UI ANIMATOR

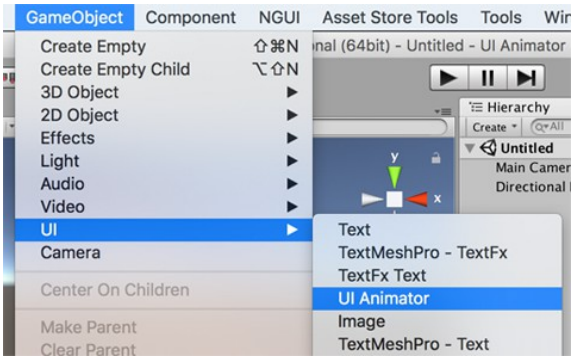
User Guide

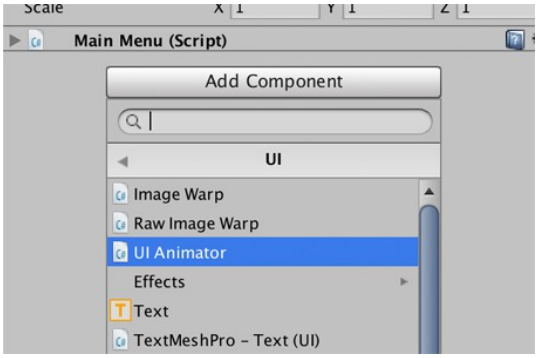
@fenderrio
fenderrio@gmail.com

Table of Contents

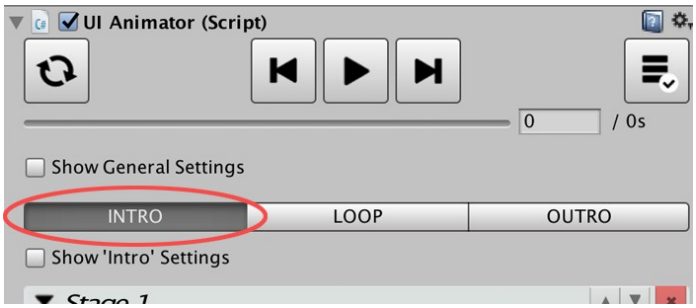
Getting Started

Add a UI Animator component in your scene.

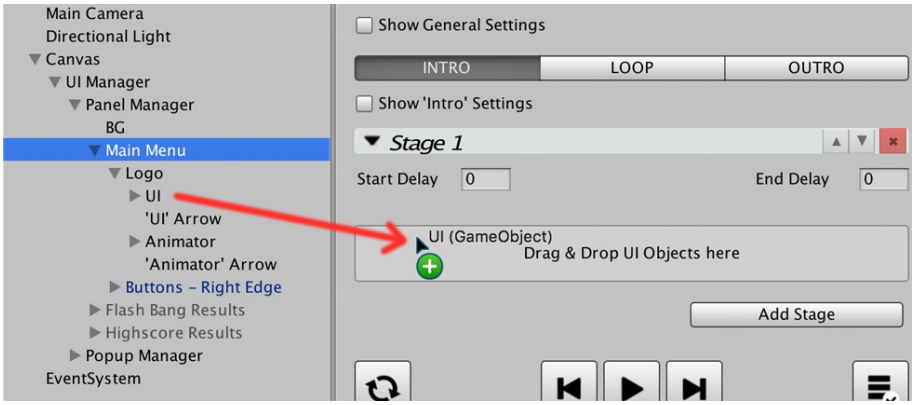




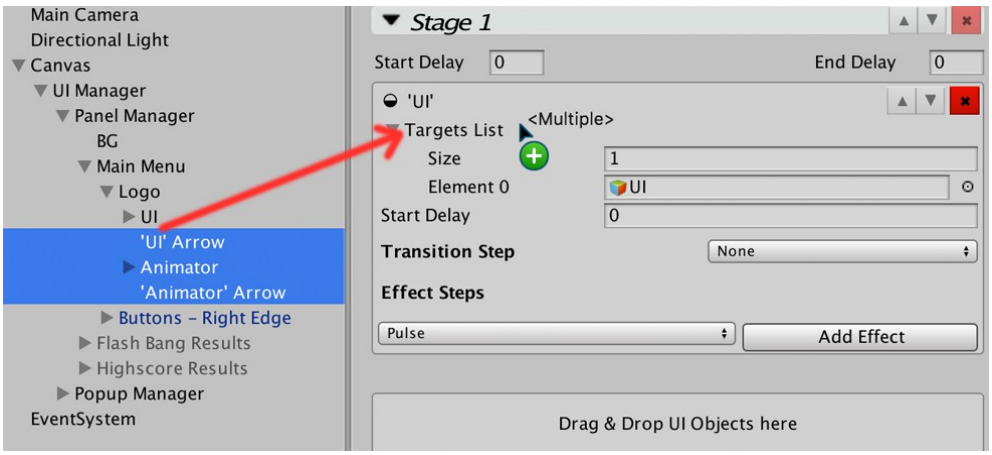
Pick the animation type you want to setup.



