UI Toolkit Playmaker Integration

Table of contents

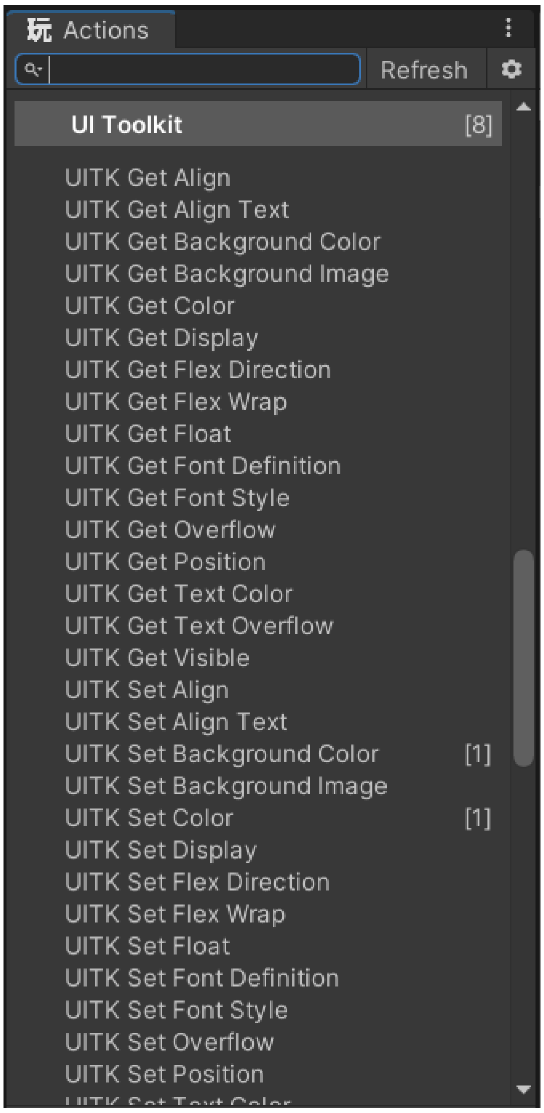
Requirements & Setup

Requirements

Unity 2021.2 or higher is required since that is when Unity added the UI Toolkit Module for runtime use. If you can, please upgrade to the highest LTS version of Unity. The newer the version the less „glitches" the UI Toolkit has.

Keep in mind, UI Toolkit as a whole is still a work in progress and not quite ready for prime time. Unity itself still recommends using UGUI instead of UI Toolkit for runtime applications (source).

PlayMaker (1.9.0 or higher is recommended though it may work with earlier version too)



UI Toolkit Actions

This secion describes the most important Actions of the UI Toolkit integration. Notice: There are more (most of them are pretty self explanatory).

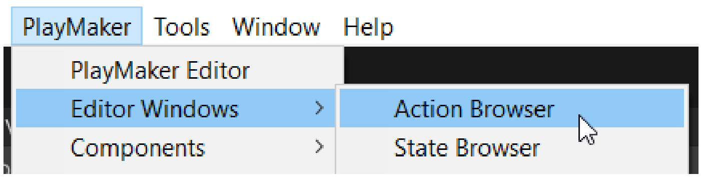
Almost all UI Toolkit actions require a visual element to work.

The process of getting hold of a visual element in UI Toolkit is a bit different than the usual Unity way. Since visual elements are not located in the scene hierarchy you will have to use a query on your UI Document.

Queries are how elements are found in UI Toolkit.

How to add a UI Toolkit Action

1) Open the PlayMaker **Actions Browser**



2) Type in „UITK query" for UI Toolkit queries. This will filter the actions to queries.

HINT: All UI Toolkit actions a prefixed with „UITK". Typing that in first will already filter out anything else.

3) Choose the query action and **add it to your state**.

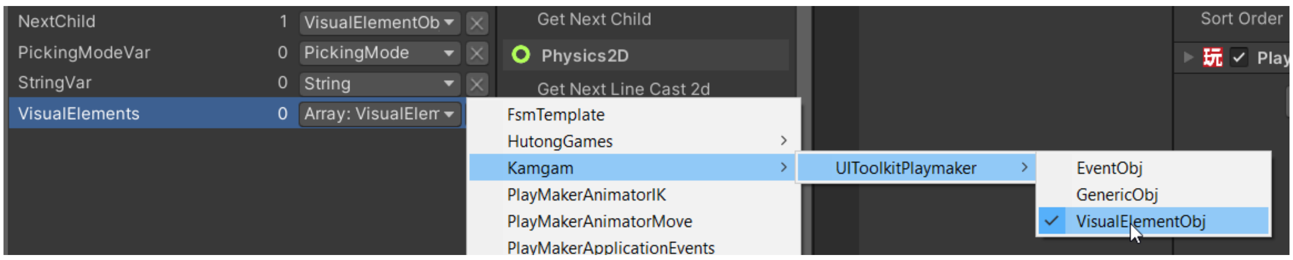
HINT: The „ButtonClickDemo" in the Examples contains this exact setup.

