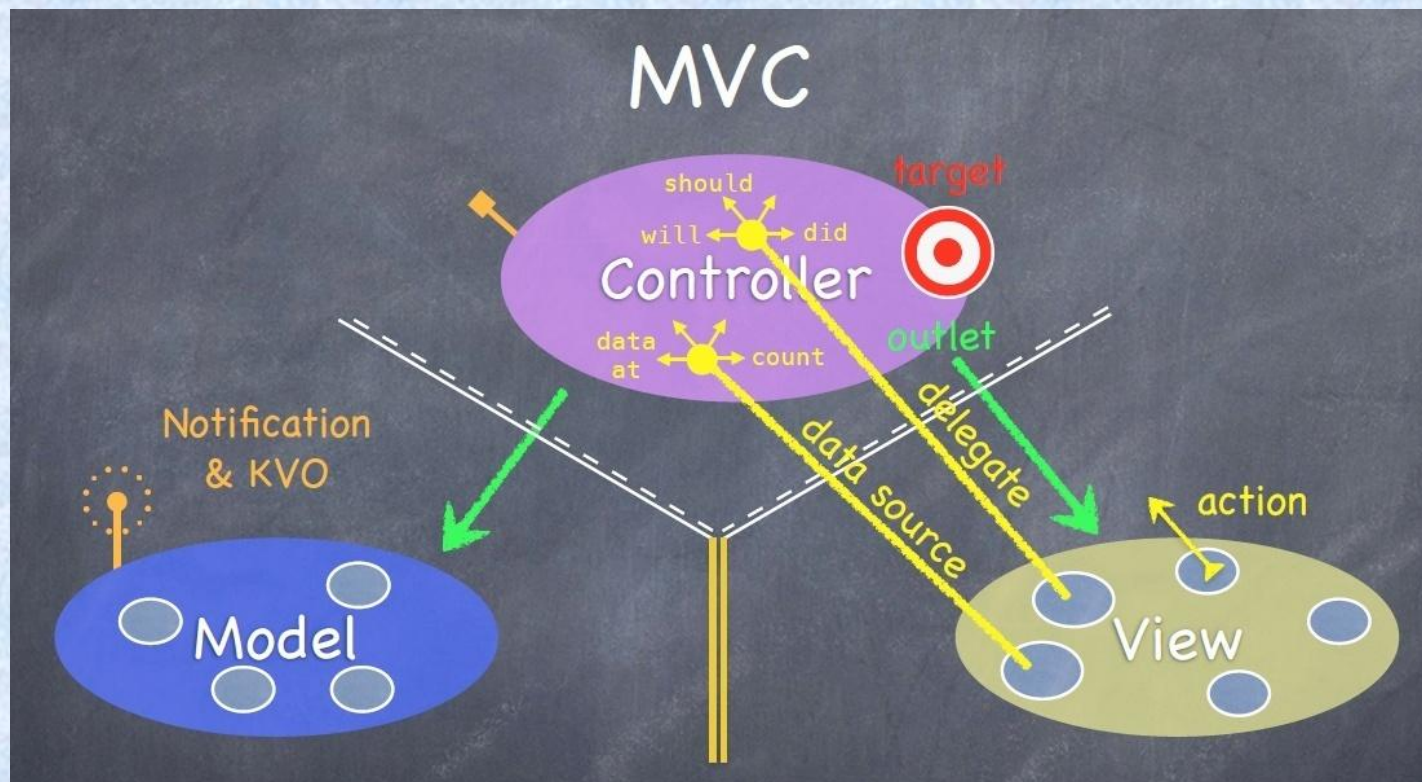




通知、手势、警告视图与动作表单



一、IOS中的通知



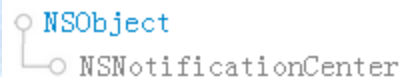


1、IOS中使用通知的步骤

- 步骤1： 创建一个通知对象。
- 步骤2： 注册通知。
- 步骤3： 发送通知。
- 步骤4： 移除通知。



2、通知涉及到类 通知中心



NSNotificationCenter

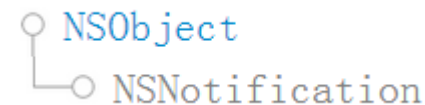
An `NSNotificationCenter` object (or simply, **notification center**) provides a mechanism for broadcasting information within a program. An `NSNotificationCenter` object is essentially a notification dispatch table.

