UIDynamicBehavior Class Reference

Developer

Contents

UIDynamicBehavior Class Reference 3
Overview 3
Tasks 4
    Configuring a Dynamic Behavior 4
Properties 4
    action 4
    childBehaviors 5
Instance Methods 5
    addChildBehavior: 5
    removeChildBehavior: 6

Document Revision History 7

UIDynamicBehavior Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/UIKit.framework
Availability	Available in iOS 7.0 and later.
Declared in	UIDynamicBehavior.h

Overview

> **Important:** This is a preliminary document for an API or technology in development. Although this document has been reviewed for technical accuracy, it is not final. This Apple confidential information is for use only by registered members of the applicable Apple Developer program. Apple is supplying this confidential information to help you plan for the adoption of the technologies and programming interfaces described herein. This information is subject to change, and software implemented according to this document should be tested with final operating system software and final documentation. Newer versions of this document may be provided with future seeds of the API or technology.

A dynamic behavior confers a behavioral configuration on one or more dynamic items for their participation in two-dimensional animation.

A **dynamic item** is any iOS or custom object that conforms to the `UIDynamicItem` protocol. The `UIView` and `UICollectionViewLayoutAttributes` classes implement this protocol starting in iOS 7.0. You can use a custom object as a dynamic item for such purposes as reacting to rotation or position changes computed by a dynamic animator—an instance of the `UIDynamicAnimator` class.

This parent class, `UIDynamicBehavior`, is inherited by the primitive dynamic behavior classes `UIAttachmentBehavior`, `UICollisionBehavior`, `UIGravityBehavior`, `UIDynamicItemBehavior`, `UIPushBehavior`, and `UISnapBehavior`.

You can subclass `UIDynamicBehavior`. By using the `addChildBehavior:` (page 5) method in an instance of this class or in a custom subclass, you can create composite behaviors of your own design.

When you subclass `UIDynamicBehavior`, you typically need to provide one or more initializers, along with other housekeeping methods such as those implemented in the iOS primitive dynamic behaviors.

To perform per-step logic in a dynamic animation, provide a block object using the `action` (page 4) property.

Tasks

Configuring a Dynamic Behavior

`action` (page 4) *property*
The block you want to execute during dynamic animation.

– `addChildBehavior:` (page 5)
Adds a dynamic behavior, as a child, to a custom dynamic behavior.

`childBehaviors` (page 5) *property*
Returns the array of dynamic behaviors that are children of a custom dynamic behavior. (read-only)

– `removeChildBehavior:` (page 6)
Removes a child dynamic behavior from a custom dynamic behavior.

Properties

action

The block you want to execute during dynamic animation.

```
@property(nonatomic, copy) void (^action)(void)
```

Discussion

The dynamic animator calls the action block on every animation step.

Availability

Available in iOS 7.0 and later.

See Also

`@property condition` (page ?)

Declared in

`UIDynamicBehavior.h`

childBehaviors

Returns the array of dynamic behaviors that are children of a custom dynamic behavior. (read-only)

```
@property(nonatomic, readonly, copy) NSArray *childBehaviors
```

Discussion

Only custom subclasses of the class can have child behaviors.

Availability

Available in iOS 7.0 and later.

See Also

– `addChildBehavior:` (page 5)
– `removeChildBehavior:` (page 6)

Declared in

`UIDynamicBehavior.h`

Instance Methods

addChildBehavior:

Adds a dynamic behavior, as a child, to a custom dynamic behavior.

```
- (void)addChildBehavior:(UIDynamicBehavior *)behavior
```

Parameters

`behavior`

The dynamic behavior you want to add as a child.

Discussion

Call this method only on custom subclasses of the `UIDynamicBehavior` class.

Special Considerations

If you attempt to call this method on a UIKit concrete dynamic behavior (such as an instance of `UICollisionBehavior`), the behavior raises an exception.

Availability

Available in iOS 7.0 and later.

See Also

`@property childBehaviors` (page 5)

2013-06-10 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

– `removeChildBehavior:` (page 6)

Declared in
`UIDynamicBehavior.h`

removeChildBehavior:

Removes a child dynamic behavior from a custom dynamic behavior.

```
- (void)removeChildBehavior:(UIDynamicBehavior *)behavior
```

Parameters

`behavior`

The child dynamic behavior you want to remove.

Discussion

This method applies only to custom subclasses of the `UIDynamicBehavior` class. UIKit concrete dynamic behaviors (such as an instance of `UICollisionBehavior`) cannot have child behaviors.

Availability

Available in iOS 7.0 and later.

See Also

– `addChildBehavior:` (page 5)
`@property childBehaviors` (page 5)

Declared in

`UIDynamicBehavior.h`

Document Revision History

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

Date	Notes
2013-06-10	New document that describes an abstract dynamic behavior for a view.

Apple Inc.
Copyright © 2013 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.

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.

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

Apple and the Apple logo are trademarks of Apple Inc., registered in the U.S. and other countries.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

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.