There are two main types of query actions: „RegisterEvent" queries and „SetVariable" queries.

What variable type to use?

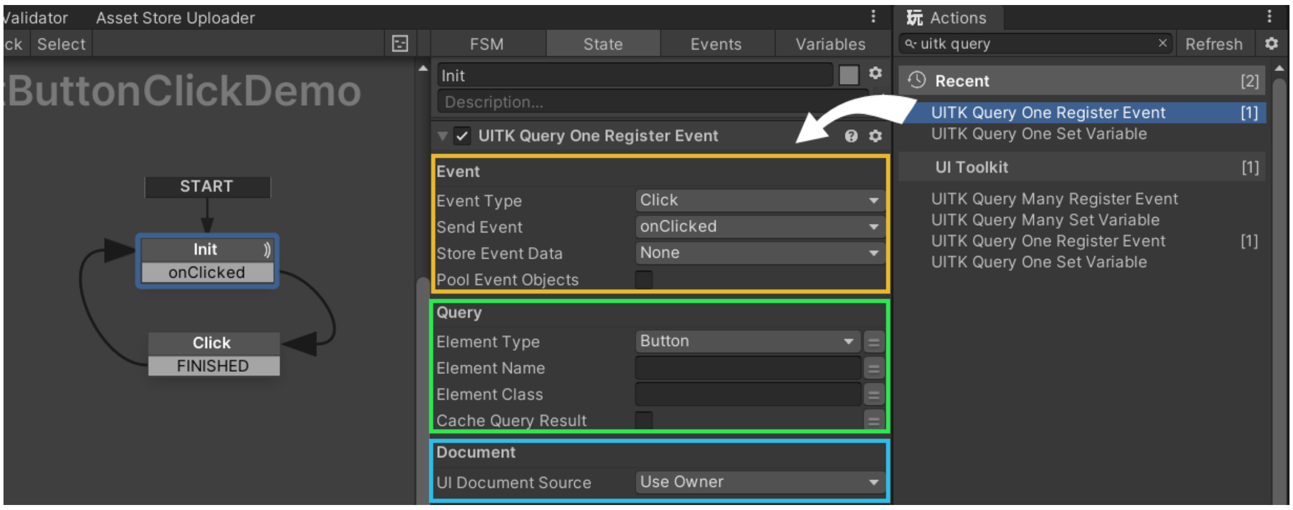
To store the queried elements you will need to use an Object variable with an Object Type set to „**VisualElementObj**".

Object → Kamgam → UIToolkitPlaymaker → VisualElementObj



UITK Query One Register Event

RegisterEvent queries will **query** the **UI Document** for one (or more) elements and immediately **register an event** callback that triggers a PlayMaker event.



Init

Description...

UITK Query One Register Event

Event

Event Type: Click

Send Event: onClicked

Store Event Data: None

Pool Event Objects

Query

Element Type: Button

Element Name

Element Class

Cache Query Result

Document

UI Document Source: Use Owner

uitk query

Recent [2]

UITK Query One Register Event [1]

UITK Query One Set Variable

UI Toolkit [1]

UITK Query Many Register Event

UITK Query Many Set Variable

UITK Query One Register Event [1]

UITK Query One Set Variable

Event:

Event Type: A list of visual element events you can register to.

HINT: You may have noticed that „Drap" events are not in there. Sadly these are EditorOnly in UI Toolkit. Once Unity makes them runtime compatible they will be added.

Send Event: The PlayMaker event that should be triggered.

Store Event Data: Variable where the event data from the visual element callback will be stored.

Pool Event Objects: If enabled then event objects are reused the next time the event is triggered. This is done to preserve memory. Enable only if you see a need for it in the profiler. Usually its fine to leave this turned off.

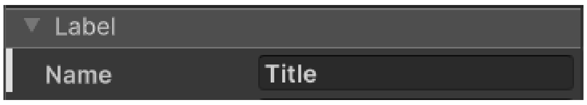
Query:

If you are not yet familiar with how queries work in UI Toolkit then please read up on them in the Unity Manual first. The options here correspond to those mentioned in the manual.

Element Type: A list of visual element types like „Button", „Label", ...

HINT: You can use the „VisualElement" type to allow all element types. Useful if you want to filter only by Name or ClassName.

Element Name: The name of the element as defined in the UI Builder or UXML. If left empty then this will be ignored.



