iOS Lab for Devs

Developing Apps for iPhone and iPad

Lesson 3

Topics of the day

Mattina

- Animations
- Notifications
- Third-party components
- MapKit

Pomeriggio

- Gestures
- Categories
- Modern Objective-C

Animations

Animazioni

Ogni vista ha le seguenti proprietà che possono essere animate:

@property frame
@property bounds
@property center
@property transform
@property alpha
@property backgroundColor
@property contentStretch

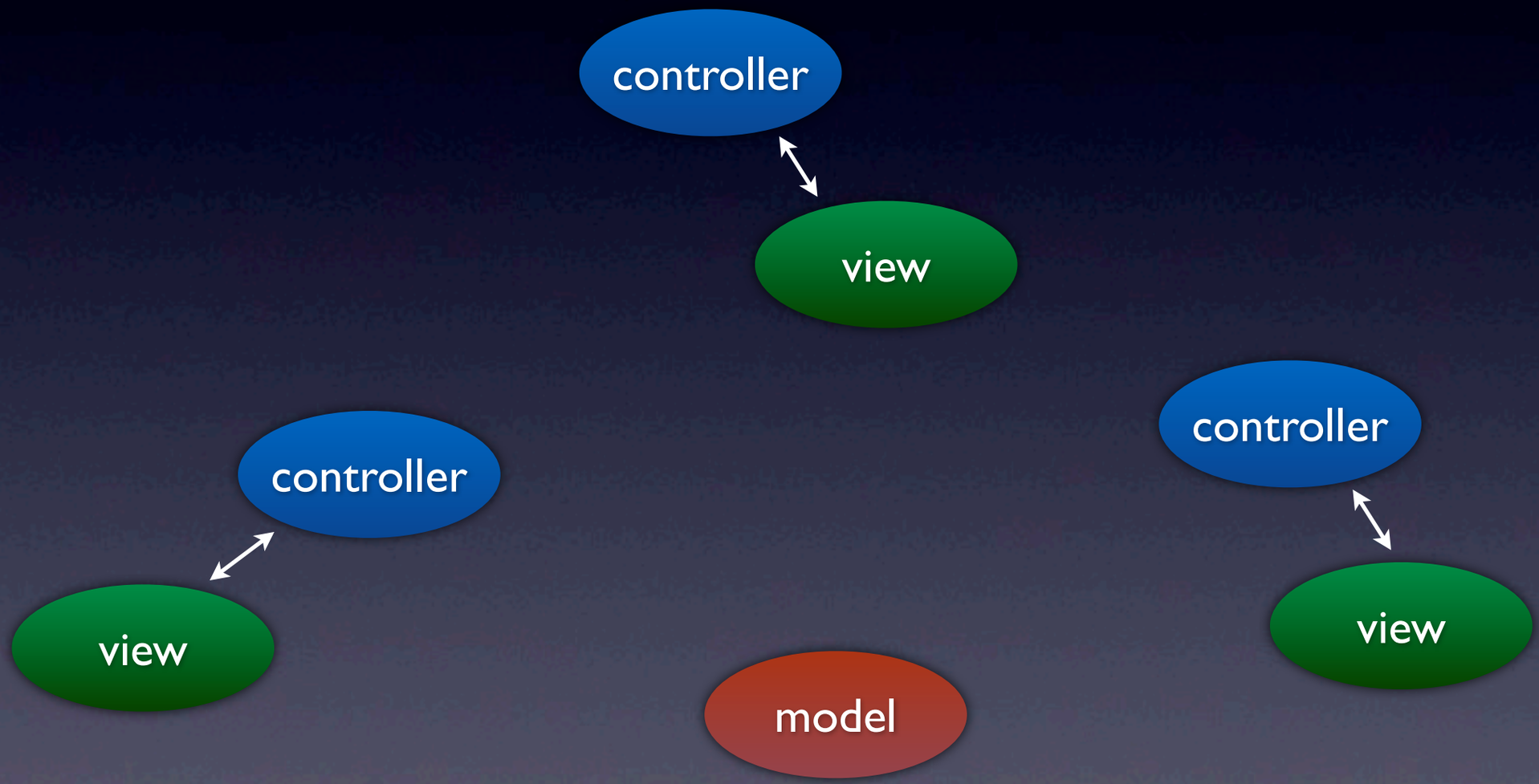
Animazioni

Le viste si possono animare in due modi.

