

UIToolkit Script Components Manual

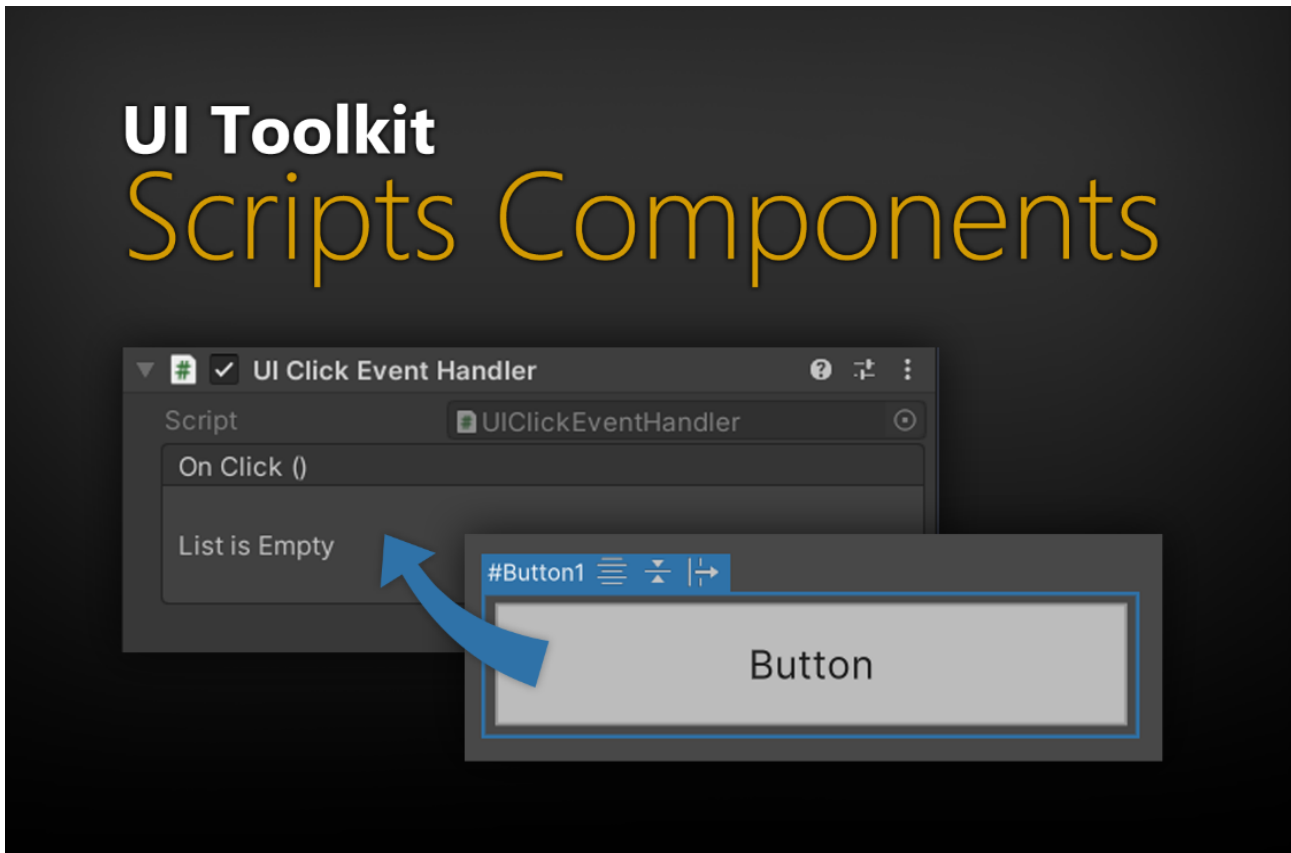


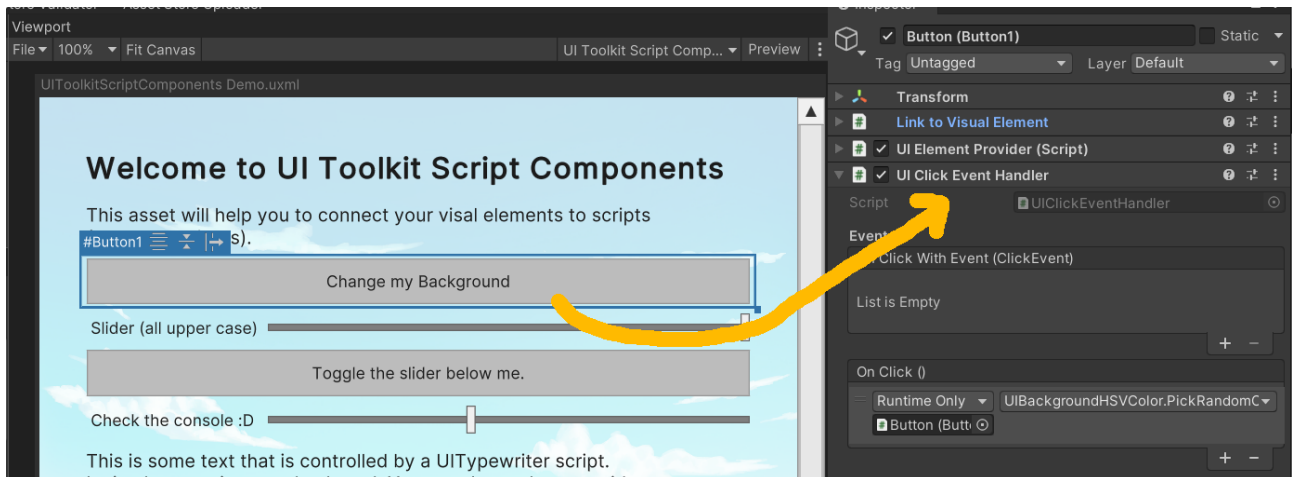
Table of contents

| | |
|--|---|
| What's it good for?..... | 2 |
| Usage..... | 3 |
| Creating a LINK between a UI Element and a GameObject..... | 3 |
| Adding Components..... | 3 |
| Using a QUERY (instead of a LINK)..... | 5 |