Label

Name: Title

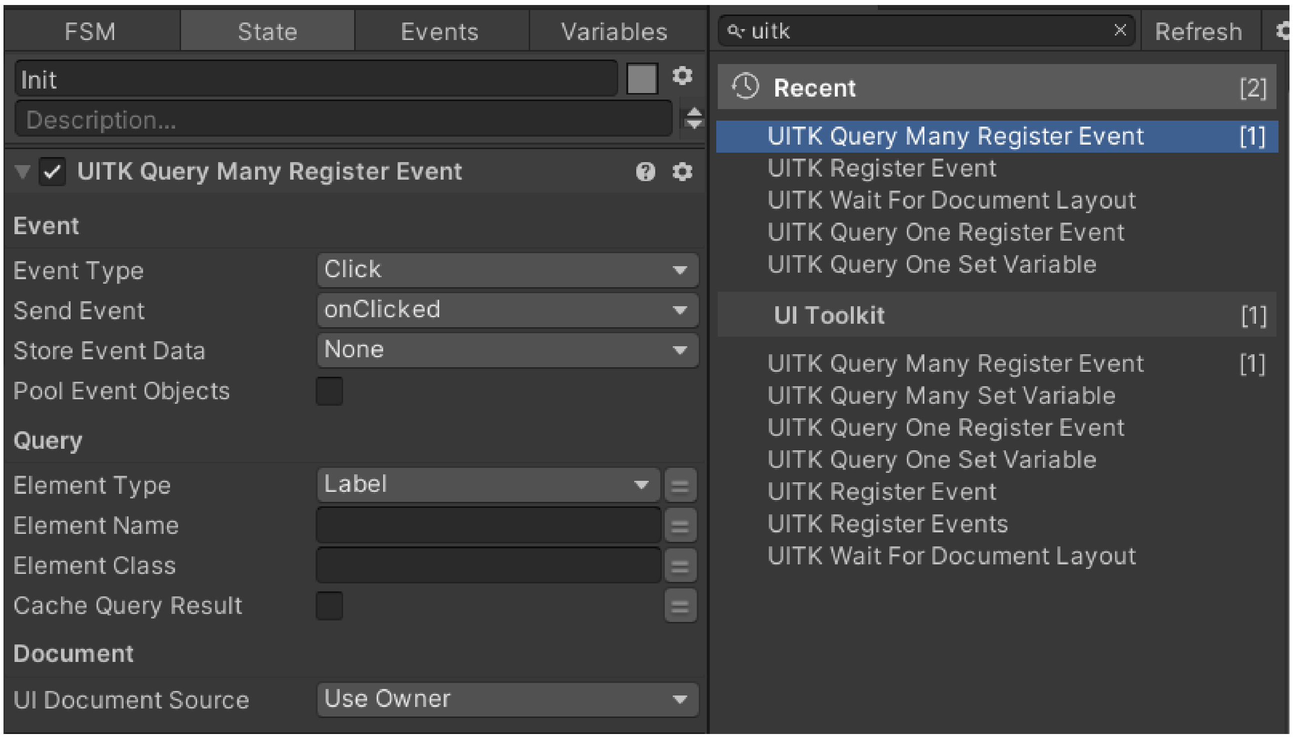
Element Class: The USS class name of the element. If left empty then this will be ignored.

Document:

UI Document Source: The game object with the UI Document that should be used for queries. The action will search for the UIDocument component on the given game object. Usually „Use Owner" is fine but you can also specify another game object or the UI Document directly.

UITK Query Many Register Event

This works just like the QueryOneRegisterEvent except that it queries many elements and registers the event callback on each of them.


FSM
State
Events
Variables
Init
Description...
uitk
Refresh
Recent [2]
UITK Query Many Register Event [1]
UITK Register Event
UITK Wait For Document Layout
UITK Query One Register Event
UITK Query One Set Variable
UI Toolkit [1]
UITK Query Many Register Event [1]
UITK Query Many Set Variable
UITK Query One Register Event
UITK Query One Set Variable
UITK Register Event
UITK Register Events
UITK Wait For Document Layout

