing asset *Graph Maker* into the *Assets* folder of your project. This is the one for Unity 2017, which, at the time of writing, is not yet available on the Asset store. To get this version, you may want to contact the creator here.

2) Add it to *Resources* Folder

In the *Assets* folder, create a new folder called *Resources*. Move the *Graph Maker* Asset into this folder.



3) Check Graphing Asset File Directories

In the *Graph Maker* folder, ensure that the `EmptyGraph` Prefab is in file directory `Assets/Resources/Graph_Maker/Prefabs/Graphs/AxisGraphs/` and the `EmptyPie` Prefab is in file directory `Assets/Resources/Graph_Maker/Examples/X_Simple_Pie/`.

4) Setup DOTween Animations

Graph Maker should also have installed an animation Asset called *DOTween*. To set this up correctly, go to Tools in the menu bar and click on "DOTween Setup". The pop-up window will show you how to complete this.

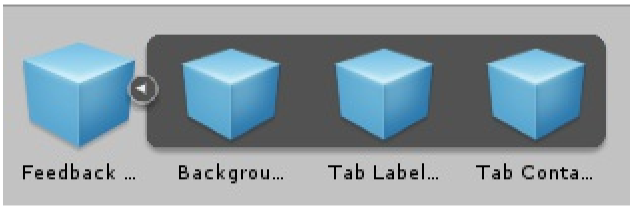
5) Download Feedback Manager

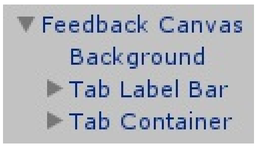
Download the **Feedback Manager** Asset into the *Assets* folder.

Setting up a Feedback Canvas

1) Add UI to Scene

In the **Feedback Manager** folder in your *Assets*, there will be a prefab called `Feedback Canvas`. Drag the Prefab into your Scene or Heirarchy.





2) Display in World Space

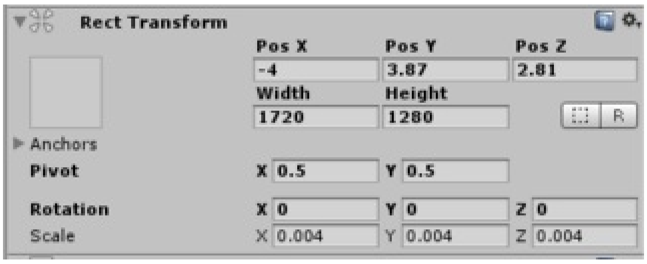
Ensure that the Feedback Canvas *Canvas* Component `Render Mode` is set to `World Space`, so that it will be displayed in the Scene. If using inside of another Canvas, ensure that the *Dynamic Pixels Per Unit* is set to `1`.

3) Adjust Size

To change the size of the Feedback Canvas, change the `Scale` attribute in the *Transform* component. All three scale dimentions should be set to the same value.

4) Adjust Shape

To change the rectangular shape of the UI, change the `Width` and `Height` attributes in the *Transform* component. I would suggest keeping the dimentions nearby the 1000 to 2000 range.



Rect Transform

Pos X -4 Pos Y 3.87 Pos Z 2.81

Width 1720 Height 1280

Anchors

Pivot X 0.5 Y 0.5

Rotation X 0 Y 0 Z 0

Scale X 0.004 Y 0.004 Z 0.004

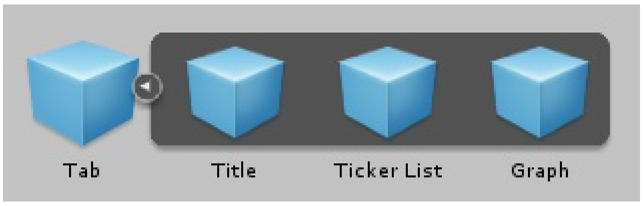
5) Position in Scene

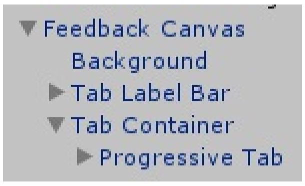
Resize and re-position the Feedback Canvas until you are happy with how it is displayed in the Scene.

Adding Feedback Tabs to a Feedback Canvas

1) Add Tab to Scene

In the Heirarchy of your Feedback Canvas, there will be a GameObject called `Tab Container`. In the **Feedback Manager** folder in your *Assets*, there will be a Prefab called `Tab`. Drag the `Tab` Prefab into the `Tab Container`.