Drag & drop UI objects that you want to animate.

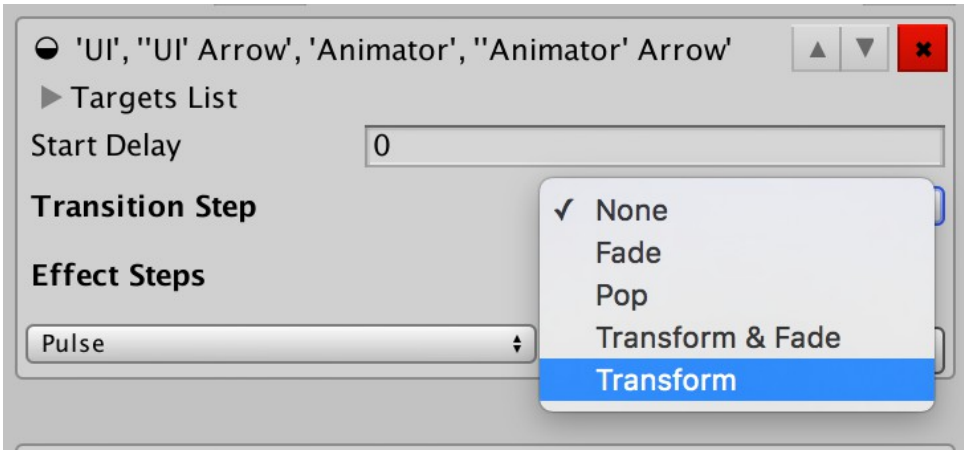


- You can drag more that one object to animate a group.
- You can always add/remove targets later from the targets list.



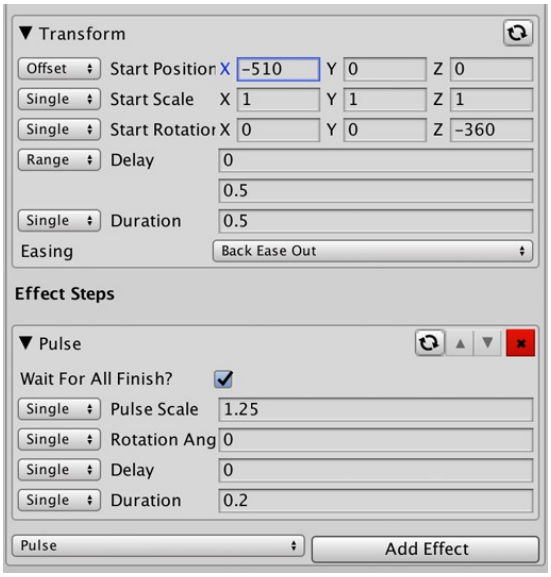
TIP: Remember to make use of the Inspector 'Lock'

Add some animation steps.

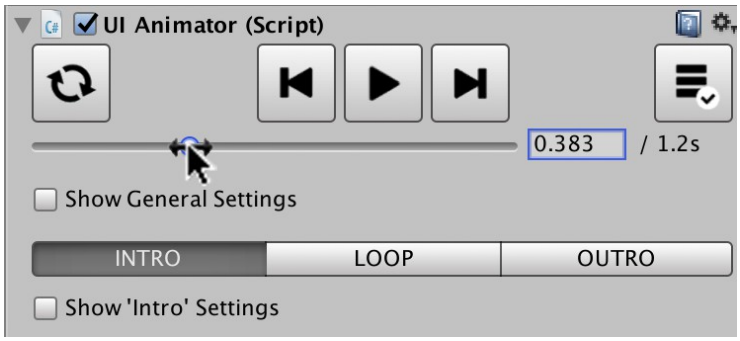


- There are two type of animation step; **Transition** & **Effect.**
  - **Transition Steps** – Meant for transitioning objects in/out
    eg. Sliding a button in from off-screen left.
  - **Effect Steps –** Applies an animation effect, and always returns it to its original state
    eg. a button *pulse* effect.
- Each animation type (**Intro, Loop, Outro)** is made up differently.
  - **Intro –** Zero or one *Transition* Step, followed by zero or more *Effect* steps.
  - **Loop –** Only *Effect* steps.
  - **Outro –** Zero or more *Effect* steps, followed by zero or one *Transition* step.