| Details on the „LinkToVisualElement“ component..... | 7 |

What's it good for?

In the UI Toolkit there are no more GameObjects in the UI and thus it's impossible to add components directly to a UI element.

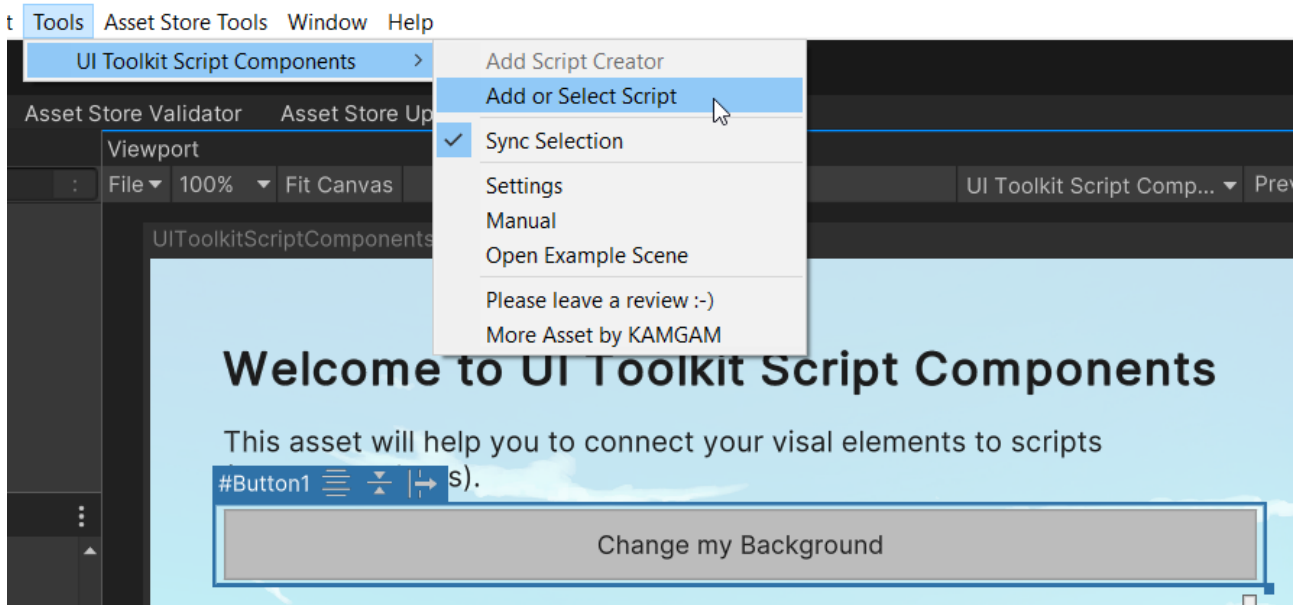
This tool generates a GameObject for each element and links it to a UI element. You can then add components to that GameObject.