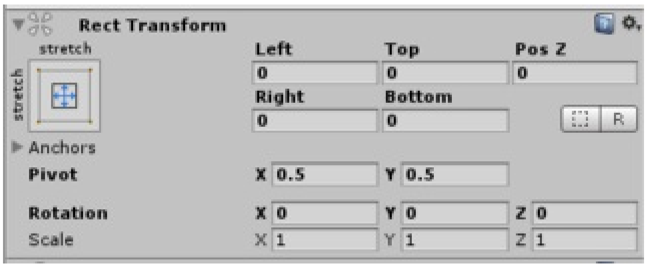




▼ Feedback Canvas
Background
▶ Tab Label Bar
▼ Tab Container
▶ Progressive Tab

2) Reset Transform

When you click on the Tab you have added to your Scene, you may see that the Transform variables are set to strange values. Set the *Left*, *Top*, *PosZ*, *Right*, *Bottom* and *X, Y, Z Rotation* values all to `0`, and the *X, Y, Z Scale* values to `1`. *X and Y Pivot* should be `0.5`.



Rect Transform

stretch

Left 0 Top 0 Pos Z 0

Right 0 Bottom 0

Anchors

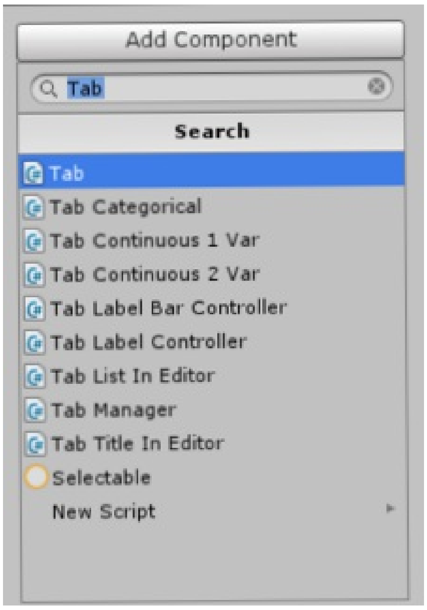
Pivot X 0.5 Y 0.5

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

3) Set up with a Graph-Specific Component

Click on *Add Component* in the Inspector for your Tab to choose what type of data it will display. You will see something like this:



Currently, options for Graph-Specific Components are:

- `Tab Continuous 1 Var` to display a single variable's progress;
- `Tab Continuous 2 Var` to display a Scatter Graph;
- `Tab Categorical` for a Pie Graph

For more information about these components and their variables, refer to the section *Graph-Specific Tab Components*.

4) Add more Tabs

Repeat **steps 1 - 4** to add multiple Tabs to the same Canvas. *Note: This will mean you will see the Tabs on top of one another in the Scene. If you want to edit just one, de-activate the other Tabs until you are finished editing it.*

5) Change the Color Scheme

If you like, adjust the color scheme of the Feedback Canvas and its Tabs using the *Appearance Manager* Component of the Feedback Canvas (refer to *Feedback Canvas* section to find out how to use this).

Controlling a Tab with a (C#) Script

1) Make sure Tab is fully set up

Make sure the values of the Graph-Specific Tab Component you have added are all set up as you want them.

2) Make sure Script is fully set up

Your MonoBehaviour Script must be an active Component of a GameObject to attach a Tab to it.

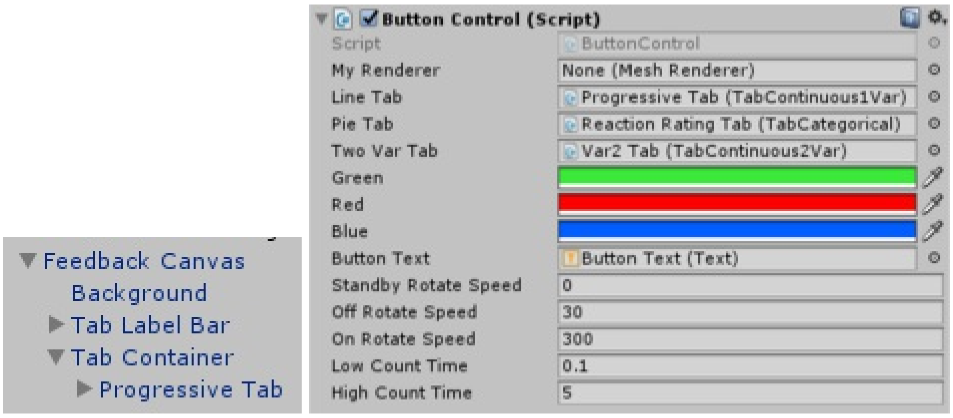
2) Reference Graph-Specific Tab Component in Script

In your MonoBehaviour class, create a public variable which will hold the Graph-Specific Tab Component.