Objects register with a notification center to receive notifications (`NSNotification` objects) using the `addObserver:selector:name:object:` or `addObserverForName:object:queue:usingBlock:` methods. Each invocation of this method specifies a set of notifications. Therefore, objects may register as observers of different notification sets by calling these methods several times.

Each running Cocoa program has a default notification center. You typically don't create your own. An `NSNotificationCenter` object can deliver notifications only within a single program. If you want to post a notification to other processes or receive notifications from other processes, use an instance of `NSDistributedNotificationCenter`.



3、通知



NSNotification

NSNotification objects encapsulate information so that it can be broadcast to other objects by an NSNotificationCenter object. An NSNotification object (referred to as a notification) contains a name, an object, and an optional dictionary. The name is a tag identifying the notification. The object is any object that the poster of the notification wants to send to observers of that notification (typically, it is the object that posted the notification). The dictionary stores other related objects, if any. NSNotification objects are immutable objects.

三个属性

@property(readonly, copy) NSString *name

@property(readonly, retain) id object

@property(readonly, copy) NSDictionary *userInfo



4、创建一个通知对象

+ defaultCenter

Returns the process's default notification center.

Declaration

OBJECTIVE-C

+ (NSNotificationCenter *)defaultCenter

Return Value

The current process's default notification center, which is used for system notifications.

`NSNotificationCenter *center = [NSNotificationCenter defaultCenter];`
每一个程序都有一个自己的通知中心，即NSNotificationCenter对象。采用defaultCenter方法就可以获得唯一的NSNotificationCenter对象



5、注册通知

- addObserver:selector:name:object:

Adds an entry to the receiver's dispatch table with an observer, a notification selector and optional criteria: notification name and sender.

Declaration

OBJECTIVE-C

```
- (void)addObserver:(id)notificationObserver  
    selector:(SEL)notificationSelector  
        name:(NSString *)notificationName  
    object:(id)notificationSender
```

接受通知的对象

处理的方法

通知名称

Parameters

<i>notificationObserver</i>	Object registering as an observer. This value must not be nil.
<i>notificationSelector</i>	Selector that specifies the message the receiver sends <i>notificationObserver</i> to notify it of the notification posting. The method specified by <i>notificationSelector</i> must have one and only one argument (an instance of <i>NSNotification</i>).
<i>notificationName</i>	<p>The name of the notification for which to register the observer; that is, only notifications with this name are delivered to the observer.</p> <p>If you pass nil, the notification center doesn't use a notification's name to decide whether to deliver it to the observer.</p>
<i>notificationSender</i>	<p>The object whose notifications the observer wants to receive; that is, only notifications sent by this sender are delivered to the observer.</p> <p>If you pass nil, the notification center doesn't use a notification's sender to decide whether to deliver it to the observer.</p>



6、发送通知

- postNotification:

Posts a given notification to the receiver.

Declaration

OBJECTIVE-C

- (void)postNotification:(NSNotification *)notification

Parameters

notification	The notification to post. This value must not be nil.
--------------	-------------------------------------------------------

```
- postNotification:  
  
- postNotificationName:object:  
  
- postNotificationName:object:userInfo:
```




7、移除通知

- removeObserver:

Removes all the entries specifying a given observer from the receiver's dispatch table.

Declaration

OBJECTIVE-C

```
- (void)removeObserver:(id)notificationObserver
```

Parameters

<i>notificationObserver</i>	The observer to remove. Must not be nil.
-----------------------------	------------------------------------------

[center removeObserver:self];



- addObserver:name:object:

Removes matching entries from the receiver's dispatch table.

