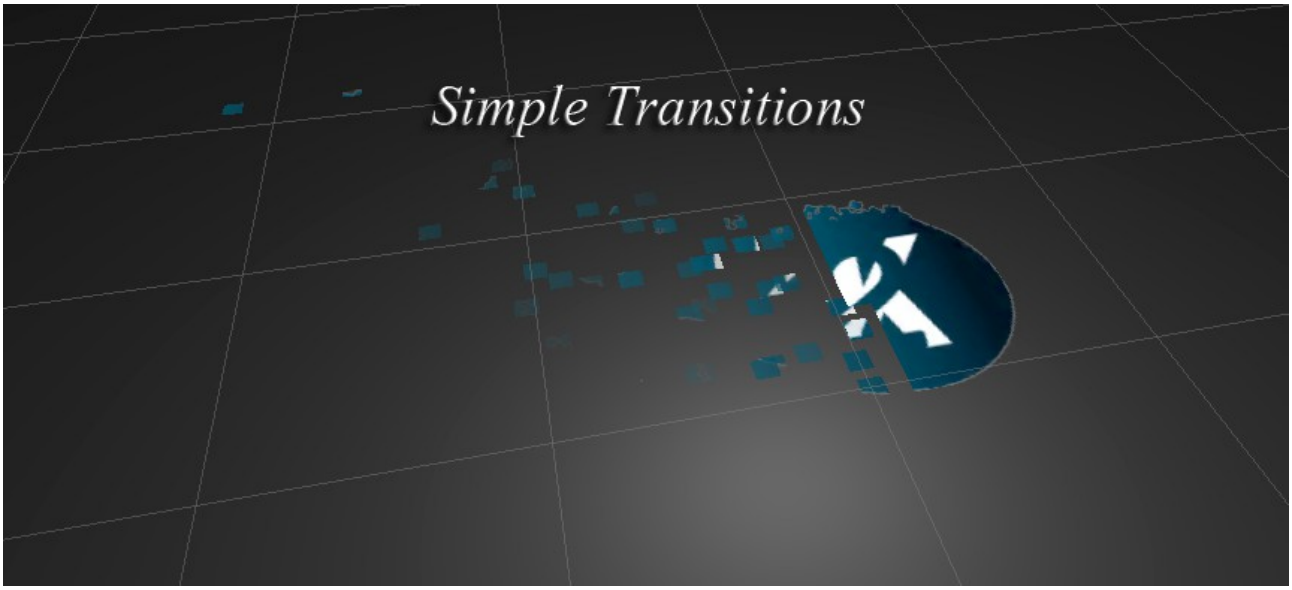
Simple Transitions

Simple Transitions 3.0

To use this framework simply add the TransitionalObject script to any game object, then add which ever transitions you need using the 'Add New' drop down.

Transition Types

This framework supports many types of animation:

Alpha
Colour
Movement
Rotations
Scaling
Manual

Alpha Transition

This is a very common transition type that will turn the given object and any of its children either invisible or visible. It will work automatically with uGUI Images and Sprites however if you work with nGUI you will need to uncomment any instances of:

#define UsingNGUI

Alpha transitions work by searching for instance of objects to effect before the game is run. These can be found and edited under the 'Affected Renderers' tab at the bottom of any Transitional Object with a Colour or Alpha component.

Note: If you notice any flashing whilst using an alpha transition try turning on the value 'Start Faded' under the data tab. This can sometimes happen if you object is deactivated.

Important:

Alpha and Colour transitions will work on any Renderer component it finds in the hierarchy however its important to note that any materials on that Renderer must have an exposed Color field. As an example the old Unlit/Transparent shader will not work but the VertixLit/Transparent shader will.

Colour Transition

Colour transitions are almost identical to the alpha transition except it can also effect the colour tint (RGB values) of an object rather than just the alpha component.

Moving Transition

Moving transitions will move the object from its given start point (found under the 'Data' tab) to its end point and can do so in a few ways:

Absolute:

This method will simply set the Transform.position of the object and completely ignore its parents position over the course of the animation.

Local:

The local method respects the parents animation and is probably used most often. It works by editing the Transform.localPosition

Difference:

A difference transition is a bit more specialised and probably least used. Whenever called it will simply add the difference between the start and end

point to the current position.

Note that there are also options to deviate either the start or end point if you feel you need to randomise your animation or to 'Read Current'. This checkbox means that whenever TriggerTransition is called the start point is automatically read to be the current position of the object.