- Tab Continuous 1 Var: `public TabContinuous1Var myTab;`
- Tab Continuous 2 Var: `public TabContinuous2Var myTab;`
- Tab Categorical: `public TabCategorical myTab;`

3) Link Tab to Script

Your variable should appear in your Script in the Inspector. Click and drag your Tab onto the variable in the Inspector to link your Tab to it.



- ▼ Feedback Canvas
  - Background
  - ▶ Tab Label Bar
  - ▼ Tab Container
    - ▶ Progressive Tab

☑ **Button Control (Script)**

Script	ButtonControl
My Renderer	None (Mesh Renderer)
Line Tab	Progressive Tab (TabContinuous1Var)
Pie Tab	Reaction Rating Tab (TabCategorical)
Two Var Tab	Var2 Tab (TabContinuous2Var)
Green	🟩
Red	🟥
Blue	🟦
Button Text	Button Text (Text)
Standby Rotate Speed	0
Off Rotate Speed	30
On Rotate Speed	300
Low Count Time	0.1
High Count Time	5

4) Use Scipt to add data to Tab

Each type of Graph-Specific Tab Component has a different function to use to add data to the Tab. These can be seen in the section called *Tab Functions*. For see how to use the functions, see the Demo Scene in the *Feedback Manager* Folder.

5) Another Method: *UnityEvents*

A *UnityEvent*, is an alternative way to link Tab functions to the script which allows the linked functions to be changed in the inspector rather than the script. Here is Unity's documentation on *UnityEvents*

Prefabs and their Components

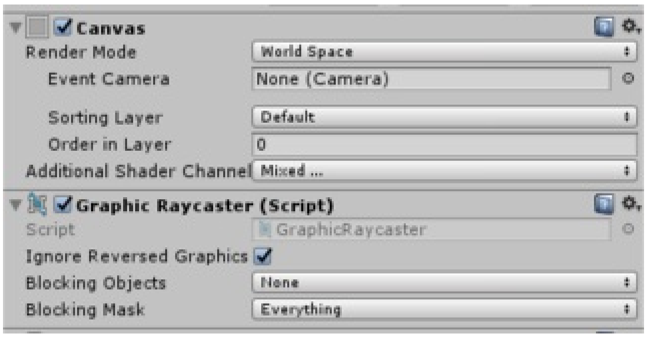
Feedback Canvas

Displays a "virtual tablet" in your Scene, which can display Tabs. Uses a UI Canvas for the base, to which Tabs can be added. the Feedback Canvas has the following GameObjects: the *Background*, a *Tab Label Bar* to display Tab titles along the top of the UI and a *Tab Container* to hold Tab GameObjects.

Components of the Feedback Canvas

Canvas

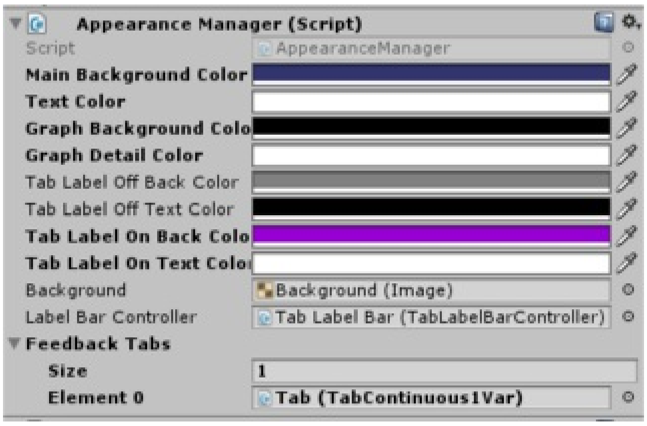
Allows *UI Panels* (i.e. Tabs) to be added to the UI. *Note: Should have `Render Mode` set to `World Space`*.



☑ Canvas	
Render Mode	World Space
Event Camera	None (Camera)
Sorting Layer	Default
Order in Layer	0
Additional Shader Channel	Mixed ...
☑ Graphic Raycaster (Script)	
Script	GraphicRaycaster
Ignore Reversed Graphics	☑
Blocking Objects	None
Blocking Mask	Everything

Appearance Manager

Allows you to easily change the color scheme of the entire Feedback Canvas and all of its Tabs. Appearance updates whenever an Appearance Manager value is changed (in OnValidate).