UITK Query Many Register Event

Event

Event Type	Click
Send Event	onClicked
Store Event Data	None
Pool Event Objects	☐

Query

Element Type	Label
Element Name	
Element Class	
Cache Query Result	☐

Document

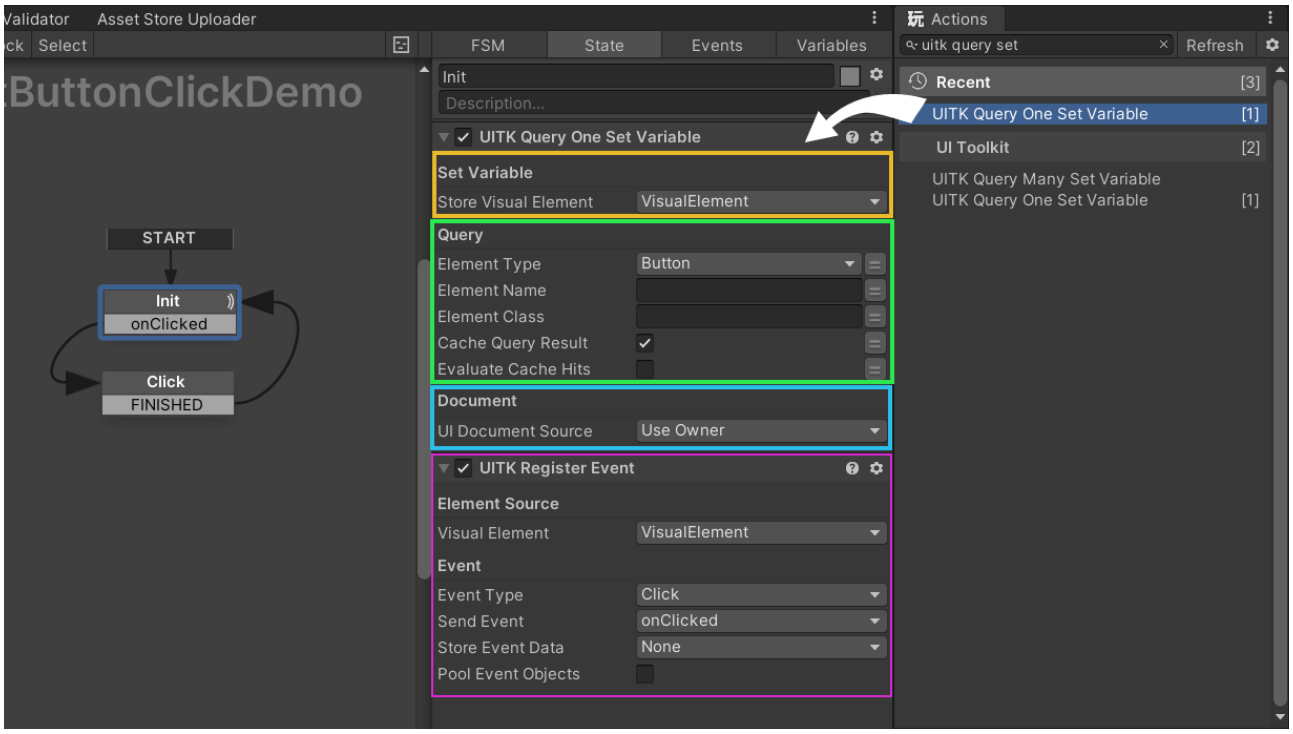
UI Document Source	Use Owner

UITK Query One Set Variable

SetVariable queries will **query** the **UI Document** for one (or more) elements and store the result(s) in a PlayMaker **variable**.

HINT: VisualElements are stored as Object variables in PlayMaker because PlayMaker can not (yet) store VisualElements directly in variable.

The example below has the same effect as the **RegisterEvent** query above. The only difference is that it uses a variable to store the visual element and then uses a separate **„Register Event“ action** to add the event callback.



Init
Description...

UITK Query One Set Variable

Set Variable
Store Visual Element: VisualElement

Query
Element Type: Button
Element Name:
Element Class:
Cache Query Result: ✓
Evaluate Cache Hits:

Document
UI Document Source: Use Owner

UITK Register Event

Element Source
Visual Element: VisualElement

Event
Event Type: Click
Send Event: onClicked
Store Event Data: None
Pool Event Objects:

Set Variable:

Store Visual Element: The variable where the result of the query will be stored.

HINT: The result actually is a wrapper around a VisualElement called „VisualElementObj“. You can read more on why this is done here.

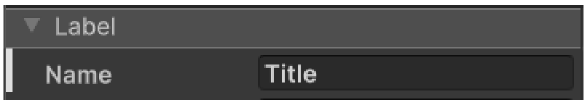
Query:

If you are not yet familiar with how queries work in UI Toolkit then please read up on them in the Unity Manual first. The options here correspond to those mentioned in the manual.

Element Type: A list of visual element types like „Button“, „Label“, ….

HINT: You can use the „VisualElement“ type to allow all element types. Useful if you want to filter only by Name or ClassName.

Element Name: The name of the element as defined in the UI Builder or UXML. If left empty then this will be ignored.



