TWEEN

View online: https://www.construct.net/en/make-games/manuals/construct-3/behavior-reference/tween

The **Tween behavior** animates the values of properties over time. For example you can "tween" an object's position to (100, 100), which will move it from its current position to the target position over time. Different *ease functions* can be used to alter the rate at which the value changes over time, for example using a *Linear* function for a constant-rate motion, *In Out Sinusoidal* for a sine-curve based movement which speeds up then slows down, and many more. Custom ease curves can also be designed with the Ease editor.

Click here to open an example of the Tween behavior.

The term "tween" comes from the term "Inbetweening", referring to generating intermediate frames in between two states.

> *Tweening to a position moves objects over a fixed time. If you need to move objects to a position using a maximum speed with acceleration and deceleration, use the Move To behavior instead.*

Scripting

When using JavaScript or TypeScript coding, the features of this behavior can be accessed via the ITweenBehaviorInstance script interface.

Tween types

Construct's Tween behavior has three types of tween:

1. One-property tween, affecting only a single value, e.g. the X co-ordinate or the angle
2. Two-property tween, affecting a pair of values, e.g. the position (X and Y co-ordinates), size (width and height) or the scale (a percentage of real size)
3. Value tween, which just tweens a number rather than affecting a property. This can be used to apply the tween to anything else in events, such as behavior properties or effect parameters. Value tweens must also specify the start value, since it cannot be taken from a property.

These types are used by the three tween actions: *Tween (one property)*, *Tween (two properties)* and *Tween (value)*.

Tagging

The Tween behavior can manage multiple tweens simultaneously. To help identify them separately, each tween can be given a *tag*, which is just a string of any name you like to identify the tween. The tag is optional and can be left empty if you don't need to modify or identify the tween at any point later on.

Tweens can also be given multiple tags, separated by spaces. This can be useful to group tweens together under a common tag while also providing a unique tag to target them individually.

Actions, conditions and expressions which expect tags as an argument will match any tweens which include all of the provided tags.

Tagging examples

Given these tweens with these tags:

Name	Tags
Tween1	"common size width"
Tween2	"common size height"
Tween3	"common color"

This will be the expected matches:

Tag	Matches
"common"	**All Tweens**
"common size"	**Tween1** and **Tween2**
"common size width"	**Tween1**
"color"	**Tween3**
"width"	**Tween1**
"size"	**Tween1** and **Tween2**
"common size width height"	**No Tweens**

Also note:

- Actions using tags affect all matching tweens.
- In the case of conditions, all the matching tweens are evaluated and if any is true, then the condition will be fulfilled.
- In the case of expressions, the first matching tween will be the one used to get the value from.
- Tags are case insensitive. For example "mytag", "MyTag" and "MYTAG" are all the same as far as tweens are concerned.

Tween properties

Enabled

Whether the behavior is initially enabled or disabled. If disabled, it can be enabled at runtime using the *Set enabled* action.

Tween conditions

Is any playing

True if any tween is currently playing.

Is playing

Test if a tween matching all the given tags is currently playing.

Is any paused

True if any tween is currently paused.

Is paused

Test if a tween matching all the given tags is currently paused.

Is any ping-pong

True if any tween is in the provided ping-pong *State*

Is ping-pong

True if a tween matching all the given tags is in the provided ping-pong *State*

On any finished

Triggered when any tween finishes. Use the *Tags* expression to get the tag string for the tween that finished.

On finished

Triggered when a tween matching all the given tags finishes.

On any released

Triggered when any tween is released. Use the *Tags* expression to get the tag string for the tween that was released.

On released

Triggered when a tween matching all the given tags is released.

On any looped

Triggered when any tween loops. Use the *Tags* expression to get the tag string for the tween that looped.

On looped

Triggered when a tween matching all the given tags loops.

On any ping-pong

Triggered when any tween ping-pongs. Use the *State* parameter to specify at which point of the ping-pong the trigger should take place. Use the *Tags* expression to get the tag string for the tween that ping-ponged.