Appearance Manager (Script)	
Script	AppearanceManager
Main Background Color	🟦
Text Color	⬜
Graph Background Colo	⬛
Graph Detail Color	⬜
Tab Label Off Back Color	▨
Tab Label Off Text Color	⬛
Tab Label On Back Colo	🟪
Tab Label On Text Colo	⬜
Background	Background (Image)
Label Bar Controller	Tab Label Bar (TabLabelBarController)
Feedback Tabs	
Size	1
Element 0	Tab (TabContinuous1Var)

Main Background Color

The background color of the Canvas

Text Color

The color of text in the Tabs in the Canvas, i.e. the title and the list of recent values.

Graph Background Color

The background color of the Graphs in the Tabs in the Canvas.

Graph Detail Color

The color of details of the Graphs in the Tabs in the Canvas, such as Axes and all of the text on the Graph.

Tab Label Off Back Color

The background color of Tab Labels (along the top of the Canvas) whose Tabs are not currently visible on the Canvas.

Tab Label Off Text Color

The text color of Tab Labels (along the top of the Canvas) whose Tabs are not currently visible on the Canvas.

Tab Label On Back Color

The background color of Tab Label (along the top of the Canvas) whose Tab is currently displayed on the Canvas.

Tab Label On Text Color

The text color of Tab Label (along the top of the Canvas) whose Tab is currently displayed on the Canvas.

Tab Background

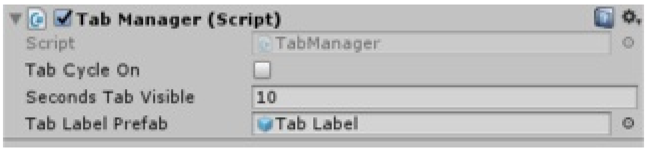
The background Image. This does not need to be modified. Label Bar Controller The controller script for the Canvas Label Bar. This does not neet to be modified.

Feedback Tabs

A list of the Tabs currently in the Tab Container of the Canvas. This should update itself when any of the Appearance Manager variables are modified.

Tab Manager

Initializes and controls display of Tabs and Tab Labels. This allows the rotation between multiple Tabs on the same Canvas to be controlled.



☑ Tab Manager (Script)

Script	TabManager
Tab Cycle On	☐
Seconds Tab Visible	10
Tab Label Prefab	Tab Label

Tab Cycle On

Choose whether to cycle between Tabs in the Canvas. If this is **on**, each Tab will be displayed for *Seconds Tab Visible*, in top-down order in the Tab Container. If this is **off**, it will only display the top Tab in the Tab Container. Inactive Tabs are ignored.

Seconds Tab Visible

The seconds for which each Tab is visible if *Tab Cycle On* is enabled. Must be an integer value. If value is equal to or less than zero, Tabs will not cycle.

Tab Label Prefab

The Prefab from which all Tab Labels are instantiated. Does not need to be modified.

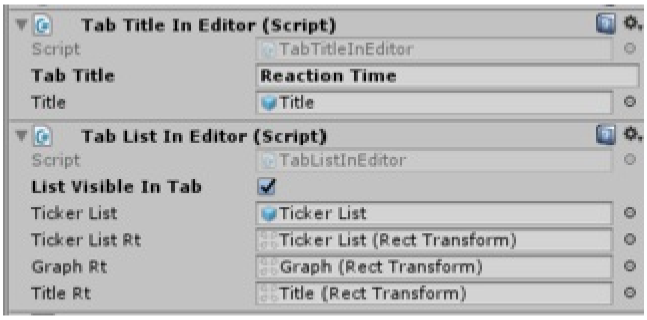
Tab

Based on a UI Panel, the Tab Prefab has a built-in structure of the following UI Panels:

- the *Graph* panel
- the *Graph Title* panel
- optionally, a *Ticker List* panel displaying a list of the most recent values.

All of the Tab values can be changed without needing to access any child GameObjects. The Tab automatically resizes to fit the size of the Feedback Canvas it is in. It requires a *Graph-Specific Tab Component* to be added to it in order display a Graph.

General Tab Components



Tab Title In Editor (Script)

Script	TabTitleInEditor
Tab Title	Reaction Time
Title	Title

Tab List In Editor (Script)

Script	TabListInEditor
List Visible In Tab	☑
Ticker List	Ticker List
Ticker List Rt	Ticker List (Rect Transform)
Graph Rt	Graph (Rect Transform)
Title Rt	Title (Rect Transform)

Tab Title In Editor

Tab Title

The title of the graph which the tab contains. Is displayed above the graph and on the Tab Label. In *Tab Continuous 1 Var*, this is also used on the X and Y Axis Titles as the name of the variable.