Declaration

OBJECTIVE-C

```
- (void)removeObserver:(id)notificationObserver  
    name:(NSString *)notificationName  
    object:(id)notificationSender
```

Parameters

<i>notificationObserver</i>	Observer to remove from the dispatch table. Specify an observer to remove only entries for this observer. Must not be <code>nil</code> , or message will have no effect.
<i>notificationName</i>	Name of the notification to remove from dispatch table. Specify a notification name to remove only entries that specify this notification name. When <code>nil</code> , the receiver does not use notification names as criteria for removal.
<i>notificationSender</i>	Sender to remove from the dispatch table. Specify a notification sender to remove only entries that specify this sender. When <code>nil</code> , the receiver does not use notification senders as criteria for removal.

[center addObserver:self name:UIContentSizeCategoryDidChangeNotification object:nil];



```
#import "ViewController.h"

@interface ViewController ()
@property (weak, nonatomic) IBOutlet UITextView *body;
@end

@implementation ViewController
-(void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];
    [self useSystemFonts];
    [[NSNotificationCenter defaultCenter] addObserver:self
        selector:@selector(SystemFontsChanged:) name:
        UIContentSizeCategoryDidChangeNotification object:nil ];
}

-(void)viewWillDisappear:(BOOL)animated
{
    [super viewWillDisappear:animated];
    [[NSNotificationCenter defaultCenter] removeObserver:self
        name:UIContentSizeCategoryDidChangeNotification object:nil
    ];
}

-(void) SystemFontsChanged:(NSNotification *) notification
{
    [self useSystemFonts];
}

-(void) useSystemFonts
{
    self.body.font=[UIFont preferredFontForTextStyle:
        UIFontTextStyleBody];
}
```

应用示例



二、IOS中使用手势

1、使用手势的步骤

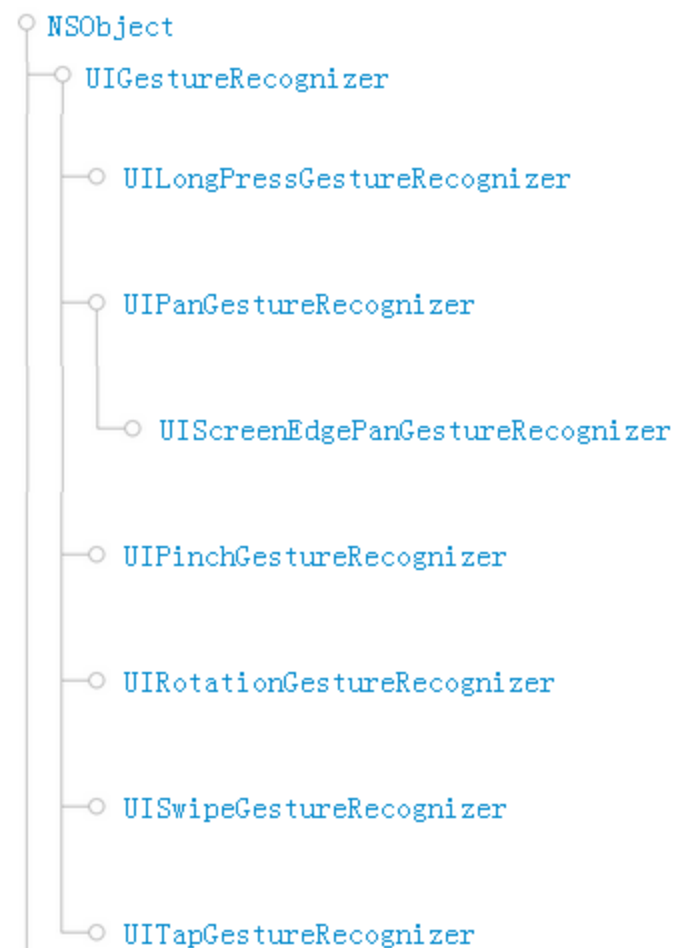
- 步骤1： 创建一个手势识别器对象。给需要手势功能的视图添加手势识别器对象。
 - 步骤2： 添加一个方法来响应手势。
-



2、手势涉及到类

UIGestureRecognizer

UIGestureRecognizer is an **abstract base class** for concrete gesture-recognizer classes. A gesture-recognizer object—or, simply, a gesture recognizer—decouples the logic for recognizing a gesture and acting on that recognition. When one of these objects recognizes a common gesture or, in some cases, a change in the gesture, it sends an action message to each designated target object.