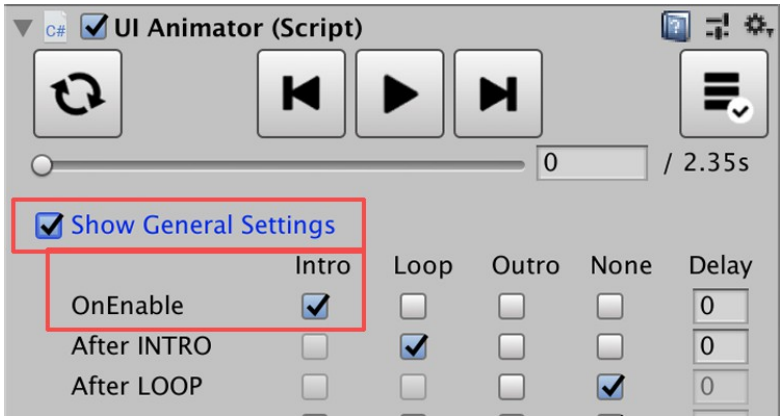
Play around a bit!



Preview your animation in the editor at any point, using the **playback controls** and the **animation slider**.



Setup to play at runtime.



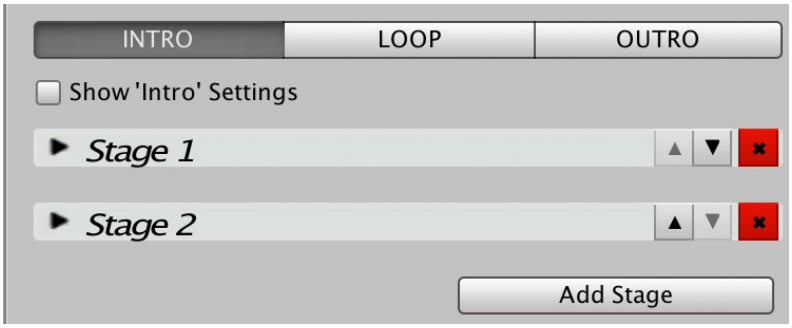
By default your new animations will be set to play the ***Intro*** animation when **OnEnable** is called, but this can be changed via the **General Settings**.

```
using UnityEngine;
using UIAnimatorCore;

public class UiAnimatorTest : MonoBehaviour {

    public UIAnimator m_uiAnimator;

    void Start ()
    {
        m_uiAnimator.PlayAnimation (AnimSetupType.Intro);
    }
}
```

- You could also trigger an animation via a scripting call to **PlayAnimation()**, like in the above example.
- See the ***Scripting API*** for full details on the scripting methods available.

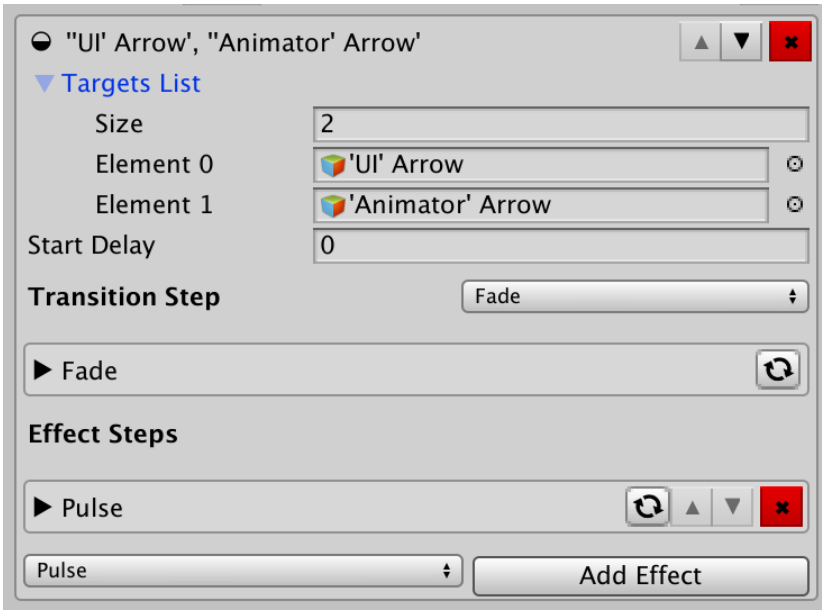
Other Fundamentals

What Are 'Stages'?



- A *Stage* is just a *collection* of *Animation Instances*.
- Each *Stage* waits for all of the contained *Animation Instances* to finish, before moving onto the next stage.
- You'll mostly only use one Stage.

What are 'Animation Instances'?



