

FOLLOW BEHAVIOR SCRIPT INTERFACE

View online: <https://www.construct.net/en/make-games/manuals/construct-3/scripting/scripting-reference/behavior-interfaces/follow>

The `IFollowBehaviorInstance` interface derives from `IBehaviorInstance` to add APIs specific to the Follow behavior.

Built-in property strings

The Follow behavior uses the following strings to refer to built-in properties in some APIs: `"x"`, `"y"`, `"z-elevation"`, `"width"`, `"height"`, `"angle"`, `"opacity"`, `"visibility"`, `"destroyed"`.

Interpolation strings

The following strings are used to refer to interpolation modes: `"step"`, `"linear"`, `"angular"`.

Follow behavior APIs

startFollowing(inst, fromCurrentPosition = false)

Begin following the specified `IInstance`. This starts recording the changes over time of the specified object, and after the delay period has passed, it will then start following the changes for the enabled properties. Until the delay has passed, `hasFollowData` will be false as there is not yet any data to follow. Alternatively the `fromCurrentPosition` parameter can be set to `true`, which allows immediate following. When enabled this creates an initial history entry based on the following object's current state in the past at the delay time. Therefore `hasFollowData` is immediately true and the object is able to immediately start updating. This has the effect of interpolating from the following object's starting position to the followed object's starting position over the delay time.

stopFollowing()

Stops recording the history of a followed object. Any recorded history of the object that was followed is still preserved, and it will still continue following changes up until the time that this action was used.

followInstance

A read-only reference to the `IInstance` being followed, or `null` if none.

mode

Set or get a string corresponding to the follow mode, which may be one of `"time"` or `"distance"`.

delay**maxDelay****historyRate**

Set or get the corresponding behavior properties. See the [Follow behavior](#) manual entry for more details.

clearHistory()

Erases any recorded history about the object being followed. This will cause `hasFollowData` to become false and stop updating the object until enough data has been collected again. This is useful for resetting the behavior.

rewindHistory(time)

Rewinds the follow time by the given time in seconds, deletes history entries past that time, and then continues recording history. Note that it is not possible to rewind further than the max delay, as data beyond that time is erased. This action allows for implementing a 'rewind time' feature where an object can go backwards in time and then continue from a different location, while preserving the history prior to the time it continued from.

hasFollowData

A read-only boolean indicating if the behavior has enough data to be able to start following on a delay. For example if the behavior starts following an object on a 5 second time delay, then for the first 5 seconds there is no follow data and so the object will not be updated, and `hasFollowData` will be false. Once 5 seconds has elapsed it then starts updating and `hasFollowData` will be true. Note that if following starts with `fromCurrentPosition` set to `true`, then that counts as having follow data immediately.

setFollowingProperty(prop, isEnabled)**isFollowingProperty(prop)**

Set or get whether a built-in property is being followed. See the section *Built-in property strings* above for the strings that can be specified for `prop`.

setPropertyInterpolation(prop, interpolation)**getPropertyInterpolation(prop)**

Set or get the interpolation mode of one of the built-in properties. See the section *Built-in property strings* above for the strings that can be specified for `prop`, and *Interpolation*

strings for the strings that can be specified for `interp`. Generally this is used to change one of the built-in properties from smooth interpolation to step interpolation. For example mirroring an object with the Platform behavior should always update the width instantly, and not interpolate any in-between values.

startFollowingCustomProperty(customProperty, interpolation)

stopFollowingCustomProperty(customProperty)

isFollowingCustomProperty(customProperty)

Start, stop, or test if following a custom property. This allows the Follow behavior to track a custom value other than one of the built-in properties such as the X and Y position. Multiple custom properties can be followed, each identified by a case-insensitive string. The `interpolation` parameter determines how values in between history entries are determined (see the section *Interpolation strings* for the strings that can be specified). Step does not interpolate and just uses the previous history entry. Linear interpolation is suitable for linear values like position and size. Angular interpolation is suitable for rotational values like angles. Note that if a custom property value is a string, then only step interpolation is supported. The value to be recorded must be set with `setCustomPropertyValue()`, and then the value to be followed can be retrieved with `getDelayedCustomPropertyValue()`.

setCustomPropertyValue(customProperty, value)

Set the current value of a custom property that is being followed. The custom property is identified by a case-insensitive string. The `value` parameter can be either a string or a number, but if it is a string then it will only use step interpolation mode. This method should be used every tick while following a custom property, so that the latest value is available when the behavior decides to add a history entry.

getDelayedCustomPropertyValue(customProperty)

Retrieve the current value to be followed for a custom property, specified by a case-insensitive string.

isPaused

Set or get whether following is paused. While paused, no further history is recorded, but it also stops advancing the follow time. Upon resuming, the behavior will restart recording the history of the followed object. If the followed object has substantially changed while following was paused, then it will skip to the new position as it follows the history, as no changes in between will have been saved.

saveHistoryToJSON(maxDelay = 0)

Save the current recorded history of the followed object to a JSON object. This can then be loaded again later using `loadHistoryFromJSON()`. The `maxDelay` parameter can be used to save only a portion of the most recent history, rather than all the recorded history, suitable for

a record/replay feature when the recording duration is less than the behavior's max delay. If `maxDelay` is 0, then it saves all the history.

loadHistoryFromJSON(json)

Load the recorded history of the object being followed from a JSON object previously saved by `saveHistoryToJSON()`. This also sets the delay to the time of the oldest history entry loaded, so it then immediately follows the amount of data originally saved. This allows for creating a record/replay feature.

isEnabled

Set or get a boolean indicating whether the behavior is enabled. If disabled it will be completely inactive, without recording any history or updating the object at all.