3、UIGestureRecognizer的几个重要的属性

state

The current state of the gesture recognizer. (read-only)

Declaration

OBJECTIVE-C

```
@property(nonatomic, readonly) UIGestureRecognizerState state
```

通过该属性可以知道手势识别器的当前状态



UIGestureRecognizerState

```
typedef enum {  
    UIGestureRecognizerStatePossible,  
  
    UIGestureRecognizerStateBegan,  
    UIGestureRecognizerStateChanged,  
    UIGestureRecognizerStateEnded,  
    UIGestureRecognizerStateCancelled,  
  
    UIGestureRecognizerStateFailed,  
  
    UIGestureRecognizerStateRecognized = UIGestureRecognizerStateEnded  
} UIGestureRecognizerState;
```




● UIGestureRecognizerStatePossible

The gesture recognizer has not yet recognized its gesture, but may be evaluating touch events. This is the default state.

● UIGestureRecognizerStateBegan

The gesture recognizer has received touch objects recognized as a continuous gesture. It sends its action message (or messages) at the next cycle of the run loop.

● UIGestureRecognizerStateChanged

The gesture recognizer has received touches recognized as a change to a continuous gesture. It sends its action message (or messages) at the next cycle of the run loop.

● UIGestureRecognizerStateEnded

The gesture recognizer has received touches recognized as the end of a continuous gesture. It sends its action message (or messages) at the next cycle of the run loop and resets its state to UIGestureRecognizerStatePossible.

● UIGestureRecognizerStateCancelled

The gesture recognizer has received touches resulting in the cancellation of a continuous gesture. It sends its action message (or messages) at the next cycle of the run loop and resets its state to UIGestureRecognizerStatePossible.

● UIGestureRecognizerStateFailed

The gesture recognizer has received a multi-touch sequence that it cannot recognize as its gesture. No action message is sent and the gesture recognizer is reset to UIGestureRecognizerStatePossible.

● UIGestureRecognizerStateRecognized

The gesture recognizer has received a multi-touch sequence that it recognizes as its gesture. It sends its action message (or messages) at the next cycle of the run loop and resets its state to UIGestureRecognizerStatePossible.



view

The view the gesture recognizer is attached to. (read-only)

Declaration

OBJECTIVE-C

```
@property(n nonatomic, readonly) UIView *view
```

通过该属性可以知道手势识别器的和那个视图关联



enabled

A Boolean property that indicates whether the gesture recognizer is enabled.

Declaration

OBJECTIVE-C

```
@property(nonatomic, getter=isEnabled) BOOL enabled
```

通过该属性决定手势识别器是否可用



4、创建和初始化手势识别器对象

- 方法一：利用代码生成

```
UIPanGestureRecognizer *gr =  
    [[UIPanGestureRecognizer alloc] initWithTarget:self.view action:@selector(pan:)];  
[self.view addGestureRecognizer:pangr];
```