Usage

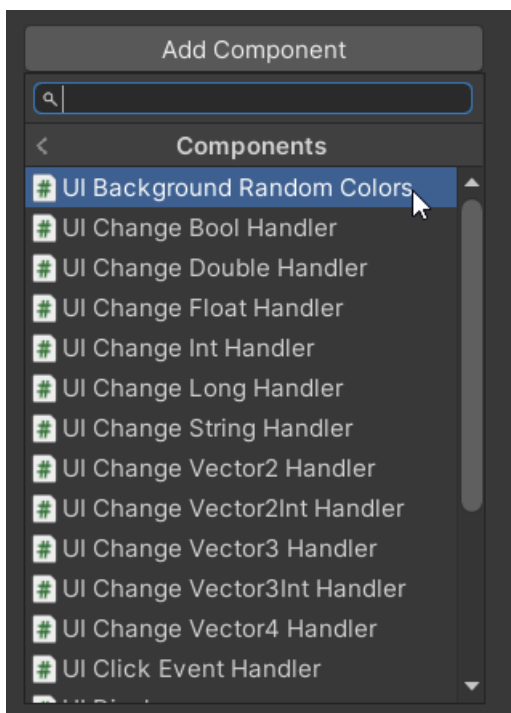
Creating a LINK between a UI Element and a GameObject

- 1) Select the element you want to link to.
- 2) Right-Click on the UIDocument in your scene and choose "UI Toolkit > Add or Select Script"
OR use the „Tools > UI Toolkit Script Components > Add or Select Script“ menu.