- An *Animation Instance* is a configuration of *Animation Steps* (Transition or Effect) which are to be applied to one or more *Target* UI gameObjects.
- Whether you are setting up an *Intro*, *Loop* or *Outro* animation, the setup of the *Animation Instance* will vary:

  - Intro – Zero or one *Transition* Step, followed by zero or more *Effect* steps.

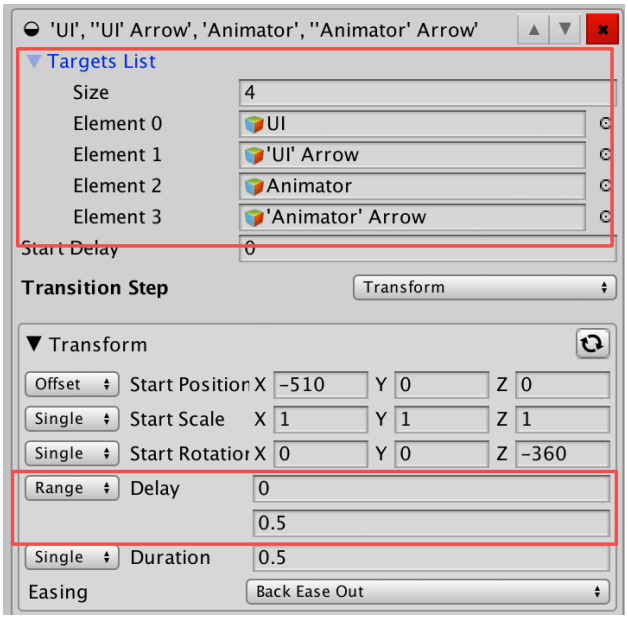
  - Loop – Only *Effect* steps.

  - Outro – Zero or more *Effect* steps, followed by zero or one *Transition* step.

How to change the UI layout after setting up an Animation?

- Animation steps ***cache the state*** of the UI when you first add them, so that they know what to animate to/from.
- If you want to change the default UI layout that gets animated, you'll need to:
  - Make sure the animation is in its Default (Non-animated) state by ***Resetting to Default State***. (*See above image*)
  - Make your changes to the UI layout.
  - Press the ***Set As Default State*** button. (*See above*). This will update the animation steps cached states to the current UI layout.

How to Animate a Group of Objects together?

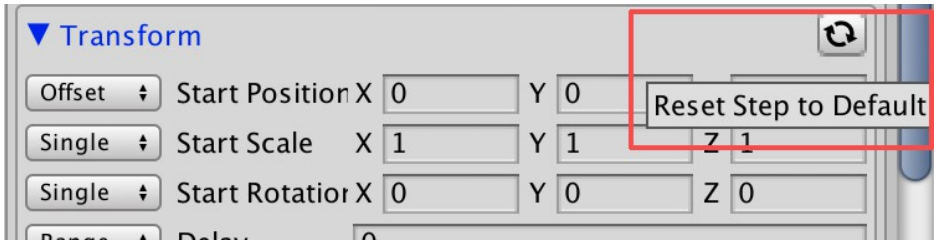


- You can assign multiple target objects to each animation setup.
- This helps to save time when you want a group of objects to animate in a similar way. *eg. a row of buttons popping into existance.*
- As soon as an animation setup has more than one target, the interface changes to show options for each variable to either be:

  - ***Single*** - A single shared value for each target.

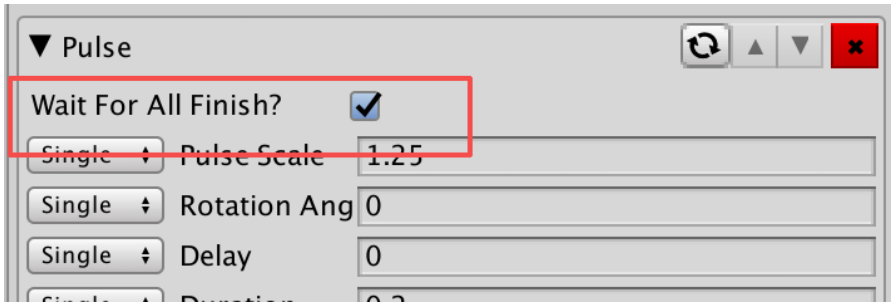
  - ***Range*** - A range of values, from/to, applied to each target in order.

How to Reset an Animation Step to Default Values.



- You can reset the values of any Animation Step by pressing the ***'Reset Step to Default'*** button.