添加手势的视图

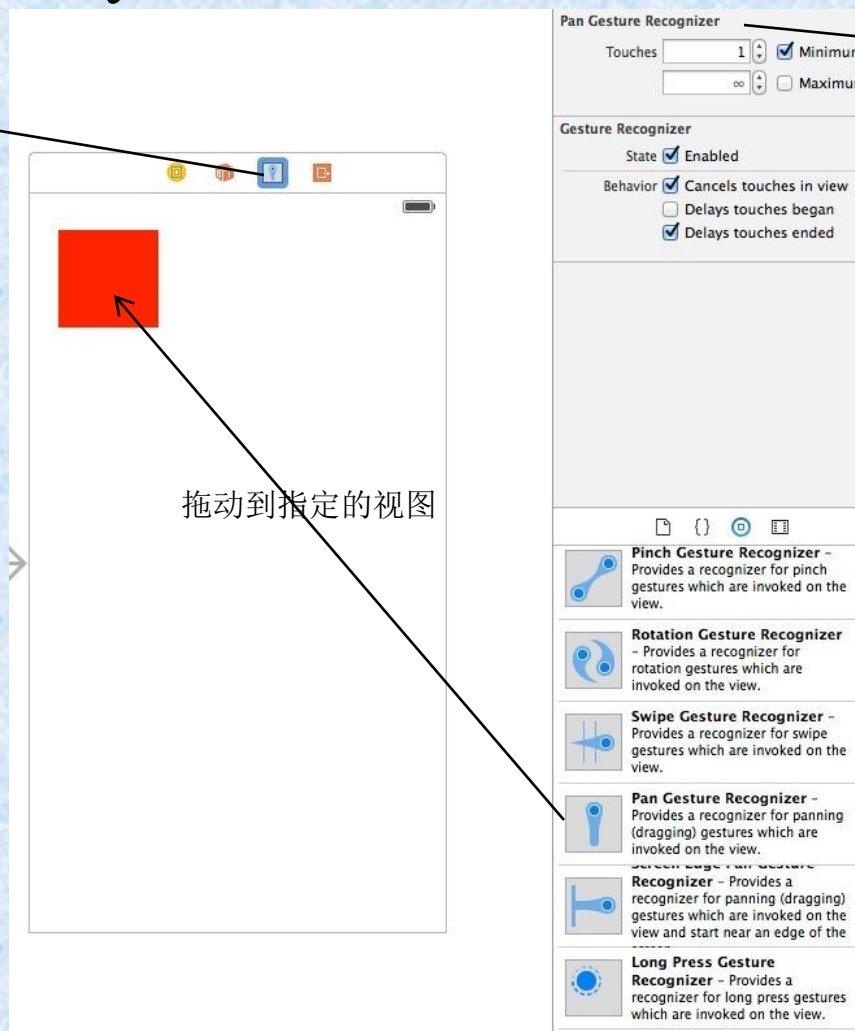
处理手势的方法

```
-(void)pan:(UIPanGestureRecognizer *)recognizer  
{  
  
    //处理手势的代码  
  
}
```