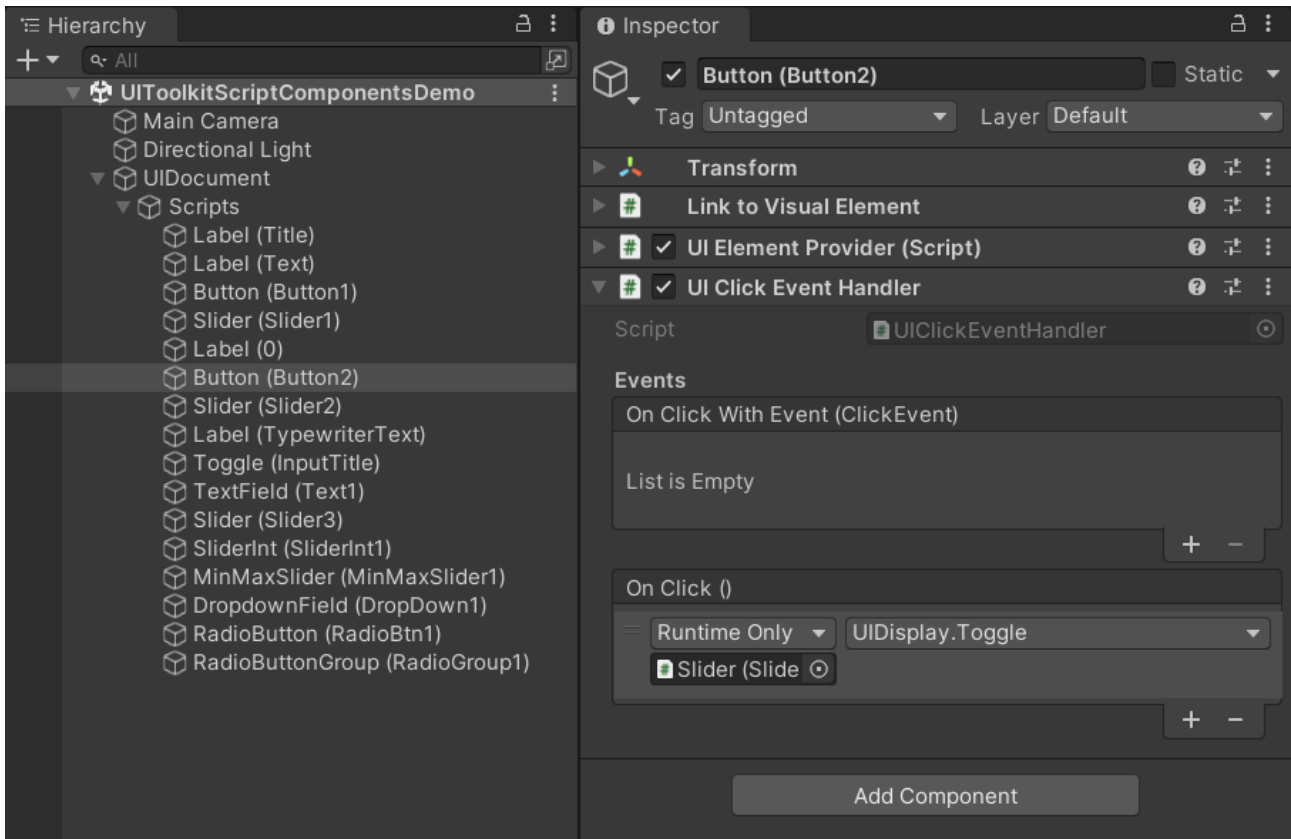


Adding Components


Once you have set up your link you can add components. You can choose one from the predefined components under „UI Toolkit > Scripts > Components“ or make your own.




Here is the „Click Event Handler“ for example:



You may have noticed that by adding the „UIClickEventHandler“ you have also got a UIElementProvider component. The reason is that the UIClickEventHandler actually needs the provider to fetch the linked element(s). The provider again needs either a LinkToVisualElement (if only one element should be affected) or a QueryForVisualElements component (if multiple should be affected). More on the distinction between those later.

 HINT: If you are a programmer then you can use the „UIElementProvider“ to access the linked elements (.Elements list) or you simply extend the UIComponentBase<T> class. The provider merges all elements found by the link and query into one list for easy access (links are always first).

 HINT: There is a handy „LogTest()“ method on every event handler component. You can use it to quickly check if your setup works.

