

FloatFieldInterpolator

Table of Contents

- Description
- Fields
 - Example

Extends: [Node](#)

Description

The **FloatFieldInterpolator** node class specifies a keyframe animation sequence to be applied to a floating point field of a node (such as, an opacity, width or height value.)

All field interpolators include a set of key/keyValue pairs that define a keyframe of the animation. Field interpolators are generally used as children of an **Animation** node. As the animation progresses, it sets the *fraction* field of its field interpolators to a value between 0 and 1, indicating the percentage of the Animation's progress. The keyframes of the interpolator include a "key", the percentage where the keyframe should occur, and a "keyValue", the value that the field should have at that percentage.

For example, if a **FloatFieldInterpolator** had three keyframes, (0.0, 10.0), (0.4, 200.0) and (1.0, 87.0), then when the interpolator's *fraction* field was 0.0 (i.e. 0%), the field would be set to 10.0. When *fraction* was 0.4 (i.e. 40%), the field would be set to 200.0. When *fraction* was 1.0 (i.e. 100%), the field would be set to 87.0.

For values of *fraction* between 0.0 and 0.4 (e.g. 0.2 or 20%), the field value is determined by linearly interpolating the keyValues for the first two keyframes. In this case, since the key of 0.2 is halfway between the key at 0.0 and the key at 0.4, the field would be set to $10.0 + 0.5 * (10.0 + 200.0) = 105.0$. Similarly, when *fraction* is between the second and third keys (i.e. between 0.4 and 1.0), the field value is determined by linearly interpolating the keyValues of the second and third keyframes.

If the first keyframe has a key percentage greater than zero, then the field value will be equal to the keyValue of the first keyframe until *fraction* reaches the first keyframe's key percentage. Similarly, if the last keyframe has a key percentage less than one, the field value will be set to the keyValue of the last keyframe from when *fraction* equals the the last keyframe's key percentage and will not change as *fraction* increases from that value to 1.0.

Note: While linearly interpolation is used to compute the keyValue's for *fraction* values between successive keys, non-linear easing functions may be applied to the *fraction* values computed by the **Animation** node, so the overall animation may vary in speed.

Fields

Field	Type	Default	Use
fieldToInterp	string	""	Specifies the field to interpolate. The string should contain the ID of a node in the scene and the name of a field of that node, separated by a dot ".". For example, "title.width" would indicate that the interpolator should be applied to the width field of a node whose id field was "title". The specified field must be of type float.
key	array of float's	[]	Specifies the key percentages for the interpolator's keyframes. Each key percentage should be a unique value from 0 to 1 indicating the percentage of the animation where the keyValue should occur. Behavior is undefined if the number of values in the key field does not match the number of values in the keyValue field.

keyValue	array of float's	[]	Specifies the key values or the interpolator's keyframes. Each value in the <i>keyValue</i> array corresponds to a value in the <i>key</i> field's array. The interpolator's behavior is undefined if the number of values in the <i>key</i> field does not match the number of values in the <i>keyValue</i> field.
fraction	float	0.0	Specifies the percentage to be used to compute a value for the field.
reverse	boolean	false	<i>This function is available in firmware 7.7 or later.</i> Set the field to <i>true</i> to play your animation in reverse.

Example

The following changes the opacity of a graphical image, gradually making it invisible, then back to visible, in 10 seconds, repeatedly.

FloatFieldInterpolator Node Class Example

```
<?xml version="1.0" encoding="utf-8" ?>

<!--***** Copyright 2015 Roku Corp. All Rights Reserved. *****-->

  <component name="animationfloattest" extends="Group" >

    <script type="text/brightscript" >
      <![CDATA[
        function init()
          m.top.setFocus(true)
        end function
      ]]>
    </script>

    <children>

      <Poster id="testPoster" uri="pkg:/images/rokuowds.png" width="0.0" height="0.0"
translation="[160,8]" opacity="0.0" />
      <Animation id="testAnimation" duration="10" repeat="true" control="start"
easeFunction="linear" >
        <FloatFieldInterpolator id="testFloat" key="[0.0, 0.5, 1.0]"
keyValue="[ 1.0, 0.0, 1.0 ]" fieldToInterp="testPoster.opacity" />
      </Animation>

    </children>

  </component>
```