• 方法二：在Storyboard利用IB创建

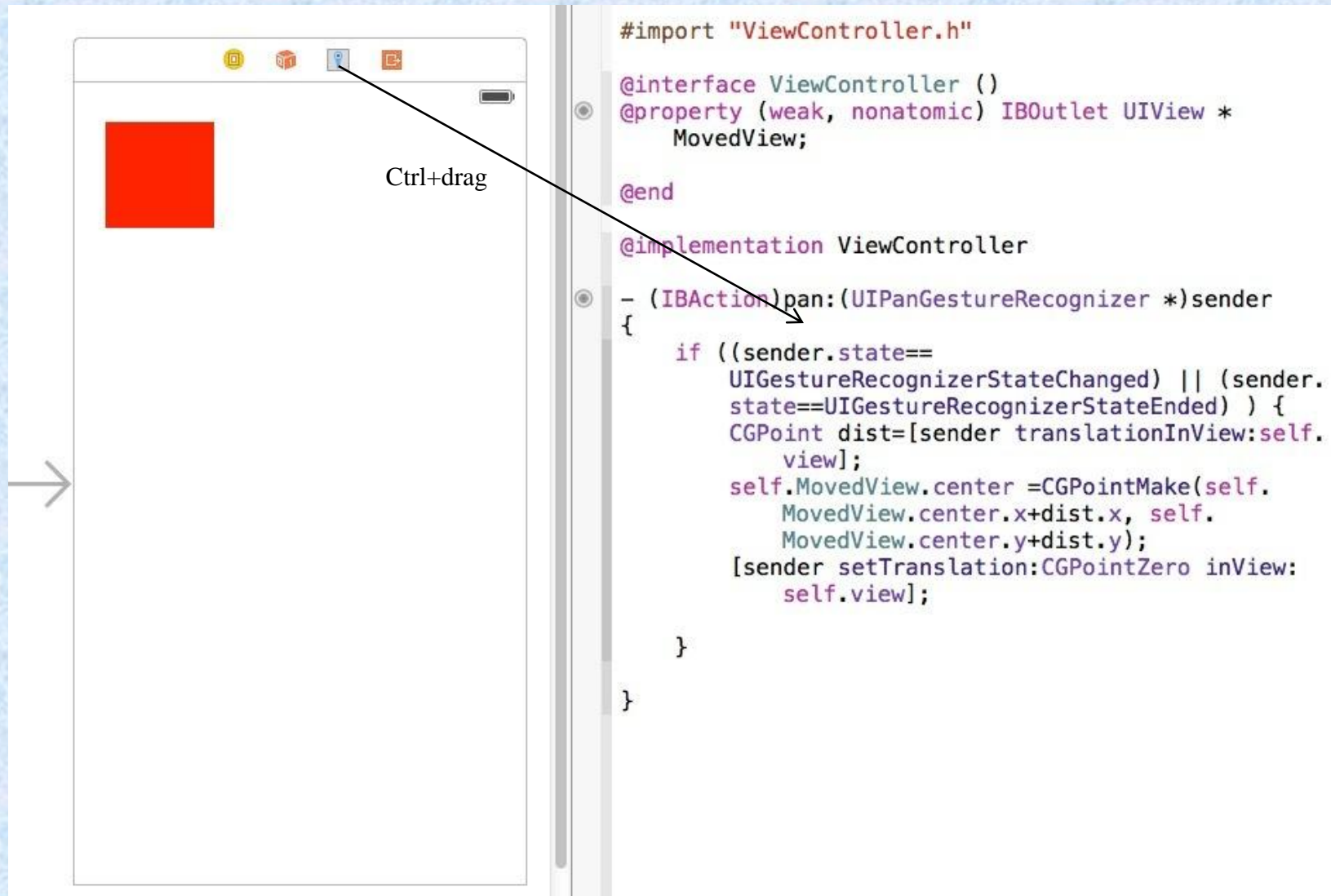
添加的手势识别器



拖动到指定的视图

可以在属性检查器中设置属性。

推荐使用Storyboard进行创建



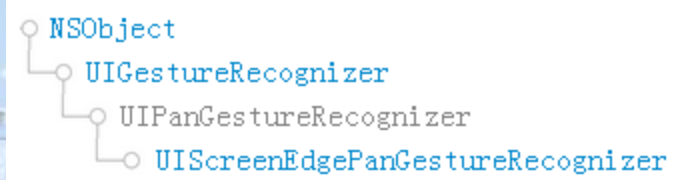
Ctrl+drag

```
#import "ViewController.h"

@interface ViewController ()
@property (weak, nonatomic) IBOutlet UIView *
    MovedView;
@end

@implementation ViewController

- (IBAction)pan:(UIPanGestureRecognizer *)sender
{
    if ((sender.state==
        UIGestureRecognizerStateChanged) || (sender.
        state==UIGestureRecognizerStateEnded) ) {
        CGPoint dist=[sender translationInView:self.
            view];
        self.MovedView.center =CGPointMake(self.
            MovedView.center.x+dist.x, self.
            MovedView.center.y+dist.y);
        [sender setTranslation:CGPointZero inView:
            self.view];
    }
}
```

5、UIPanGestureRecognizer

UIPanGestureRecognizer is a concrete subclass of UIGestureRecognizer that looks for panning (dragging) gestures. The user must be pressing one or more fingers on a view while they pan it. Clients implementing the action method for this gesture recognizer can ask it for the current translation and velocity of the gesture.