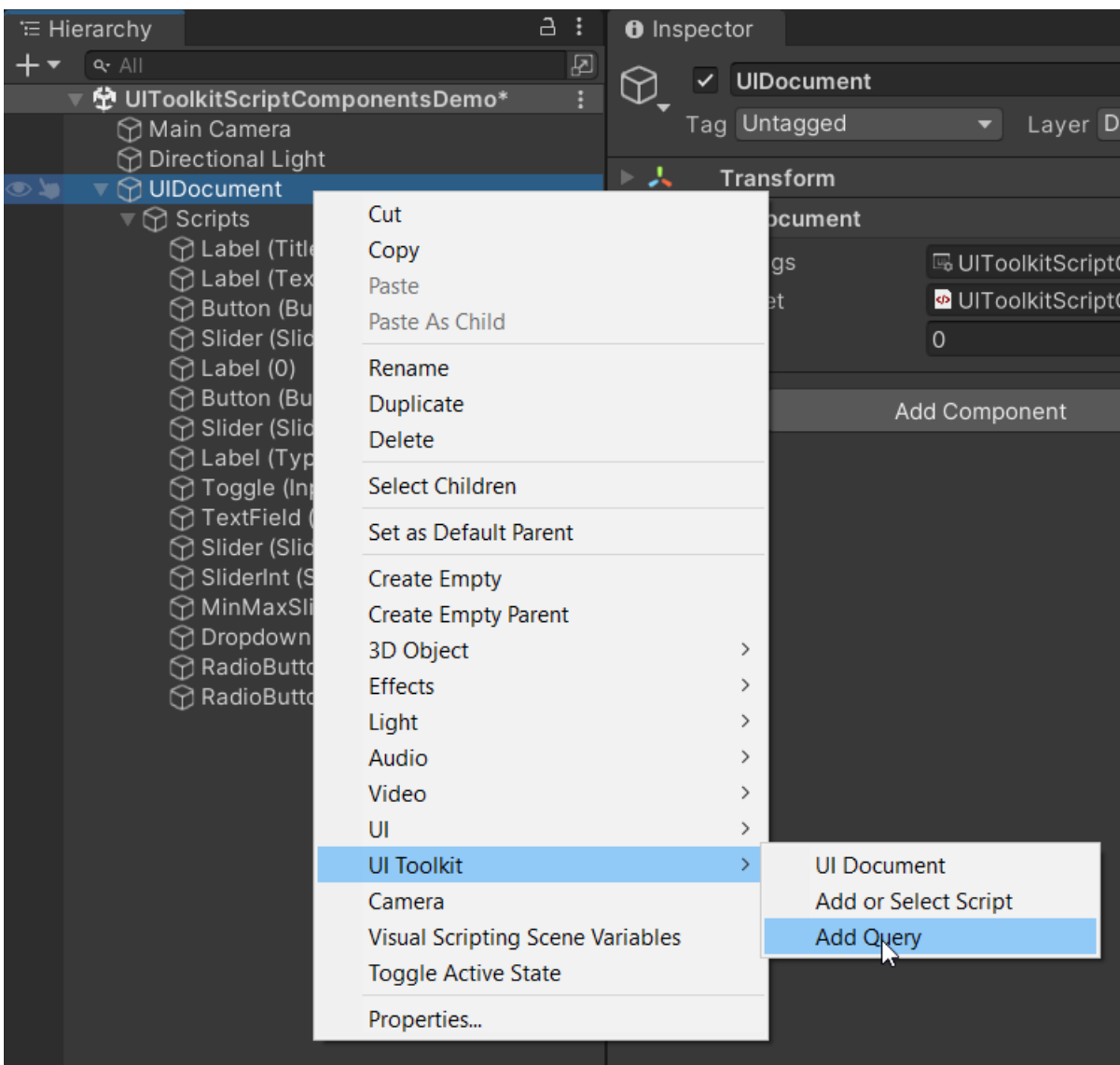
Using a QUERY (instead of a LINK).

You may not always want to create a new link object for each and every UI element. For example if there are many similar elements (like buttons). Think of queries like a search on the web. You enter the criteria and you will get one or more results.