On ping-pong

Triggered when a tween matching all the given tags ping-pongs. Use the *State* parameter to specify at which point of the ping-pong the trigger should take place. Use the *Tags* expression to get the tag string for the tween that ping-ponged.

Tween actions - one property

Set end value

Change the end value for an existing one-property tween matching all the given tags.

Tween (one property)

Start a tween for a single property. *Tags* are optional space-separated tags to identify the tween, and can be left blank if not used. *Property* chooses which property of the object to modify. The start value of the tween uses the current value of the property. *End value* specifies the value to tween to. *Time is the duration of the tween in seconds*. *Ease* specifies an ease function affecting the rate of change over time. *Destroy on complete* can be set to *Yes* to automatically destroy the instance when the tween finishes, useful for fade-out effects. Like timelines, a one property tween can be set to *Loop* and/or *Ping Pong* and given a *Repeat count*.

Tween actions - two properties

Set end values

Change the end value for an existing two-property tween matching all the given tags.

Tween (two properties)

Start a tween for two properties. *Tags* are optional space-separated tags to identify the tween, and can be left blank if not used. *Property* chooses which property pair of the object to modify. The start value of the tween uses the current value of the properties. *End X* and *End Y* specify the end value for each of the two properties. *Time is the duration of the tween in seconds*. *Ease* specifies an ease function affecting the rate of change over time. *Destroy on complete* can be set to *Yes* to automatically destroy the instance when the tween finishes, useful for fade-out effects. Like timelines, a two property tween can be set to *Loop* and/or *Ping Pong* and given a *Repeat count*.

Tween actions - value

Set end value

Change the end value for an existing value tween matching all the given tags.

Set start value

Change the start value for an existing value tween matching all the given tags.

Tween (value)

Start a tween for a number, independent of any properties. *Tags* are optional space-separated tags to identify the tween, and can be left blank if not used. *Start value* and *End value* specify the start and end value to tween through. *Time is the duration of the tween in seconds*. *Ease* specifies an ease function affecting the rate of change over time. *Destroy on complete* can be set to *Yes* to automatically destroy the instance when the tween finishes, useful for fade-out effects. Use the *Value* expression to retrieve the current value of the tween over time, such as to apply it to a different object, behavior or effect. Like timelines, a value tween can be set to *Loop* and/or *Ping Pong* and given a *Repeat count*.

Tween actions - playback

Pause

Resume

Pause and resume an existing tween matching all the given tags. Pausing a tween will stop it at its current progress, and resuming will continue from where it was paused.

Pause all

Resume all

Pause and resume all current tweens.

Stop

Stop a tween matching all the given tags. Stopping a tween permanently ends a tween - it cannot be resumed afterwards.

Stop all

Stop all tweens. This permanently ends all tweens so no more tweens can be referenced until a new tween is started.

Tween actions - setters

Set destroy on complete

Set all destroy on complete

Set ease

Set all eases

Set the *Destroy on complete* and *Ease* parameters specified in the *Tween* action for existing tweens matching all the given tags. The *all* variants modify all tweens regardless of tags.

Set playback rate

Set all playback rates

Set the playback rate of existing tweens matching all the given tags, or all tweens regardless of tags. A playback rate of 1 is normal speed, 0.5 half as fast, 2 twice as fast, and so on.

Set time

Set all times

Set the playback time in seconds of existing tweens matching all the given tags, or all tweens regardless of tags. For example setting a time of 1 will skip the tween to playing as if it was 1 second after the tween was started.

Set enabled

Enable or disable the entire behavior. If disabled, the behavior will not affect any properties or advance any tweens.

Tween expressions

Note expressions can only return a single value. When specifying tags, expressions return the value for the first tween with all the given tags.

Progress(tag)

Return the progress of a tween by its tags in the range 0-1.

Tags

This expression can be used in the various triggers to get the tag string of the associated tween.

Time(tag)

Return the playback time of a tween by its tags, in seconds since the tween was started.

PlaybackRate(tag)

Return the playback rate of a tween by its tags.

Value(tag)

Return the current value of a value tween by its tags.