UIDynamicItemBehavior Class Reference

Developer

Contents

UIDynamicItemBehavior Class Reference

Inherits from	UIDynamicBehavior : NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/UIKit.framework
Availability	Available in iOS 7.0 and later.
Declared in	UIDynamicItemBehavior.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 item behavior represents a base dynamic animation configuration for one or more dynamic items. Each of its properties overrides a corresponding default value.

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.

One notable and common use of a dynamic item behavior is to confer a velocity to a dynamic item to match the ending velocity of a user gesture.

To use a dynamic item behavior with a dynamic item, perform these two steps:

1. Associate the item with the behavior using the `addItem:` (page 9) method, or initialize a new dynamic item behavior with an array of items using the `initWithItems:` (page 10) method

2. Enable the behavior by adding it to an animator using the `addBehavior:` method

The coordinate system that pertains to a dynamic item behavior, and the types of dynamic items you can use with the behavior, depend on how you initialized the associated animator. For details, read the Overview of *UIDynamicAnimator Class Reference* .

You can disable rotation for a dynamic item behavior's items by returning `NO` from the `allowsRotation` (page 5) property. To configure interaction among the behavior's items, use the `elasticity` (page 7) and `friction` (page 7) properties.

You can include a dynamic item behavior in a custom, composite behavior by starting with a `UIDynamicBehavior` object and adding a dynamic item behavior with the `addChildBehavior:` method. If you want to influence a dynamic item behavior at each step of a dynamic animation, implement the inherited `action` method.

If you add more than one dynamic item behavior to an animator, you effectively create a behavior tree. Only one configuration of a given property applies to any given dynamic item. For a property configured in more than one dynamic item behavior, the last one in the behavior tree, starting from the dynamic animator and going depth first toward the dynamic item, wins.

In the case of an animator with exactly one dynamic item behavior, you can restore default values for all dynamic item behavior properties by removing the behavior. In the case of an animator to which you've applied multiple dynamic item behaviors, removing one takes its property contribution out of the behavior tree.

Tasks

Initializing and Managing a Dynamic Item Behavior

– `addItem:` (page 9)
Adds a dynamic item to the dynamic item behavior's item array.

– `initWithItems:` (page 10)
Initializes a dynamic item behavior with an array of dynamic items.

– `removeItem:` (page 11)
Removes a specific dynamic item from the dynamic item behavior.

`items` (page 7) *property*
Returns the set of dynamic items you've added to the dynamic item behavior. (read-only)

Configuring a Dynamic Item Behavior

– `addAngularVelocity:forItem:` (page 8)
  Adds a specified angular velocity to a dynamic item.

– `addLinearVelocity:forItem:` (page 9)
  Adds a specified linear velocity to a dynamic item.

`allowsRotation` (page 5) *property*
  Specifies whether rotation is allowed for the behavior's dynamic items.

`angularResistance` (page 6) *property*
  The angular resistance for the behavior's dynamic items.

– `angularVelocityForItem:` (page 10)
  Returns the angular velocity for a specified dynamic item.

`density` (page 6) *property*
  The relative mass density of the behavior's dynamic items.

`elasticity` (page 7) *property*
  The amount of elasticity applied to collisions for the behavior's dynamic items.

`friction` (page 7) *property*
  The linear resistance for the behavior's dynamic items when two slide against each other.

`resistance` (page 8) *property*
  The linear resistance for the behavior's dynamic items, which reduces their linear velocity over time.

– `linearVelocityForItem:` (page 10)
  Returns the linear velocity for a specified dynamic item.

Properties

allowsRotation

Specifies whether rotation is allowed for the behavior's dynamic items.

```
@property(readwrite, nonatomic) BOOL allowsRotation
```

Discussion
Default value is `YES`. To disable rotation, set this property to `NO`.

Availability
Available in iOS 7.0 and later.

Declared in
`UIDynamicItemBehavior.h`

angularResistance

The angular resistance for the behavior's dynamic items.

```
@property(readwrite, nonatomic) CGFloat angularResistance
```

Discussion
Valid range is `0` through `CGFLOAT_MAX`. The greater the value, the greater the angular damping and the faster rotation slows to a stop.

Availability
Available in iOS 7.0 and later.

Declared in
`UIDynamicItemBehavior.h`

density

The relative mass density of the behavior's dynamic items.

```
@property(readwrite, nonatomic) CGFloat density
```

Discussion
A dynamic item's relative density, along with its size, determines its effective mass when it participates in UIKit Dynamics behaviors—including friction, collisions, pushes, and so on. For example, say you have two dynamic items with the same density but different sizes: item one is `100 x 100` points and item two is `100 x 200` points. In this example, item two has twice the effective mass of item one. In an elastic collision, these items exhibit a natural conservation of momentum according to their relative masses.

A `100 x 100` point dynamic item with a density of `1.0`, to which you apply a force (via a push behavior) of magnitude `1.0`, accelerates at `100` points per second^2.

Availability
Available in iOS 7.0 and later.