A panning gesture is continuous. It begins ([UIGestureRecognizerStateBegan](#)) when the minimum number of fingers allowed ([minimumNumberOfTouches](#)) has moved enough to be considered a pan. It changes ([UIGestureRecognizerStateChanged](#)) when a finger moves while at least the minimum number of fingers are pressed down. It ends ([UIGestureRecognizerStateEnded](#)) when all fingers are lifted.



UIGestureRecognizer的几个重要的方法

- translationInView:

The translation of the pan gesture in the coordinate system of the specified view.

Declaration

OBJECTIVE-C

```
- (CGPoint)translationInView:(UIView *)view
```

Parameters

<i>view</i>	The view in whose coordinate system the translation of the pan gesture should be computed. If you want to adjust a view's location to keep it under the user's finger, request the translation in that view's superview's coordinate system.
-------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Return Value

A point identifying the new location of a view in the coordinate system of its designated superview.

通过该方法可获得移动的位置距离



UIGestureRecognizer的几个重要的方法

- `setTranslation:inView:`

Sets the translation value in the coordinate system of the specified view.

Declaration

OBJECTIVE-C

```
- (void)setTranslation:(CGPoint) translation  
        inView:(UIView *)view
```

Parameters

<i>translation</i>	A point that identifies the new translation value.
<i>view</i>	A view in whose coordinate system the translation is to occur.

通过该方法可设置移动的视图的初始位置



UIGestureRecognizer的几个重要的方法

- `velocityInView:`

The velocity of the pan gesture in the coordinate system of the specified view.

Declaration

OBJECTIVE-C

```
- (CGPoint)velocityInView:(UIView *)view
```

Parameters

<i>view</i>	The view in whose coordinate system the velocity of the pan gesture is computed.
-------------	----------------------------------------------------------------------------------

Return Value

The velocity of the pan gesture, which is expressed in points per second. The velocity is broken into horizontal and vertical components.

通过该方法获得移动速度相关信息



5、 其它的一些手势识别器

UIPinchGestureRecognizer

@property CGFloat scale; // 只读，放大或缩小的尺度

@property (readonly) CGFloat velocity; // 缩放的速度

UIRotationGestureRecognizer

@property CGFloat rotation; // 旋转的角度

@property (readonly) CGFloat velocity; // 只读，旋转的速度

UITapGestureRecognizer

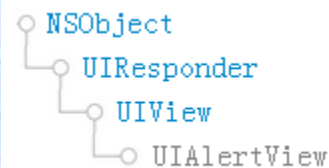
@property NSUInteger numberOfTapsRequired; // 单次还是多次轻拍

@property NSUInteger numberOfTouchesRequired; // 手指数量



三、警告视图与动作表单

1、警告视图 (UIAlertView)



```
- (IBAction)Demo:(UIButton *)sender {
    UIAlertView *alert=[[UIAlertView alloc]initWithTitle:@"警告提示" message:@"这是一个警告视图的测试" delegate:nil cancelButtonTitle:@"YES"
    otherButtonTitles:@"NO", nil];
    [alert show];
}
```




警告视图

警告提示

这是一个警告视图的测试

YES

NO

OK

```
UIAlertView *alert=[[UIAlertView alloc]initWithTitle:@"警告提示" message:@"这是一个警告视图的测试" delegate:nil cancelButtonTitle:@"OK"
otherButtonTitles:@"YES",@"NO", nil];
[alert show];
```



iOS Simulator - iPhone 5s - iPhone 5s / iOS 8...

Carrier

7:40 PM

警告视图

请等待

程序正在加载中...

```
UIAlertView *alert=[[UIAlertView alloc]initWithTitle:@"请等待"  
message:@"程序正在加载中..." delegate:nil cancelButtonTitle:nil  
otherButtonTitles:nil, nil];  
[alert show];
```




iOS Simulator - iPhone 5s - iPhone 5s / iOS 8...