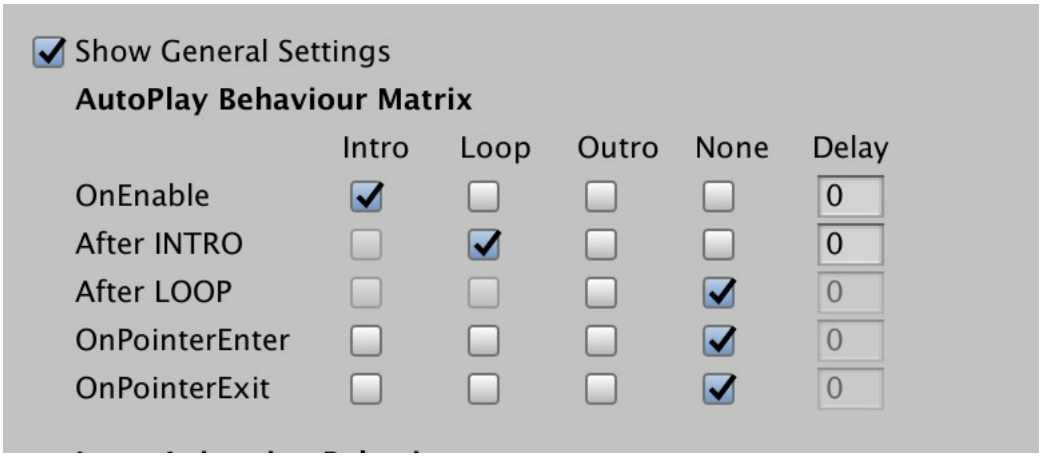
What Does 'Wait For All Finish?' do?



- If you're animating on a group of targets, then any Animation Step after the first will have the option '***Wait For All Finish?***'
- This option denotes whether the animation step should *wait for all targets of the previous steps* to be *finished*, or just start whenever any of the targets has finished.
- This only has an effect if you're setting a ***range*** of different ***delays*** or ***durations***.

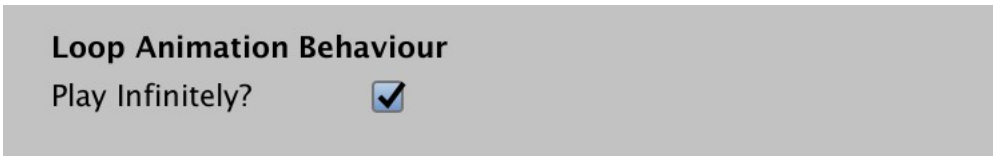
Further Topics

The Animation 'AutoPlay Behaviour Matrix'



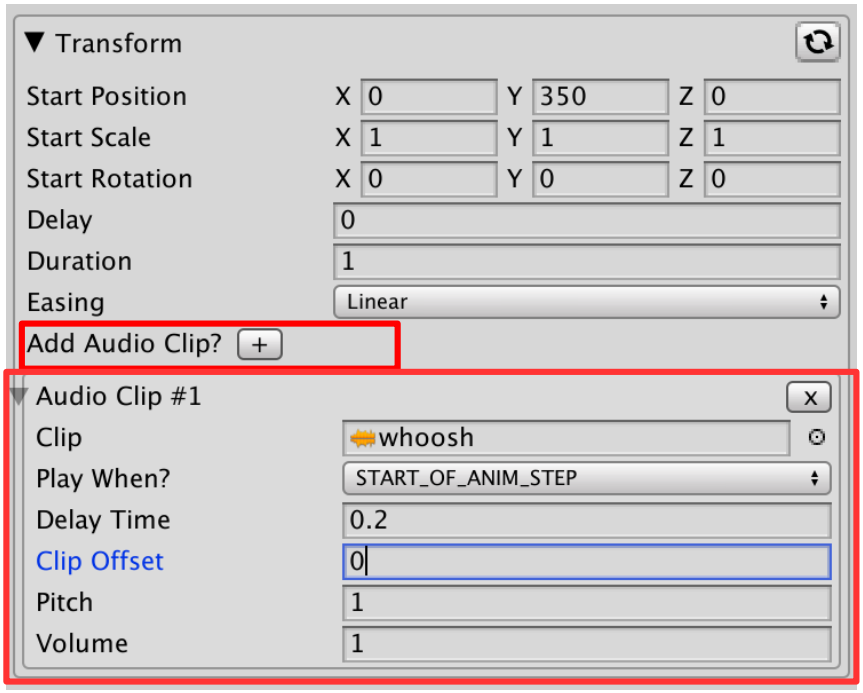
- This is a time saving feature to reduce the need for coding calls to trigger animations.
- Using the AutoPlay Behaviour Matrix, you can configure your UI Animation to automatically play a particular animation state after certain events:
  - **OnEnable** – Set the animation to be played when the *UI Animator* gameObject is first active in the scene.
    - *By default this is set to play the 'Intro' animation onEnable.*
    - *Setting this option to **'None'** will present an additional option to set which Animation State pose (Intro, Loop, Outro) the UI Animator instance should be set OnEnable.*
  - **After INTRO** - Set the animation to be played when the *'Intro'* animation has finished playing.
  - **After LOOP –** Set the animation to be played when the *'Loop'* animation has finished playing.
  - **OnPointerEnter –** Set the animation to be played when the *UI Animator* content registers an *OnPointerEnter* event.
    - ***TIP:** It's often useful to add an additional **transparent Image** object to be used as a consistent onPointer trigger area, which isn't animated.*
  - **OnPointerExit** – Set the animation to be played when the UI Animator content registers an *OnPointerExit* event.