Label

Name: Title

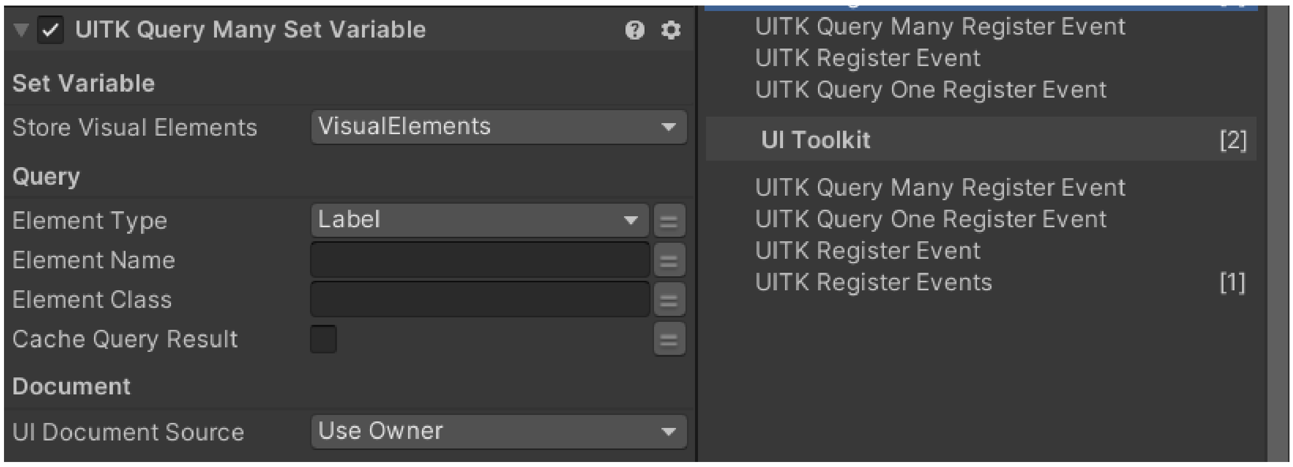
Element Class: The USS class name of the element. If left empty then this will be ignored.

__Document:__

UI Document Source: The game object with the UI Document that should be used for queries. The action will search for the UIDocument component on the given game object. Usually „Use Owner" is fine but you can also specify another game object or the UI Document directly.

UITK Query Many Register Event

This works like the QueryOneRegisterEvent except that it stores a list of elements in the variable.



UITK Query Many Set Variable

Set Variable

Store Visual Elements: VisualElements

Query

Element Type: Label

Element Name:

Element Class:

Cache Query Result: ☐

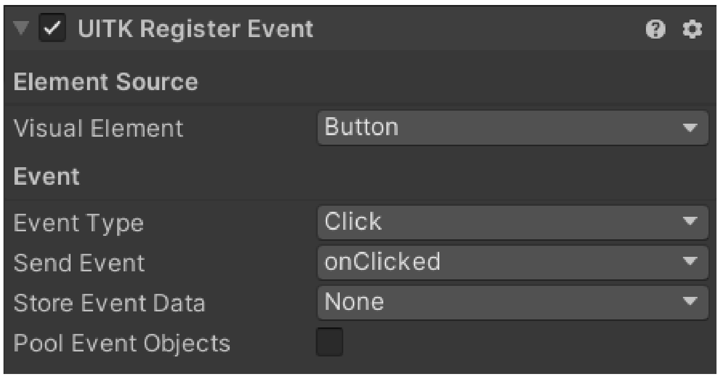
Document

UI Document Source: Use Owner

UITK Query Many Register Event
UITK Register Event
UITK Query One Register Event
UI Toolkit [2]
UITK Query Many Register Event
UITK Query One Register Event
UITK Register Event
UITK Register Events [1]

UITK Register Event

Works like the Query Register Event actions only that this focused on the register event part.


UITK Register Event

Element Source

Visual Element	Button

Event

Event Type	Click
Send Event	onClicked
Store Event Data	None
Pool Event Objects	☐

Visual Element: The visual element on which the event will be registered.

<u>**Event:**</u>

Event Type: A list of visual element events you can register to.

HINT: You may have noticed that „Drap“ events are not in there. Sadly these are EditorOnly in UI Toolkit. Once Unity makes them runtime compatible they will be added.

Send Event: The PlayMaker event that should be triggered.

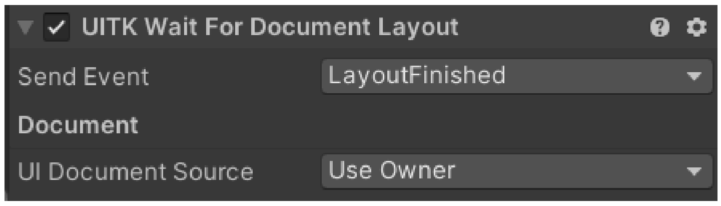
Store Event Data: Variable where the event data from the visual element callback will be stored.

Pool Event Objects: If enabled then event objects are reused the next time the event is triggered. This is done to preserve memory. Enable only if you see a need for it in the profiler. Usually its fine to leave this turned off.