Carrier

7:46 PM

警告视图

注意

请输入用户名和密码

user

••

OK

```
UIAlertView *alert=[[UIAlertView alloc]initWithTitle:@"注意"  
    message:@"请输入用户名和密码" delegate:nil cancelButtonTitle:@"OK"  
    otherButtonTitles:nil, nil];  
alert.alertViewStyle=UIAlertViewStyleLoginAndPasswordInput;  
[alert show];
```

```
typedef enum {  
    UIAlertViewStyleDefault = 0,  
    UIAlertViewStyleSecureTextInput ,  
    UIAlertViewStylePlainTextInput ,  
    UIAlertViewStyleLoginAndPasswordInput  
} UIAlertViewStyle;
```



响应警告视图



```
UIAlertView *alert=[[UIAlertView alloc]initWithTitle:@"颜色选择"
message:@"这是一个关于警告视图颜色选择的测试" delegate:self
 cancelButtonTitle:@"Red" otherButtonTitles:@"Blue", nil];
[alert show];
```

委托给自己

```
-(void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)buttonIndex
{
    if ([[alertView buttonTitleAtIndex:buttonIndex] isEqualToString:@"Red"]) {
        self.view.backgroundColor=[UIColor redColor];
    }else{
        self.view.backgroundColor=[UIColor blueColor];
    }
}
```

```
@interface ViewController : UIViewController<UIAlertViewDelegate>

@end
```

委托协议



如何让上面警告视图消失？

- dismissWithClickedButtonIndex:animated:

Dismisses the receiver, optionally with animation.

Declaration

OBJECTIVE-C

```
- (void)dismissWithClickedButtonIndex:(NSInteger)buttonIndex  
    animated:(BOOL)animated
```

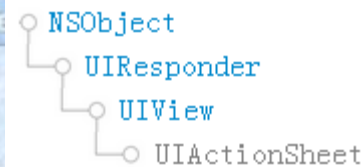
Parameters

<i>buttonIndex</i>	The index of the button that was clicked just before invoking this method. The button indices start at 0.
<i>animated</i>	YES if the receiver should be removed by animating it first; otherwise, NO if it should be removed immediately with no animation.

```
UIAlertView *alert=[[UIAlertView alloc]initWithTitle:@"请等待"  
    message:@"程序正在加载中..." delegate:nil cancelButtonTitle:nil  
    otherButtonTitles:nil, nil];  
dispatch_queue_t queue =dispatch_get_global_queue  
    (DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);  
dispatch_async(queue, ^{  
    [NSThread sleepForTimeInterval:3.0];  
    [alert dismissWithClickedButtonIndex:0 animated:YES];  
});  
[alert show];
```



2、动作表单 (UIActionSheet)



```
UIActionSheet *actionsheet=[[UIActionSheet alloc] initWithTitle:@"颜色选择"  
    delegate:self cancelButtonTitle:@"Green" destructiveButtonTitle:@"Blue"  
    otherButtonTitles:@"Red",@"Yellow", nil];  
[actionsheet showInView:self.view];
```

委托给自己

```
-(void)actionSheet:(UIActionSheet *)actionSheet clickedButtonAtIndex:(NSInteger)  
    buttonIndex  
{  
    if ([[actionSheet buttonTitleAtIndex:buttonIndex] isEqualToString:@"Red"]) {  
        self.view.backgroundColor=[UIColor redColor];  
    }else if ([[actionSheet buttonTitleAtIndex:buttonIndex] isEqualToString:@"Blue"]){  
        self.view.backgroundColor=[UIColor blueColor];  
    }else if ([[actionSheet buttonTitleAtIndex:buttonIndex] isEqualToString:@"Yellow"]  
    ){  
        self.view.backgroundColor=[UIColor yellowColor];}  
    else  
    {  
        self.view.backgroundColor=[UIColor greenColor];  
    }  
}
```

委托协议

```
@interface ViewController:UIViewController<UIActionSheetDelegate>  
  
@end
```