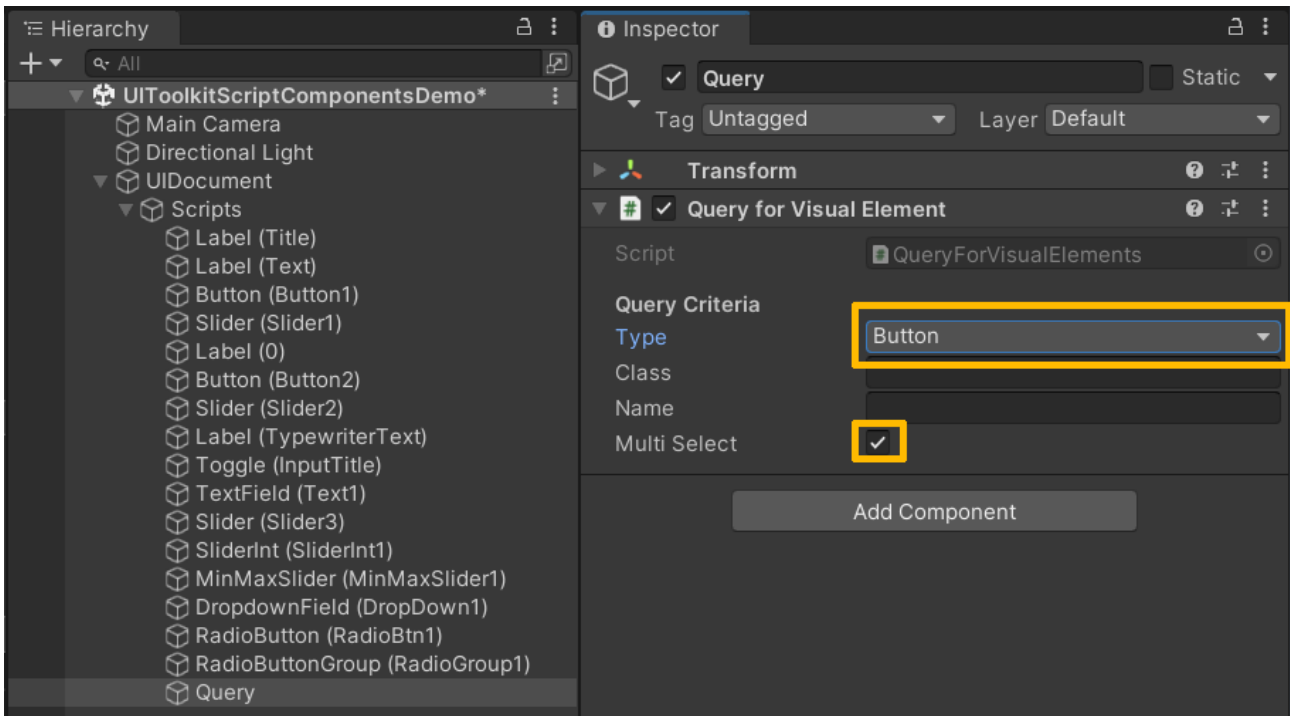
In this example we will add a script which logs a message if ANY button is pressed. We will only use a single game object to achieve that.

Let's start by adding a query game object:

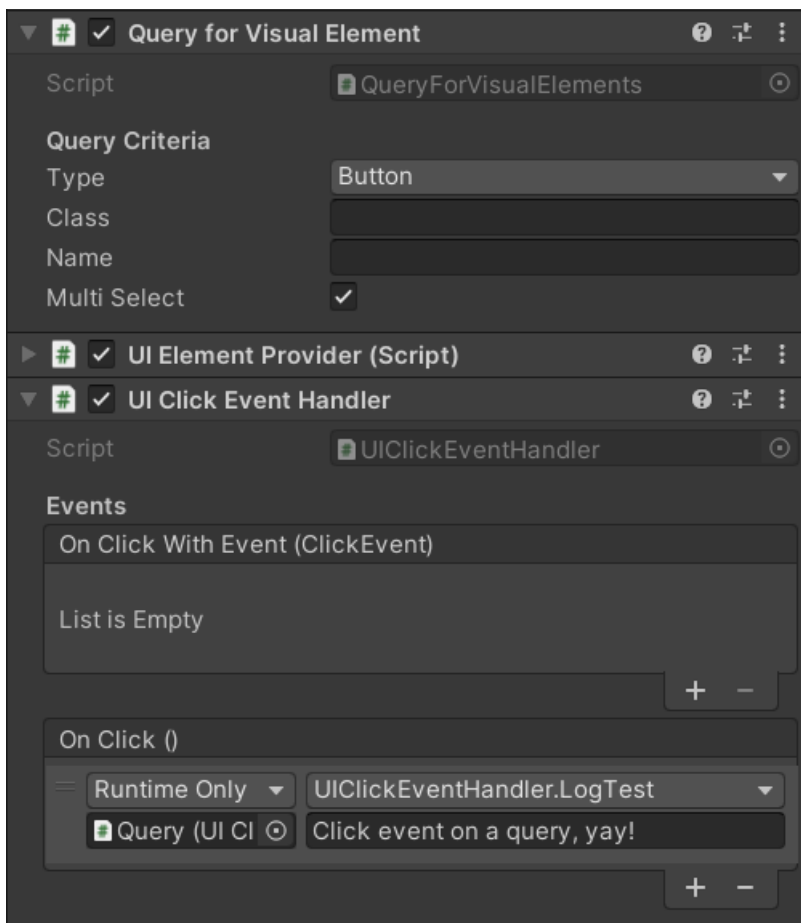
„Right-Click > UI Toolkit > Add Query“ on the UIDocument



Select „Button“ for the type and enable the „Multi Select“ so we will get more than one result.