Title

The Title Panel. This variable does not need to be changed.

Tab List In Editor

List Visible In Tab

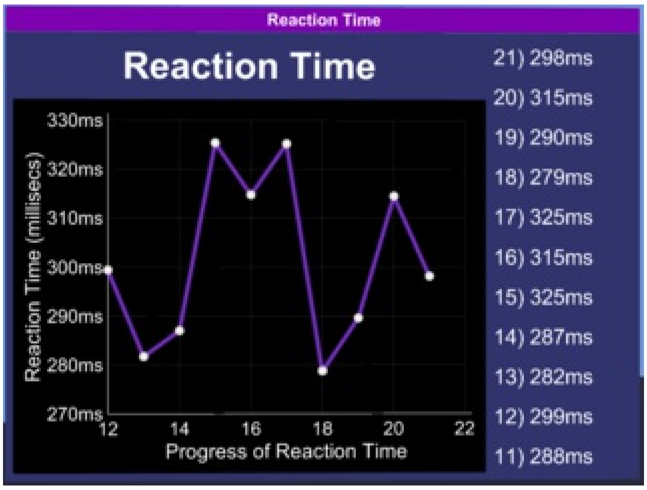
Toggles whether the Ticker List (list of most recent values) is visible in the Tab.

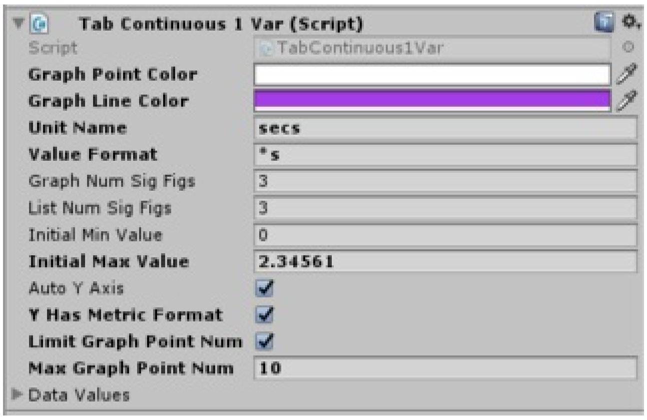
Ticker List and *Rect Transforms*

references to Tab child objects. These variables do not need to be modified.

Graph-Specific Tab Components

Tab Continuous 1 Var (Progressive Graph)





Tab Continuous 1 Var (Script)

Script	TabContinuous1Var
Graph Point Color	
Graph Line Color	
Unit Name	secs
Value Format	*s
Graph Num Sig Figs	3
List Num Sig Figs	3
Initial Min Value	0
Initial Max Value	2.34561
Auto Y Axis	☑
Y Has Metric Format	☑
Limit Graph Point Num	☑
Max Graph Point Num	10
▶ Data Values	

Graph Point Color

The color of points on the graph.

Graph Line Color

The color of the line connecting points on the Graph.

Has Point Connect Line

Toggles the line on the Graph which connects the points on or off.

Unit Name

The name of the unit of the variable. Displayed on the Y Axis title.

Value Format

The format in which the Y Axis labels will be displayed. `*` or `{0}` will be replaced with the value. e.g. `*s` will display all labels with an "s" after them. Useful for unit symbols.

Graph Num Sig Figs

The number of significant figures to which Y Axis labels will be displayed. Extra significant figures may be automatically added if no change in value can be seen between two labels.

List Num Sig Figs

The number of significant figures the list of most recent values will be displayed to.

Initial Min Value

The initial minimum value of the Y Axis. This is "initial" because the minimum may change on start-up to fit a suitable number system (e.g. `0 to 8`, not `0.03 to 8`). If *Auto Y Axis* is enabled, it may also change when extra points are added.

Initial Max Value

The initial maximum value of the Y Axis. This is "initial" because the maximum may change on start-up to fit a suitable number system (e.g. `0 to 8`, not `0 to 7.89`). If *Auto Y Axis* is enabled, it may also change when extra points are added.

Auto Y Axis

Toggles whether to automatically adjust Y Axis minimum and maximum to fit points on the graph. This will take into account having a readable number system (e.g. `5 to 15`, not `6.67 to 14.89`). The labels will automatically reformat to be of the optimal interval and quantity.

