CAKeyframeAnimation Class Reference

Graphics & Animation: Animation

2010-05-25

Contents

CAKeyframeAnimation Class Reference

Inherits from	CAPropertyAnimation : CAAnimation : NSObject
Conforms to	NSCoding (CAAnimation) NSCopying (CAAnimation) CAAction (CAAnimation) CAMediaTiming (CAAnimation) NSObject (NSObject)
Framework	/System/Library/Frameworks/QuartzCore.framework
Availability	Available in iOS 2.0 and later.
Declared in	CAAnimation.h
Companion guides	Core Animation Programming Guide Core Animation Cookbook

Overview

`CAKeyframeAnimation` provides generic keyframe animation capabilities for a layer property in the render tree. You create an `CAKeyframeAnimation` instance using the inherited `animationWithKeyPath:` method, specifying the key path of the property updated in the render tree during the animation. The animation provides a series of keyframe values, either as an array or a series of points in a `CGPathRef`. While animating, it updates the value of the property in the render tree with values calculated using the specified interpolation calculation mode.

Tasks

Providing Keyframe Values

`path` (page 8) *property*
An optional `CGPathRef` that provides the keyframe values for the receiver.

`values` (page 9) *property*
An array of objects that provide the keyframe values for the receiver.

Keyframe Timing

`keyTimes` (page 7) *property*
An optional array of `NSNumber` objects that define the duration of each keyframe segment.

`timingFunctions` (page 9) *property*
An optional array of `CAMediaTimingFunction` instances that defines the pacing of the each keyframe segment.

`calculationMode` (page 7) *property*
Specifies how intermediate keyframe values are calculated by the receiver.

Rotation Mode Attribute

`rotationMode` (page 8) *property*
Determines whether objects animating along the path rotate to match the path tangent.

Cubic Mode Attributes

`tensionValues` (page 9) *property*
An array of `NSNumber` objects that define the tightness of the curve.

`continuityValues` (page 7) *property*
An array of `NSNumber` objects that define the sharpness of the timing curve's corners.

`biasValues` (page 6) *property*
An array of `NSNumber` objects that define the position of the curve relative to a control point.

Properties

For more about Objective-C properties, see "Properties" in *The Objective-C Programming Language*.

biasValues

An array of `NSNumber` objects that define the position of the curve relative to a control point.

```
@property(copy) NSArray *biasValues
```

Discussion
This property is used only for the cubic calculation modes. Positive values move the curve before the control point while negative values move it after the control point. The first value defines the behavior of the tangent to the first control point, the second value controls the second point's tangents, and so on. If you do not specify a value for a given control point, the value 0 is used.

Availability
Available in iOS 4.0 and later.

Declared In
`CAAnimation.h`

calculationMode

Specifies how intermediate keyframe values are calculated by the receiver.

```
@property(copy) NSString *calculationMode
```

Discussion
The possible values are described in "`Value calculation modes`" (page 10). The default is `kCAAnimationLinear` (page 10).

Availability
Available in iOS 2.0 and later.

Declared In
`CAAnimation.h`

continuityValues

An array of `NSNumber` objects that define the sharpness of the timing curve's corners.

```
@property(copy) NSArray *continuityValues
```

Discussion
This property is used only for the cubic calculation modes. Positive values result in sharper corners while negative values create inverted corners. The first value defines the behavior of the tangent to the first control point, the second value controls the second point's tangents, and so on. If you do not specify a value for a given control point, the value 0 is used.

Availability
Available in iOS 4.0 and later.

Declared In
`CAAnimation.h`

keyTimes

An optional array of `NSNumber` objects that define the duration of each keyframe segment.

```
@property(copy) NSArray *keyTimes
```

Discussion
Each value in the array is a floating point number between 0.0 and 1.0 and corresponds to one element in the values array. Each element in the `keyTimes` array defines the duration of the corresponding keyframe value as a fraction of the total duration of the animation. Each element value must be greater than, or equal to, the previous value.

The appropriate values in the `keyTimes` array are dependent on the `calculationMode` (page 7) property.

- If the calculationMode is set to `kCAAnimationLinear`, the first value in the array must be 0.0 and the last value must be 1.0. Values are interpolated between the specified key times.
- If the calculationMode is set to `kCAAnimationDiscrete`, the first value in the array must be 0.0.
- If the calculationMode is set to `kCAAnimationPaced` or `kCAAnimationCubicPaced`, the `keyTimes` array is ignored.