Rotating Transition

As you would expect by now this transition rotates the given object. The only thing to keep in mind is that there is no way to know which direction you mean to rotate. E.G with a start value of 0 and end of 180 you might mean to go down from 360 to 180.

Sometimes setting the rotation to say 359 will fix this situations however you can also check the 'Reverse Rotation' checkbox and it should also fix these situations.

Scaling Transition

Scaling simply edits the transform.localScale property which will in turn scale all children of the object.

User Interface

Each component in the UI can be toggled to show and hide data you want to see. Most of the elements should be pretty self explanatory so I will try and cover this as lightly as possible.

Data Tab
This tab contains the data specific for each component, for Colour its the Colour data and Moving its the start and end points.

Transition In
This section controls what happens as the animation is appearing and has 3 fields.

The delay field is used to delay the animation rather than running the instant it is called. It can also be set to be a 'One Shot' delay. This means the first time it is triggered the delay is run but any future calls to trigger transition will be instant.

The run time is the time it takes to completely fade in.

The final field is the graph that controls how this transition is smoothed (clicking it will bring up the graph editor window). This is very important to give more natural animations to make objects appear to accelerate and decelerate.

Fade Out
This tab is identical to the 'Transition In' tab however it also has the option to match its run time value. This is useful whilst first testing your animations however it can feel nicer if your fade-outs are a little quicker than your transition in times.

Timings
Initially this appears to only have 2 options however if you un-check the stay forever tab 2 more will be revealed.

'Trigger Instantly' means that the animation will run as soon as it is awake.

'Stay Forever' is automatically checked as a default. It means that once you trigger a transition, say to show a menu, then it will stay open until the 'TriggerFadeout' method is called. If unchecked then you have more options:

'Display Time' will then control how long to stay on screen after a transition is called. This can be useful for message boxes that automatically disappear after a few seconds.

'Is Looping' will of course cause the transition to loop continuously. Loops can be stopped at any time by calling the 'Stop' method.

Messages
Messages are used to send events at any given point during an animation. As an example when your menu has finished transitioning in you can use events to signal the menu state has changed.

To send an event first click the '+' button. You should now see a default Unity events tab and a drop down box above it to control when to send the message (covered further below). To send an event click the smaller '+' button then you should see a box to drag in the target of your event. Once you have a target the second drop down should be enabled allowing you to select the Class and method to call.

There are 7 different states that a message can be sent:

Delay: This is called when 'TriggerTransition' has been called and the object is in its delay state before actually animating.

Transition In: This is for when the object has finished its delay and has actually started animating.

Waiting: This is the default state for any events. This is the state for when a transition has finished. E.G when your menu is now open.

Fade Out Delay:
Same as the other delay state however this only triggers whilst the object is fading out.

Fading Out:
Called once the delay has finished and the object is starting to fade out.

Finished:
For once the Fade Out has finished, so when your menu screen has been closed.

Any State:
Called when the transition has changed to any of the other states.

Scripting

Interacting with this framework should be pretty painless. When your opening your menu screens call 'TriggerTransition', when your closing them call 'TriggerFadeout'.

If you need to alter any values for a specific component they can be accessed via the 'transitions' array of a Transitional Object. The arrays order is the same as you see in the inspector.

Transition Groups

Transition groups are just collections of Transitional Objects that all trigger simultaneously. To trigger all the transitions call:

TriggerGroupTransition
TriggerGroupFadeOut

Groups also allow you to edit all values of any transitions in the group. Note that these changes are applied at run time to make development easier.

The values can be edited with given maths operators that should be self explanatory however there is also an option to apply a stagger. Stagger means that your value will be multiplied according to the transitions position within the group.

As an example you might have 5 buttons drop onto your UI in overly epic style, or you might have your score breakdown screen animate to show each piece of data one at a time. Rather than manually adding a delay to each object a Stagger basically lets you define the time between each object so they appear one at a time.

Additional Settings

To support editing many objects at once there are some menu functions that should help, these can be found under Edit/Transitions (Soon to be moved to a Tools menu) and are explained in more detail in the videos on the asset store.

There is also a small preferences section under Edit/Preferences.../K2 Games. Its currently very empty but likely to expand in future updates.

Support

If you have any issues just drop us an email at:

Suport@K2Games.co.uk

Or find us on Facebook at:

www.facebook.com/K2Games