Y Has Metric Format

If Y Axis labels are less than 1 or more than 1000, this will re-format them to include a metric prefix, such as *k* or *m*. It will also add the full prefix to the unit name, such as *kilometers* or *millimeters* instead of *meters*.

Limit Graph Point Num

This will limit the history of the graph to the value of *Max Graph Point Num*. Any values before this will be removed from the Graph, though not from the Tab data.

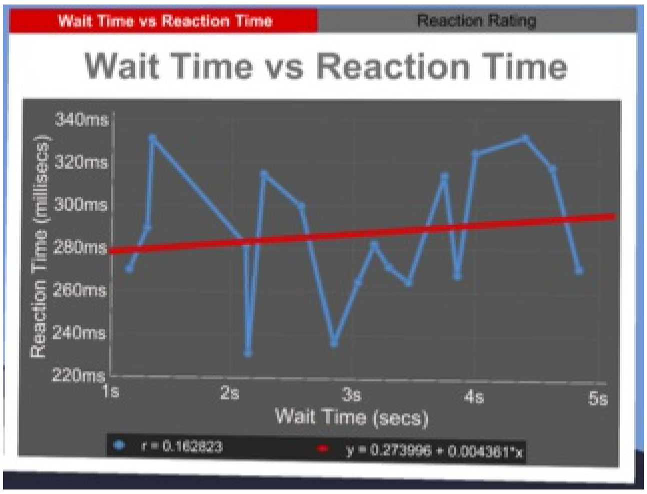
Max Graph Point Num

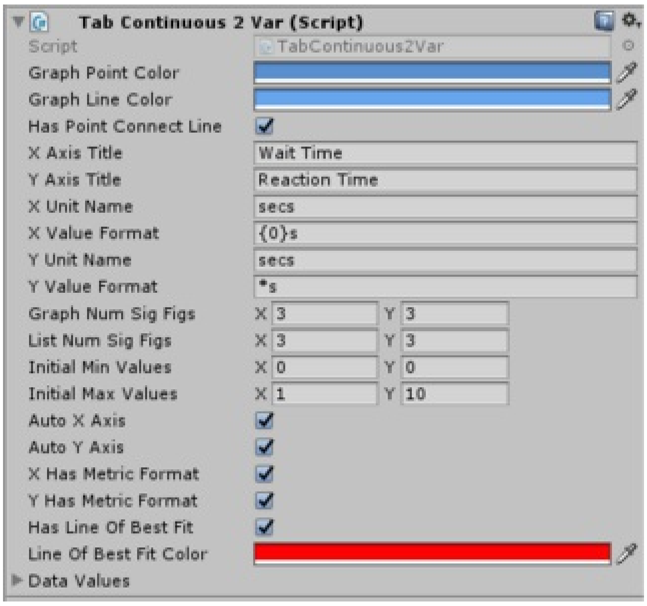
The number of points which the Graph history is limited to if *Limit Graph Point Num* is enabled.

Data Values

A List of type `float` storing values for all the data added to the Tab. This is **not** necessarily the same as the values shown on the graph, and should not be written to or changed directly.

Tab Continuous 2 Var (Scatter Graph)





Tab Continuous 2 Var (Script)

Script	TabContinuous2Var	
Graph Point Color	🟦	
Graph Line Color	🟦	
Has Point Connect Line	☑	
X Axis Title	Wait Time	
Y Axis Title	Reaction Time	
X Unit Name	secs	
X Value Format	{0}s	
Y Unit Name	secs	
Y Value Format	*s	
Graph Num Sig Figs	X 3	Y 3
List Num Sig Figs	X 3	Y 3
Initial Min Values	X 0	Y 0
Initial Max Values	X 1	Y 10
Auto X Axis	☑	
Auto Y Axis	☑	
X Has Metric Format	☑	
Y Has Metric Format	☑	
Has Line Of Best Fit	☑	
Line Of Best Fit Color	🟥	
Data Values		

Graph Point Color

The color of points on the graph.

Graph Line Color

The color of the Point Connect Line on the Graph.

Has Point Connect Line

Toggles the line on the Graph which connects the points on or off.

X Axis Title

The name of the variable displayed on the X Axis. Will be displayed in the X Axis title.

Y Axis Title

The name of the variable displayed on the Y Axis. Will be displayed in the Y Axis title.

X Unit Name

The name of the unit of the X variable. Displayed on the X Axis title.

X Value Format

The format in which the X Axis labels will be displayed. `*` or `{0}` will be replaced with the value. e.g. `*s` will display all labels with an "s" after them. Useful for unit symbols.

