
CAMediaTimingFunction Class Reference

Graphics & Animation: Animation



2009-03-09



Apple Inc.
© 2009 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Mac, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

iPhone is a trademark of Apple Inc.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR

CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

CAMediaTimingFunction Class Reference 5

Overview	5
Tasks	5
Creating Timing Functions	5
Accessing the Control Points	6
Class Methods	6
functionWithControlPoints::::	6
functionWithName:	6
Instance Methods	7
getControlPointAtIndex:values:	7
initWithControlPoints::::	7
Constants	8
Predefined Timing Functions	8

Document Revision History 11

Index 13

CAMediaTimingFunction Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSObject (NSObject)
Framework	/System/Library/Frameworks/QuartzCore.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	CAMediaTimingFunction.h
Companion guides	Core Animation Programming Guide Core Animation Cookbook
Related sample code	CocoaSlides

Overview

`CAMediaTimingFunction` represents one segment of a function that defines the pacing of an animation as a timing curve. The function maps an input time normalized to the range [0,1] to an output time also in the range [0,1].

Tasks

Creating Timing Functions

+ [functionWithName:](#) (page 6)

Creates and returns a new instance of `CAMediaTimingFunction` configured with the predefined timing function specified by *name*.

+ [functionWithControlPoints:::::](#) (page 6)

Creates and returns a new instance of `CAMediaTimingFunction` timing function modeled as a cubic bezier curve using the specified control points.

- [initWithControlPoints:::::](#) (page 7)

Returns an initialized timing function modeled as a cubic bezier curve using the specified control points.

Accessing the Control Points

- [getControlPointAtIndex:values:](#) (page 7)
Returns the control point for the specified index.

Class Methods

functionWithControlPoints::::

Creates and returns a new instance of `CAMediaTimingFunction` timing function modeled as a cubic bezier curve using the specified control points.

```
+ (id)functionWithControlPoints:(float)c1x
    :(float)c1y
    :(float)c2x
    :(float)c2y
```

Parameters

c1x

A floating point number representing the x position of the c1 control point.

c1y

A floating point number representing the y position of the c1 control point.

c2x

A floating point number representing the x position of the c2 control point.

c2y

A floating point number representing the y position of the c2 control point.

Return Value

A new instance of `CAMediaTimingFunction` with the timing function specified by the provided control points.

Discussion

The end points of the bezier curve are automatically set to (0.0,0.0) and (1.0,1.0). The control points defining the bezier curve are: [(0.0,0.0), (*c1x*,*c1y*), (*c2x*,*c2y*), (1.0,1.0)].

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CAMediaTimingFunction.h`

functionWithName:

Creates and returns a new instance of `CAMediaTimingFunction` configured with the predefined timing function specified by *name*.

```
+ (id)functionWithName:(NSString *)name
```

Parameters*name*

The timing function to use as specified in “[Predefined timing functions](#)” (page 8).

Return Value

A new instance of `CAMediaTimingFunction` with the timing function specified by *name*.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

CocoaSlides

Declared In

`CAMediaTimingFunction.h`

Instance Methods

getControlPointAtIndex:values:

Returns the control point for the specified index.

```
- (void)getControlPointAtIndex:(size_t)index values:(float)ptr
```

Parameters*index*

An integer specifying the index of the control point to return.

ptr

A pointer to an array that, upon return, will contain the x and y values of the specified point.

Discussion

The value of *index* must be between 0 and 3.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CAMediaTimingFunction.h`

initWithControlPoints:::::

Returns an initialized timing function modeled as a cubic bezier curve using the specified control points.

```
- (id)initWithControlPoints:(float)c1x
    :(float)c1y
    :(float)c2x
    :(float)c2y
```

Parameters*c1x*

A floating point number representing the x position of the c1 control point.

c1y

A floating point number representing the y position of the c1 control point.

c2x

A floating point number representing the x position of the c2 control point.

c2y

A floating point number representing the y position of the c2 control point.

Return ValueAn instance of `CAMediaTimingFunction` with the timing function specified by the provided control points.**Discussion**The end points of the bezier curve are automatically set to (0.0,0.0) and (1.0,1.0). The control points defining the bezier curve are: [(0.0,0.0), (*c1x*,*c1y*), (*c2x*,*c2y*), (1.0,1.0)].**Availability**

Available in Mac OS X v10.5 and later.

Declared In`CAMediaTimingFunction.h`

Constants

Predefined Timing Functions

These constants are used to specify one of the predefined timing functions used by [functionWithName:](#) (page 6).

```
NSString * const kCAMediaTimingFunctionLinear;
NSString * const kCAMediaTimingFunctionEaseIn;
NSString * const kCAMediaTimingFunctionEaseOut;
NSString * const kCAMediaTimingFunctionEaseInEaseOut;
NSString * const kCAMediaTimingFunctionDefault;
```

Constants`kCAMediaTimingFunctionLinear`

Specifies linear pacing. A linear pacing causes an animation to occur evenly over its duration.

Available in Mac OS X v10.5 and later.

Declared in `CAMediaTimingFunction.h`.`kCAMediaTimingFunctionEaseIn`

Specifies ease-in pacing. Ease-in pacing causes the animation to begin slowly, and then speed up as it progresses.

Available in Mac OS X v10.5 and later.

Declared in `CAMediaTimingFunction.h`.`kCAMediaTimingFunctionEaseOut`

Specifies ease-out pacing. An ease-out pacing causes the animation to begin quickly, and then slow as it completes.

Available in Mac OS X v10.5 and later.

Declared in `CAMediaTimingFunction.h`.

`kCAMediaTimingFunctionEaseInEaseOut`

Specifies ease-in ease-out pacing. An ease-in ease-out animation begins slowly, accelerates through the middle of its duration, and then slows again before completing.

Available in Mac OS X v10.5 and later.

Declared in `CAMediaTimingFunction.h`.

`kCAMediaTimingFunctionDefault`

Specifies the timing function used as the default by most animations. It approximates a bezier timing function using the control points [(0.0,0.0), (0.25,0.1), (0.25,0.1), (1.0,1.0)]. By using this constant you ensure that your animations will use the current default timing.

Available in Mac OS X v10.6 and later.

Declared in `CAMediaTimingFunction.h`.

Declared In

`CAMediaTimingFunction.h`

Document Revision History

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

Date	Notes
2009-03-09	Updated for iPhone OS 3.0. Added new constant for default timing function.
2008-07-11	Corrected descriptions of the predefined timing functions.
2007-07-24	New document that describes the class that encapsulates the pacing of an animation as a timing curve.

REVISION HISTORY

Document Revision History

Index

F

`functionWithControlPoints:::` **class method** [6](#)
`functionWithName:` **class method** [6](#)

G

`getControlPointAtIndex:values:` **instance method** [7](#)

I

`initWithControlPoints:::` **instance method** [7](#)

K

`kCAMEdiaTimingFunctionDefault` **constant** [9](#)
`kCAMEdiaTimingFunctionEaseIn` **constant** [8](#)
`kCAMEdiaTimingFunctionEaseInEaseOut` **constant** [9](#)
`kCAMEdiaTimingFunctionEaseOut` **constant** [8](#)
`kCAMEdiaTimingFunctionLinear` **constant** [8](#)

P

Predefined Timing Functions [8](#)