Loop Animation Behaviour setting



- A setting found under the ***'Show General Settings'*** option.
- When enabled, the 'Loop' animation will continue to play infinitely until manually told to stop or change. *This is the default setting*.
- When disabled, you can specify how many times the 'Loop' animation should play before it automatically stops playing.
    - **Note**: If you want the animation to play the *'Loop'* for a fixed number of times and then automatically lead into the *'Outro'* animation, use this setting in combination with the ***'AutoPlay Behaviour Matrix'*****.**

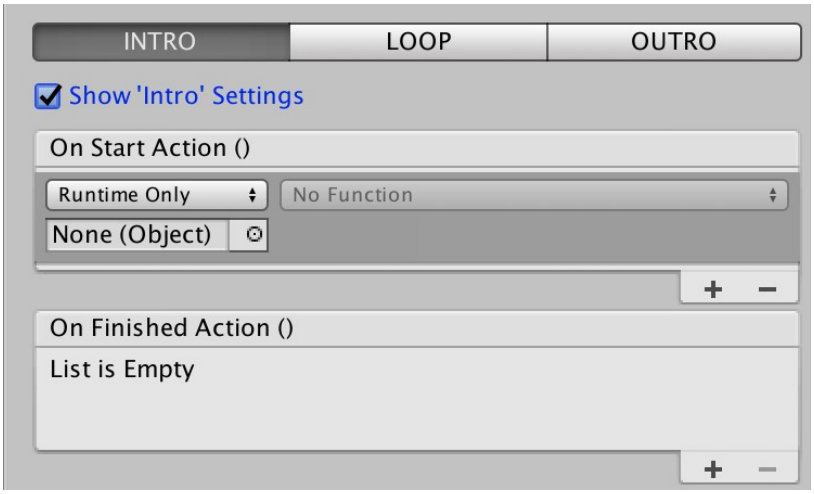
Adding AudioClips to your Animation Steps



- Any Animation Step (Transition or Effect), can have AudioClips assigned to it to triggered at certain points.
- Click the Add ('+') button next to the ***'Add Audio Clip?'*** option at the bottom of the animation step settings.
- You can assign as many AudioClips as you like.
- Simply assign an AudioClip currently imported into your Unity project, and then setup when and how it should play:
    - **Play When? -** Set whether the clip should play at the start or the end of the animation step.
    - **Delay Time**
        - If the clip is set to play from the *Start*, this will be the time delay in seconds from when the animation step starts playing.
        - If the clip is set to play from the *End*, this will be the time in seconds from the end of the animation step, that the clip should start playing.
            - eg. '0' would be at the end of the animation step. '0.5' would be 0.5 seconds before the end of the animation step.
    - **Clip Offset –** How after into the clip (in seconds) should the clip start playing.
    - **Pitch** – The pitch that the clip should be played at.
        - '1' is the default value.
    - **Volume –** The volume that the clip should be played at.

- UIAnimator will automatically create a ***pool of AudioSource gameObjects*** as children of the UIAnimator gameObject.
    - **Note:** In order to avoid potential runtime performance spikes, it's recommended that you allow UIAnimator to create these AudioSource child objects **in edit mode**, and save it in your scene. This way UIAnimator can re-use the already created AudioSources at runtime, saving the creation of new GameObjects and components at runtime, which can be computationally expensive.

Assign an onStart / onFinish callback?