Y Unit Name

The name of the unit of the Y variable. Displayed on the Y Axis title.

Y Value Format

The format in which the Y Axis labels will be displayed. `*` or `{0}` will be replaced with the value. e.g. `*s` will display all labels with an "s" after them. Useful for unit symbols.

Graph Num Sig Figs

The number of significant figures to which Axis labels will be displayed. Extra significant figures may be automatically added if no change in value can be seen between two labels.

List Num Sig Figs

The number of significant figures the list of most recent values will be displayed to.

Initial Min Values

The initial minimum values of the Axes. This is "initial" because the minimums may change on start-up to fit a suitable number system (e.g. `0 to 8`, not `0.03 to 8`). If *Auto Y Axis* is enabled, it may also change when extra points are added.

Initial Max Values

The initial maximum value of the Axes. This is "initial" because the maximums may change on start-up to fit a suitable number system (e.g. `0 to 8`, not `0 to 7.89`). If *Auto Y Axis* is enabled, it may also change when extra points are added.

Auto X Axis

Toggles whether to automatically adjust X Axis minimum and maximum to fit points on the graph. This will take into account having a readable number system (e.g. `5 to 15`, not `6.67 to 14.89`). The labels will automatically reformat to be of the optimal interval and quantity.

Auto Y Axis

Toggles whether to automatically adjust Y Axis minimum and maximum to fit points on the graph. This will take into account having a readable number system (e.g. `5 to 15`, not `6.67 to 14.89`). The labels will automatically reformat to be of the optimal interval and quantity.

X Has Metric Format

If X Axis labels are less than 1 or more than 1000, this will re-format them to include a metric prefix, such as *k* or *m*. It will also add the full prefix to the unit name, such as *kilometers* or *millimeters* instead of *meters*.

Y Has Metric Format

If Y Axis labels are less than 1 or more than 1000, this will re-format them to include a metric prefix, such as *k* or *m*. It will also add the full prefix to the unit name, such as *kilometers* or *millimeters* instead of *meters*.

Has Line Of Best Fit

Toggles a Linear Regression Line, Y on X, on or off. When enabled, this also displays the Regression Equation in the form `y = mx + c` and Pearson's Product Moment Correlation Coefficient `r`.

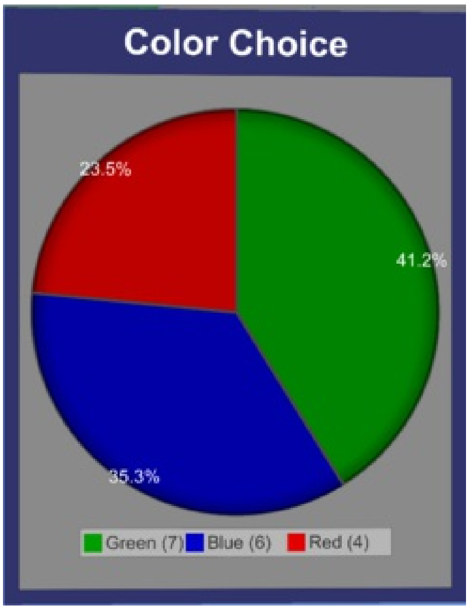
Line Of Best Fit Color

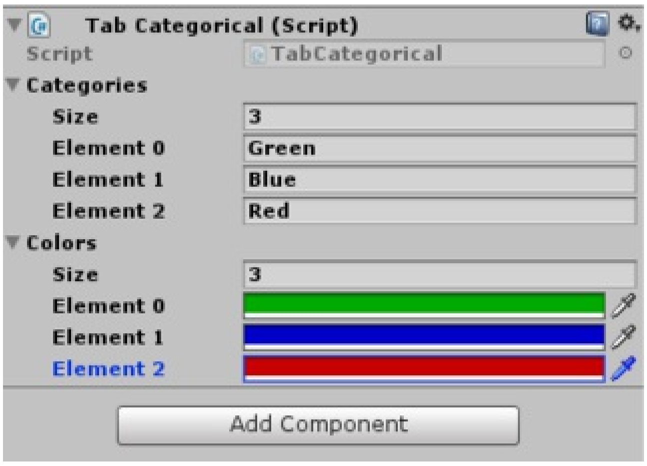
The color of above mentioned Linear Regression Line.

Data Values

A List of type `Vector2` storing values for all the data added to the Tab. This is **not** necessarily the same as the values shown on the graph, and should not be written to or changed directly.