Declared in
`UIDynamicItemBehavior.h`

elasticity

The amount of elasticity applied to collisions for the behavior's dynamic items.

```
@property(readwrite, nonatomic) CGFloat elasticity
```

Discussion

Default value is `0.0`. Valid range is from `0.0` for no bounce upon collision, to `1.0` for completely elastic collisions.

Availability

Available in iOS 7.0 and later.

Declared in

`UIDynamicItemBehavior.h`

friction

The linear resistance for the behavior's dynamic items when two slide against each other.

```
@property(readwrite, nonatomic) CGFloat friction
```

Discussion

Default value is `0.0`, which corresponds to no friction. Use a value of `1.0` to apply strong friction. To apply an even stronger friction, you can use higher numbers.

Availability

Available in iOS 7.0 and later.

Declared in

`UIDynamicItemBehavior.h`

items

Returns the set of dynamic items you've added to the dynamic item behavior. (read-only)

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

Discussion

Availability

Available in iOS 7.0 and later.

Declared in
`UIDynamicItemBehavior.h`

resistance

The linear resistance for the behavior's dynamic items, which reduces their linear velocity over time.

```
@property(readwrite, nonatomic) CGFloat resistance
```

Discussion
Default value is `0.0`. Valid range is from `0.0` for no velocity damping, to `CGFLOAT_MAX` for complete velocity damping. If you set this property to `1.0`, a dynamic item's motion stops as soon as there is no force applied to it.

Availability
Available in iOS 7.0 and later.

Declared in
`UIDynamicItemBehavior.h`

Instance Methods

addAngularVelocity:forItem:

Adds a specified angular velocity to a dynamic item.

```
- (void)addAngularVelocity:(CGFloat)velocity forItem:(id<UIDynamicItem>)item
```

Parameters
`velocity`
The angular velocity, expressed in radians per second, that you want to add to the specified dynamic item. Default value is `0`. Applying a negative value reduces the angular velocity by the specified amount.

`item`
The dynamic item whose angular velocity you want to increase (or decrease).

Availability
Available in iOS 7.0 and later.

Declared in
`UIDynamicItemBehavior.h`

addItem:

Adds a dynamic item to the dynamic item behavior's item array.

```
- (void)addItem:(id<UIDynamicItem>)item
```

Parameters

`item`

The dynamic item to add to the item array.

Discussion

You can add a dynamic item to one or more dynamic item behaviors. For example, you could add a dynamic item to one dynamic item behavior to configure the item's elasticity and to a second dynamic item behavior to configure its density. This is especially useful when you are defining custom, combined behaviors for your dynamic items.

Availability

Available in iOS 7.0 and later.

See Also

`– removeItem:` (page 11)

Declared in

`UIDynamicItemBehavior.h`

addLinearVelocity:forItem:

Adds a specified linear velocity to a dynamic item.

```
- (void)addLinearVelocity:(CGPoint)velocity forItem:(id<UIDynamicItem>)item
```

Parameters

`velocity`

The linear velocity, expressed in points per second, that you want to add to the specified dynamic item. Default value is `0`. Applying a negative value reduces the linear velocity by the specified amount.

`item`

The dynamic item whose linear velocity you want to increase (or decrease).

Availability

Available in iOS 7.0 and later.

Declared in

`UIDynamicItemBehavior.h`

angularVelocityForItem:

Returns the angular velocity for a specified dynamic item.

```
- (CGFloat)angularVelocityForItem:(id<UIDynamicItem>)item
```

Parameters

`item`

The dynamic item whose angular velocity you want to get.

Return Value

The angular velocity of the specified dynamic item, in radians per second.

Availability

Available in iOS 7.0 and later.

Declared in

`UIDynamicItemBehavior.h`

initWithItems:

Initializes a dynamic item behavior with an array of dynamic items.

```
- (instancetype)initWithItems:(NSArray *)items
```

Parameters

`items`

The dynamic items that you want to be subject to the dynamic item behavior.

Return Value

The initialized dynamic item behavior, or `nil` if there was a problem initializing the object.

Discussion

Availability

Available in iOS 7.0 and later.

Declared in

`UIDynamicItemBehavior.h`

linearVelocityForItem:

Returns the linear velocity for a specified dynamic item.

```
- (CGPoint)linearVelocityForItem:(id<UIDynamicItem>)item
```

Parameters

`item`

The dynamic item whose linear velocity you want to get.

Return Value

The linear velocity of the specified dynamic item, in points per second.

Availability

Available in iOS 7.0 and later.

Declared in

`UIDynamicItemBehavior.h`

removeItem:

Removes a specific dynamic item from the dynamic item behavior.

```
- (void)removeItem:(id<UIDynamicItem>)item
```

Parameters

`item`

The dynamic item that you want to remove.

Discussion

Availability

Available in iOS 7.0 and later.

Declared in

`UIDynamicItemBehavior.h`

Document Revision History

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

Date	Notes
2013-06-10	New document that describes a basic animation configuration template to apply to one or more dynamic onscreen items.

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

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.