UITK Wait For Document Layout

When first started (i.e. when the UI is shown for the first time) the UI Toolkit Layouting will happen within the first frame.

Only after that values like „width“ or „height“ will have valid values and all styles will be resolved.



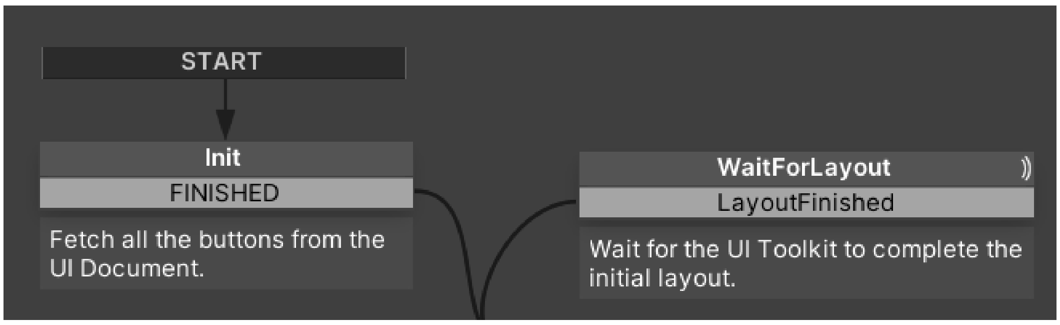
UITK Wait For Document Layout

Send Event: LayoutFinished

Document

UI Document Source: Use Owner

If your logic relies on reading values from the UI at the start then you should wait for this layouting to finishe before reading and modifying style values.



START

Init

FINISHED

Fetch all the buttons from the UI Document.

WaitForLayout

LayoutFinished

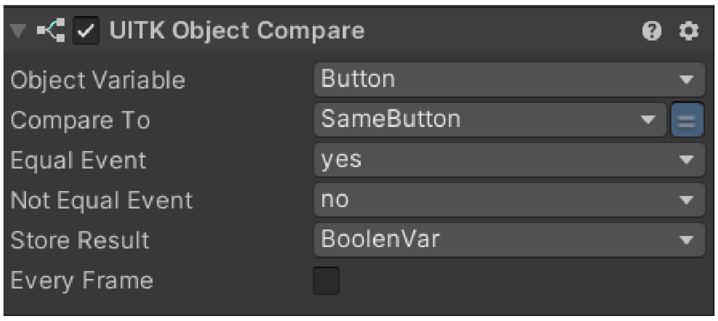
Wait for the UI Toolkit to complete the initial layout.

UITK Object Compare

Do NOT use the regular Object Compare for comparing visual elements.

The reason is that visual elements need a special comparison. The regular „Compare Object" will give you false results. If you are intereste why then read this.

The logic works just like the regular Compare Object action.



UITK Object Compare

Object Variable	Button
Compare To	SameButton
Equal Event	yes
Not Equal Event	no
Store Result	BoolenVar
Every Frame	☐

Object Variable: The first object.

Compare To: The second object.

Equal Event: Event that is called if the two objects are equal.

Not Equal Event: Event that is called if the two objects are not equal.

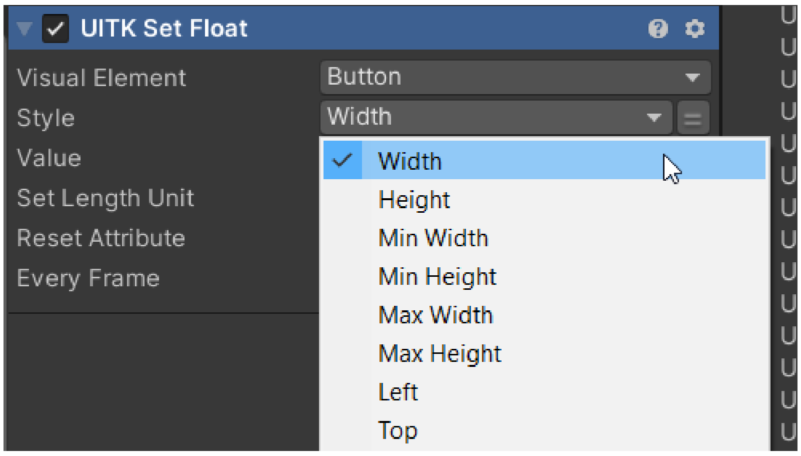
Store Result: Variable to store the boolean result.

Every Frame: Execute every frame (in Update)?

UITK Set/Get Float

This action can control many different style attributes.