If the values in the `keyTimes` array are invalid or inappropriate for the `calculationMode`, the `keyTimes` array is ignored.

Availability
Available in iOS 2.0 and later.

Declared In
`CAAnimation.h`

path

An optional `CGPathRef` that provides the keyframe values for the receiver.

```
@property CGPathRef path;
```

Discussion
Defaults to `nil`. Specifying a path overrides the `values` (page 9) property. Each point in the path, except for move-to points, defines a single keyframe segment for the purpose of timing and interpolation. For constant velocity animation along the path, `calculationMode` (page 7) should be set to `kCAAnimationPaced` (page 10).

Availability
Available in iOS 2.0 and later.

See Also
`@property rotationMode` (page 8)

Declared In
`CAAnimation.h`

rotationMode

Determines whether objects animating along the path rotate to match the path tangent.

```
@property(copy) NSString *rotationMode
```

Discussion
Possible values are described in "`Rotation Mode Values`" (page 10). The default is `nil`, which indicates that objects should not rotate to follow the path.

The effect of setting this property to a non-`nil` value when no path object is supplied is undefined.

Availability
Available in iOS 2.0 and later.

See Also
`@property path` (page 8)

Declared In
`CAAnimation.h`

tensionValues

An array of `NSNumber` objects that define the tightness of the curve.

```
@property(copy) NSArray *tensionValues
```

Discussion
This property is used only for the cubic calculation modes. Positive values indicate a tighter curve while negative values indicate a rounder curve. The first value defines the behavior of the tangent to the first control point, the second value controls the second point's tangents, and so on. If you do not specify a value for a given control point, the value 0 is used.

Availability
Available in iOS 4.0 and later.

Declared In
`CAAnimation.h`

timingFunctions

An optional array of `CAMediaTimingFunction` instances that defines the pacing of the each keyframe segment.

```
@property(copy) NSArray *timingFunctions
```

Discussion
If the receiver defines *n* keyframes, there must be *n*-1 objects in the `timingFunctions` array. Each timing function describes the pacing of one keyframe to keyframe segment.

Special Considerations

The inherited `timingFunction` value is always ignored.

Availability
Available in iOS 2.0 and later.

Declared In
`CAAnimation.h`

values

An array of objects that provide the keyframe values for the receiver.

```
@property(copy) NSArray *values
```

Discussion
The `values` property is ignored when the `path` (page 8) property is used.

Availability
Available in iOS 2.0 and later.

Declared In
`CAAnimation.h`

Constants

Rotation Mode Values

These constants are used by the `rotationMode` (page 8) property.

```
NSString * const kCAAnimationRotateAuto
NSString * const kCAAnimationRotateAutoReverse
```

Constants

`kCAAnimationRotateAuto`

The objects travel on a tangent to the path.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

`kCAAnimationRotateAutoReverse`

The objects travel at a 180 degree tangent to the path.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

Value calculation modes

These constants are used by the `calculationMode` (page 7) property.

```
NSString * const kCAAnimationLinear;
NSString * const kCAAnimationDiscrete;
NSString * const kCAAnimationPaced;
NSString * const kCAAnimationCubic;
NSString * const kCAAnimationCubicPaced;
```

Constants

`kCAAnimationLinear`

Simple linear calculation between keyframe values.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

`kCAAnimationDiscrete`

Each keyframe value is used in turn, no interpolated values are calculated.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

`kCAAnimationPaced`

Keyframe values are interpolated to produce an even pace throughout the animation.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

`kCAAnimationCubic`
: Intermediate frames are computed using a Catmull-Rom spline that passes through the keyframes. You can adjust the shape of the spline by specifying an optional set of tension, continuity, and bias values, which modify the spline using the standard Kochanek-Bartels form.

  Available in iOS 4.0 and later.

  Declared in `CAAnimation.h`.

`kCAAnimationCubicPaced`
: Intermediate frames are computed using the cubic scheme but the `keyTimes` and `timingFunctions` properties of the animation are ignored. Instead, timing parameters are calculated implicitly to give the animation a constant velocity.

  Available in iOS 4.0 and later.

  Declared in `CAAnimation.h`.

Document Revision History

This table describes the changes to *CAKeyframeAnimation Class Reference*.

Date	Notes
2010-05-25	Updated to include symbols introduced in iOS 4.0.
2009-05-17	Corrected kCAAnimationPaced description to indicate that this is implemented.
2007-07-24	New document that describes the class that provides keyframe interpolation of a layer property.