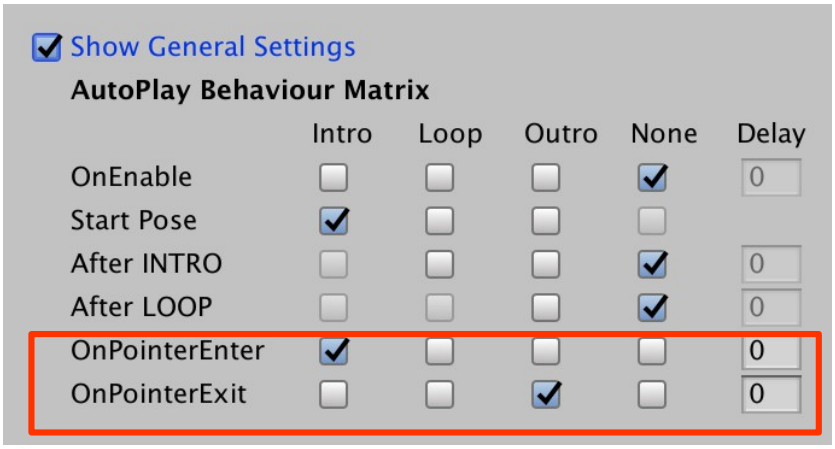


- Each animation type (*Intro, Loop, Outro*) has optional ***OnStart*** & ***OnFinish*** callback events which can be configured in the inspector.

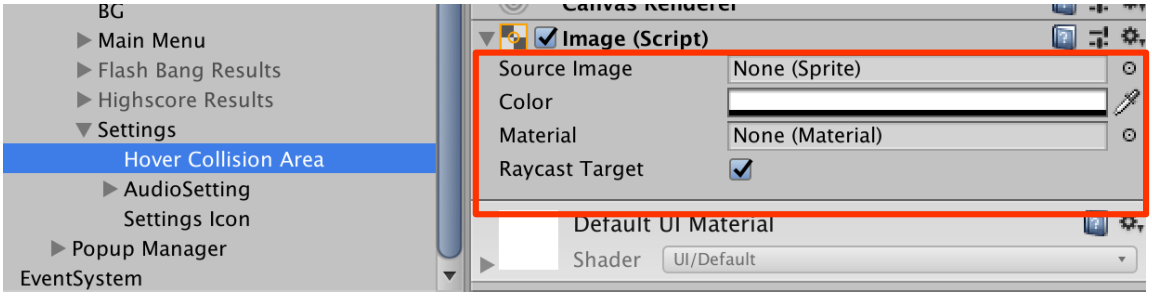
```
1 using UnityEngine;
2 using UIAnimatorCore;
3
4 public class UiAnimatorTest : MonoBehaviour {
5
6     public UIAnimator m_uiAnimator;
7
8     void Start ()
9     {
10        m_uiAnimator.PlayAnimation (AnimSetupType.Intro, a_onFinish: () => {
11
12            Debug.Log("UI Animation finished");
13
14        });
15    }
16 }
```

- You can also provide a callback method to the ***a_onFinish*** parameter of *PlayAnimation()*.

Trigger Animations OnPointerEnter / Exit

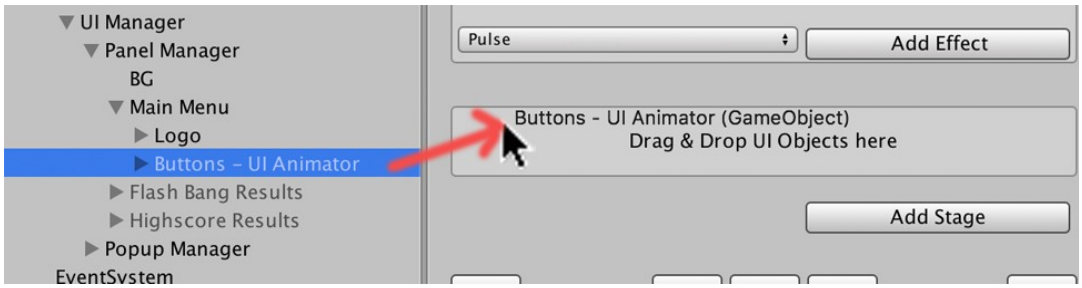


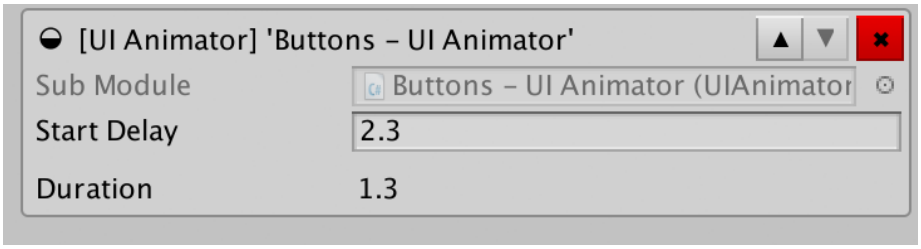
- Sometimes you may want to trigger a UI Animation when the user hovers the pointer over a particular area of your UI.
  - *eg. Hovering over a settings icon in the top left of the screen could reveal additional UI elements, and then hide them away when the pointer leaves the area.* ***See the UI Animator Demo Scene for an example of this!***
- This functionality can be achieved by setting up the *OnPointerEnter / Exit* **autoplay behaviours** under the *General Settings* for your UI Animator instance.
  - Set which animation you'd like to be triggered on each event, and specify a delay if required.
  - It's as easy as that!