It supports these attributes: Width, Height, MinWidth, MinHeight, MaxWidth, MaxHeight, Left, Top, Right, Bottom, Opacity, FlexGrow, FlexShrink, FontSize, TextOutlineWidth, BorderWidth, BorderLeftWidth, BorderTopWidth, BorderRightWidth, BorderBottomWidth, BorderRadius, BorderTopLeftRadius, BorderTopRightRadius, BorderBottomLeftRadius, BorderBottomRightRadius, Margin, MarginLeft, MarginTop, MarginRight, MarginBottom, Padding, PaddingLeft, PaddingTop, PaddingRight, PaddingBottom, Rotation, Scale, ScaleX, ScaleY



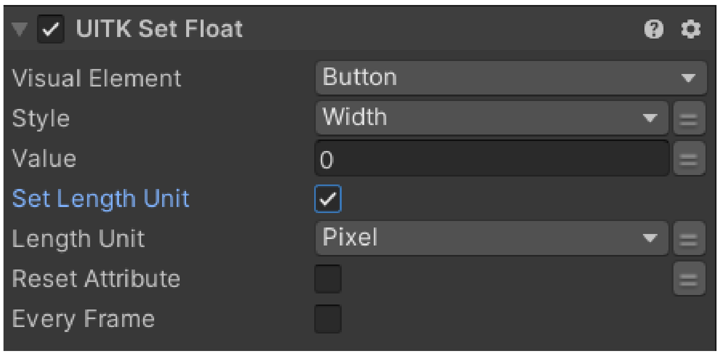
Visual Element: The visual element on which to operate on.

Style: Which attribute to change.

Value: The new value to set (float).

Set Length Unit: You can set the length unit for attributes that do support units.

Length Unit: Only visible if „Set Length Unit“ is enabled.


UITK Set Float

Visual Element	Button
Style	Width
Value	0
Set Length Unit	✓
Length Unit	Pixel
Reset Attribute	
Every Frame	

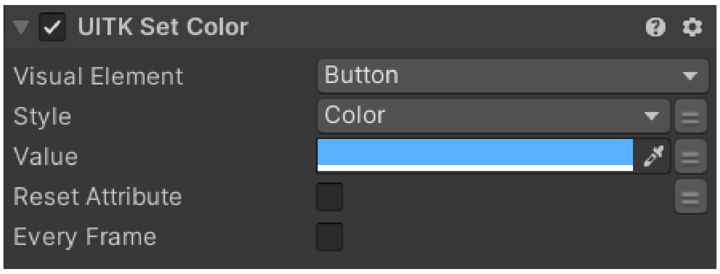
Reset Attribute: Reset the attribute to being „undefined“ (uses StyleKeyword = null).

Every Frame: Execute every frame (in Update)?

UITK Set/Get Color

This action can control many different style attributes.

It supports these attributes: Color, TextColor (alias for „Color“), BackgroundColor, TextOutlineColor, BackgroundImageTint, BorderLeftColor, BorderTopColor, BorderRightColor, BorderBottomColor.



UITK Set Color

Visual Element	Button
Style	Color
Value	🟦
Reset Attribute	☐
Every Frame	☐

Visual Element: The visual element on which to operate on.

Style: Which attribute to change.

Value: The new value to set (color).

Reset Attribute: Reset the attribute to being „undefined“ (uses StyleKeyword = null).

Every Frame: Execute every frame (in Update)?

Frequently Asked Questions

Here are some common issues that have been reported.

If you can, please upgrade to the latest LTS version of Unity and PlayMaker. The newer the versions the less „glitches“ the UI Toolkit has.

Keep in mind, UI Toolkit as a whole it is still a work in progress and not quite ready for prime time. Unity itself still recommends using UGUI instead of UI Toolkit for runtime applications (source).

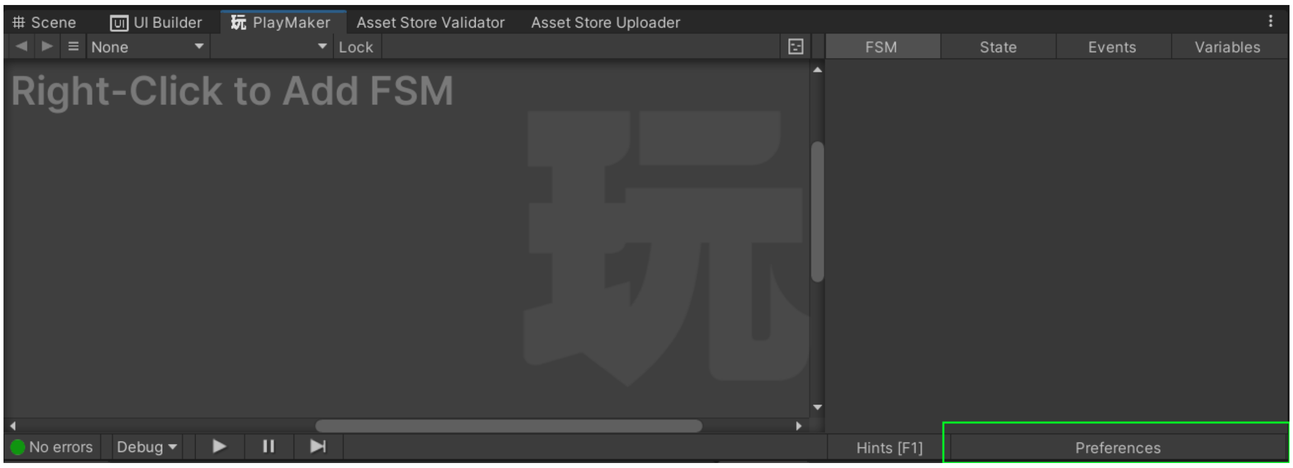
Playmaker keeps adding a „PlayMakerGUI“ object. Is it needed?