Tab Categorical (Pie Graph)




Tab Categorical (Script)
Script: TabCategorical
Categories
Size: 3
Element 0: Green
Element 1: Blue
Element 2: Red
Colors
Size: 3
Element 0: 🟩
Element 1: 🟦
Element 2: 🟥

Categories

The names of the categories in the Tab. When these categories are referenced in the code, the strings used to reference them must be *identical* to the strings written here.

Colors

Optionally, this can be used to set the colors of the categories. The lists are parallel, so Element 2 in Colors will set the color for Element 2 in Categories. Colors list must be the same length as Categories list for the colors to be applied. To **not** use the Colors list, leave its length at `0`.

Tab Functions

All Tab functions require the relevant Tab reference created in the code (`TabContinuous1Var`, `TabContinuous2Var` or `TabCategorical`) to be linked to the Tab by a click-and-drag in the Inspector.

One Variable: AddValue

Adds a point to the graph, with with the X Axis chronological and the Y Axis displaying the given value. To use with *Tab Continuous 1 Var*

One Variable: Parameters

- `float newValue: The value to be added to the graph.`

One Variable: Code in C Sharp

```
TabContinuous1Var progressTab;
float someValue;
...
// some code which includes getting value for someValue
...
progressTab.AddValue(someValue)
```

Two Variable: AddPoint

Adds a point to a Scatter Graph, with the X and Y Axes displaying data fr two different variables. To use with *Tab Continuous 2 Var*.

Two Variable: Parameters

- `newXValue: the x coordinate of the point to be added.`
- `newYValue: the y coordinate of the point to be added.`

Two Variable: Code in C Sharp

```
TabContinuous2Var twoVarTab;
float someXValue;
float someYValue;
...
// some code that includes getting someXValue and someYValue
...
twoVarTab.AddPoint(someXValue, someYValue);
```

Categorical: AddToCategory

Adds one to an existing category on the categorical graph (currently a Pie Graph). Optional parameter to choose what value to add to that category (default 1). To use with *Tab Categorical*.

Categorical: Parameters

- `string category: category to add a value to.`
- `int value: value to add to that category [default 1]`

Warning: Category strings referenced in the code must be *identical* to those specified in the Tab in the Inspector!

Categorical: Code in C Sharp

```
TabCategorical pieTab;
string someCategory;
int someAmount;
...
// some code which includes setting someCategory to one of pieTab's existing
categories
...
pieTab.AddToCategory(someCategory);
// OR define some amount to add to the category
pieTab.AddToCategory(someCategory, someAmount);
```

Clear

Clears all data on a tab and resets it to its original state at the beginning of runtime. Takes no parameters. Can be implemented the same way on all Tabs.

Clear: Code in C Sharp

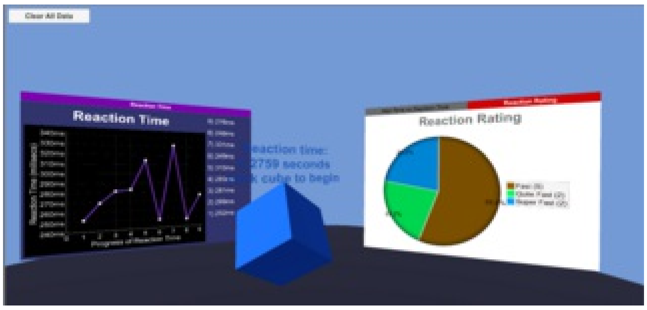
```
// create tab, etc.
...
someTab.Clear();
```

Common Issues

When this Asset has been more frequently used, this section will record commonly encountered issues and how to fix them.

Examples

Button Clicker



This is an example project, where the user waits for the rotating cube to turn green, then clicks on it as fast as possible. The Scene contains two Feedback Canvases to demonstrate how they would work. Use the arrow keys, Space Bar and Left Shift during runtime to move and view the User Interfaces from different angles. You can see how to control a Tab through a C# Script in the *Button Control* script. The places where Tabs are referenced have been highlighted.

VR Golf Experiment

This is an experiment run at University of Leeds, where a Virtual Reality Golf game is used to research motor skills and motor learning. (...)

Future work

- Some way to switch between tabs on a UI in Virtual Reality other than a timed rotation between them.
- Functionality to allow users to remove values from a graph.
- An improved system for working out the spacing of labels on the X Axis of the Scattergraph.
- Add additional types of graphs.
- Versions to work with other Graphing Assets (primarily the Graph scripts will need to be rewritten for this).