- ***TIP:*** *Create a non-animated transparent Image, to use as the pointer collision area.*
  - *Make sure to set this as a* ***'Raycast Target'*** *(set to* ***true****), so that it receives the pointer enter/exit events.*
  - *Set* ***all other non-interactive UI elements*** *to* ***not*** *be 'Raycast Target's (set to* ***false****), so that they don't accidentally trigger enter/exit states when you don't want them to!*
  - *See the 'Settings' UI elements in the* ***Demo Scene*** *for an example of this.*

How Can I Reuse UI Animations in a Modular Way?





- You can drag other ***child*** UI Animator instances onto a parent UI Animator, and then control it as part of its animation sequence.
- This way you can have a ***shared UI Animation prefab*** which you assign to multiple parent UI Animation instances, and save having to setup and maintain the animation in multiple places.

Scripting With UI Animator

The Basics

- Include the ***UIAnimatorCore*** namespace.
- Keep a reference of a *UIAnimator* component from your scene.
- Call to play an animation, and optionally listen for the onFinish callback event.

```
1 using UnityEngine;
2 using UIAnimatorCore;
3
4 public class UiAnimatorTest : MonoBehaviour {
5
6     public UIAnimator m_uiAnimator;
7
8     void Start ()
9     {
10        m_uiAnimator.PlayAnimation (AnimSetupType.Intro, OnIntroAnimationFinished);
11    }
12
13    void OnIntroAnimationFinished()
14    {
15        Debug.Log ("Intro finished");
16    }
17 }
```

Scripting API – UIAnimator class

Public Properties

```
AnimSetupType CurrentAnimType { get; }
bool IsPlaying { get; }
bool Paused { get; set; }
PlayTimeMode TimeMode { get; set; }
float Timer { get; }
```

Public Static Methods

```
void SetUIAudioState( bool a_audioIsPlaying );
```

Public Methods

```
void      ForceStopAllAudioSources ();
float     GetAnimationDuration ();
float     GetAnimationDuration ( AnimSetupType a_animType );
void      PlayAnimation ( AnimSetupType a_animType );
void      PlayAnimation (     AnimSetupType a_animType,
                              float a_delay,
                              System.Action a_onFinishCallback );
void      ResetToEnd ();
void      ResetToEnd ( AnimSetupType a_animType );
```

```
void      ResetToDefault ();

void      ResetToStart ();

void      ResetToStart ( AnimSetupType a_animType );

void      SetAnimationTimer ( float a_timerValue );

void      SetAnimationTimer ( AnimSetupType a_animType,
                              float a_timerValue );

void      SetAnimType ( AnimSetupType a_animType );

void      SetPlayOnEnable(    bool a_playOnEnable,
                              AnimSetupType a_animToPlay,
                              float a_delay = 0);

void      SetPlayAfterIntro(  bool a_playAfterIntro,
                              AnimSetupType a_animToPlay,
                              float a_delay = 0);

void      SetPlayAfterLoop(   bool a_playAfterLoop,
                              AnimSetupType a_animToPlay,
                              float a_delay = 0);

void      SetPlayOnPointerEnter(   bool a_playOnPointerEnter,
                                   AnimSetupType a_animToPlay,
                                   float a_delay = 0);

void      SetPlayOnPointerExit(    bool a_playOnPointerExit,
                                   AnimSetupType a_animToPlay,
                                   float a_delay = 0);

bool      UpdateState ( float a_deltaTime );
```

Appendix

AnimSetupType

```
public enum AnimSetupType
{
    Intro,
    Loop,
    Outro
}
```

PlayTimeMode

```
public enum PlayTimeMode
{
    GAME_TIME,
    REAL_TIME
}
```

Support

Support email: **fenderrio@gmail.com**

Thank you for buying UI Animator! :)

Changelog

v1.0.0 – 02/04/2018

- First release

v1.1.0 – 08/09/2018

- Added option to trigger AudioClips during UI Animations.
- Tidied up the 'AutoPlay' functionality; now an easy-to-use Behaviour Matrix (in General Settings)
- Added support for triggering animations OnPointerEnter / Exit
- Tidied up and improved the User Guide; now a proper PDF with hyperlinked contents table.
- Bug fixes.