No, this object is for UGUI and you do not need it for UI Toolkit.

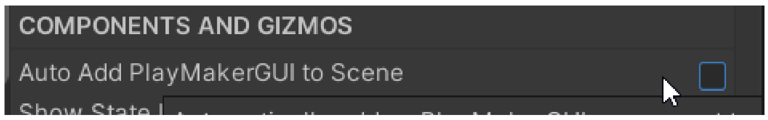
You can disable the auto-add in the preferences.

NOTICE: The preferences button is in a rather odd position (bottom right of the PlayMaker Window). You will not find it in the normal menus.

Here is the „preferences“ button:



In there you can disable the auto-add.



What about Drag and Drop Events?

You may wonder why the drag and drop events have no actions (DragEnterEvent, DragLeaveEvent, DragExitedEvent, DragPerformEvent, DragUpdatedEvent).

The answer is that these are not supported at runtime.
Source: https://forum.unity.com/threads/dragexitedevent-does-not-exist-in-the-namespace-unityengine-uielements.1430083/

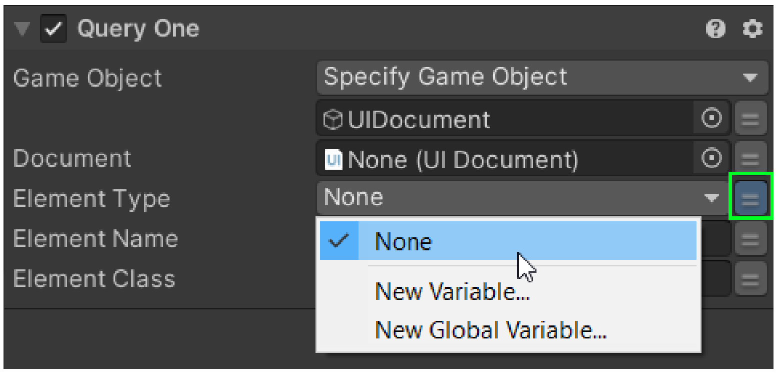
To implement drag and drop at runtime Unity recommends using pointer capturing. Here is the page explaining it in the Unity Manual: https://docs.unity3d.com/Manual/UIE-create-drag-and-drop-ui.html

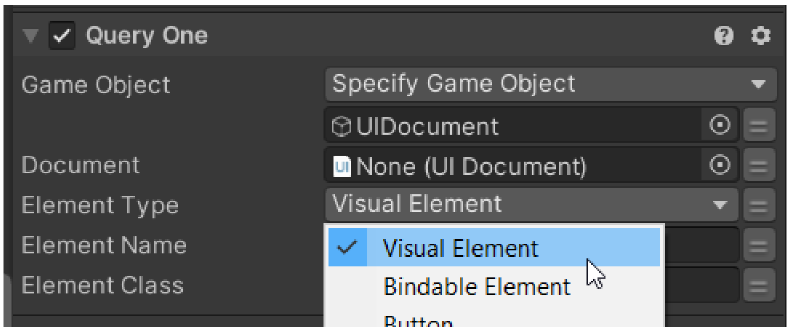
Getting some style values (color, width) returns invalid values at start?

Some values in the UI are only set properly after the first layout cycle has completed. This is a limitation of UI Toolkit. To get proper values you will have to wait one frame or use the „UITK Wait For Document Layout“ action to wait for the layout to finish.

I can’t select the „Element Type“. It always shows „None“?

You may be in „use variable mode“. Click on the „=“ button on the right. That should switch you over to the enum selection mode.


Query One
Game Object: Specify Game Object
UIDocument
Document: None (UI Document)
Element Type: None
Element Name
Element Class


Query One
Game Object: Specify Game Object
UIDocument
Document: None (UI Document)
Element Type: Visual Element
Element Name
Element Class