The remaining workflow is the same as for linked objects. Let's use the „UI Click Event Handler“ and configure it to log a message.



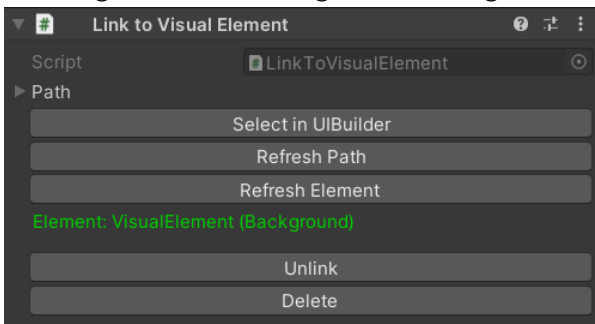
And that's all you have to do. Easy, right?

Details on the „LinkToVisualElement“ component

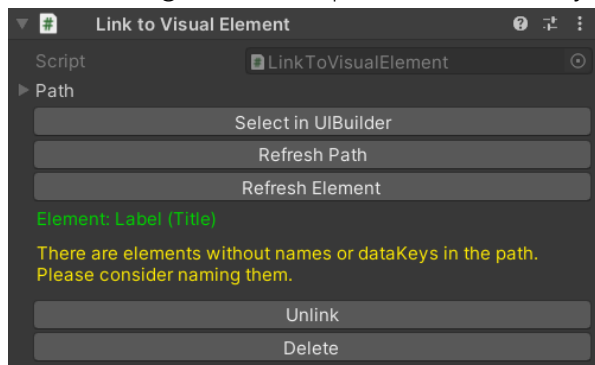
The link component matches one (and only one) element. However this single element is usually instantiated twice. Once in the UI Builder and once in UIDocument. It helps to understand that actually there are TWO instances of the UI hierarchy. One is used by the UI Builder window (the stuff you click on while editing) and the other one is used by the UI Document in your scene (that's the one shown in the Game View).

The link component keeps track of those and stores some persistent linking information so it can find these elements. That info is called the „path“. In the inspector the link shows some information on the current state of the path (can it find the elements for the path in the builder and on the document).

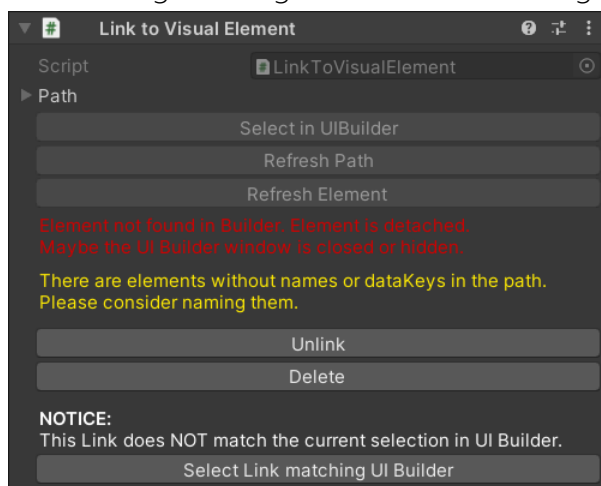
If all is good it shows a green message:



If some things can be improved it shows a yellow message:



If something is wrong it shows a red message:



The path of an element can get rather long. The longer the path the easier it is for it to break. Therefore it is best to use unique names for your elements. If the name is unique the path resolution can short-circuit to using only the name, which is much more reliable (and faster).