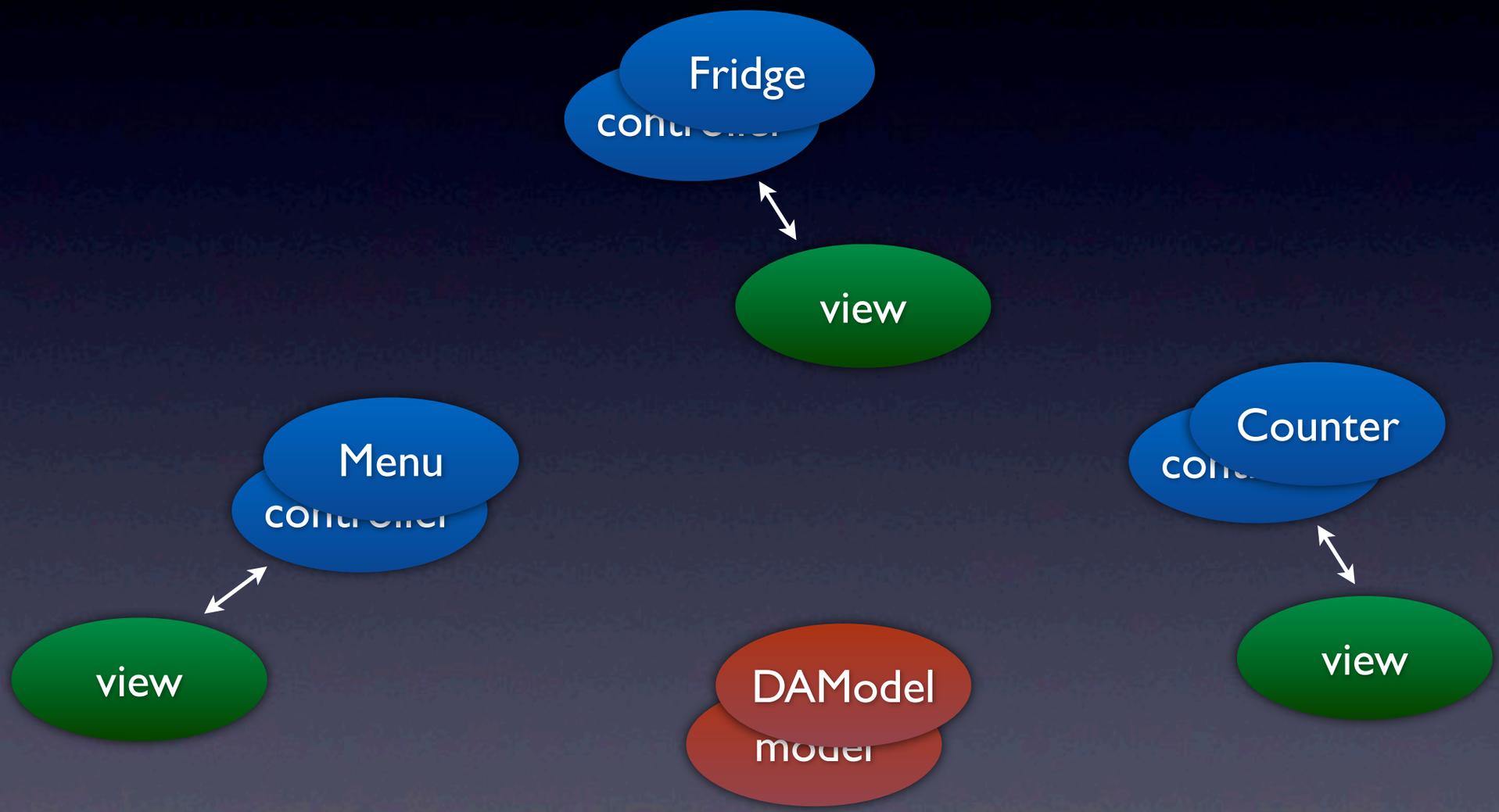
```
1.
[UIView beginAnimations:@"hideButton" context:nil];
...
[UIView commitAnimations];

2.
[UIView
  animateWithDuration:time
                delay:time
              options:UIViewAnimationOptionBeginFromCurrentState
           animations:^{
                         ...
                      }
           completion:^(BOOL finished) {
                         ...
                      }
];
```

Notifications

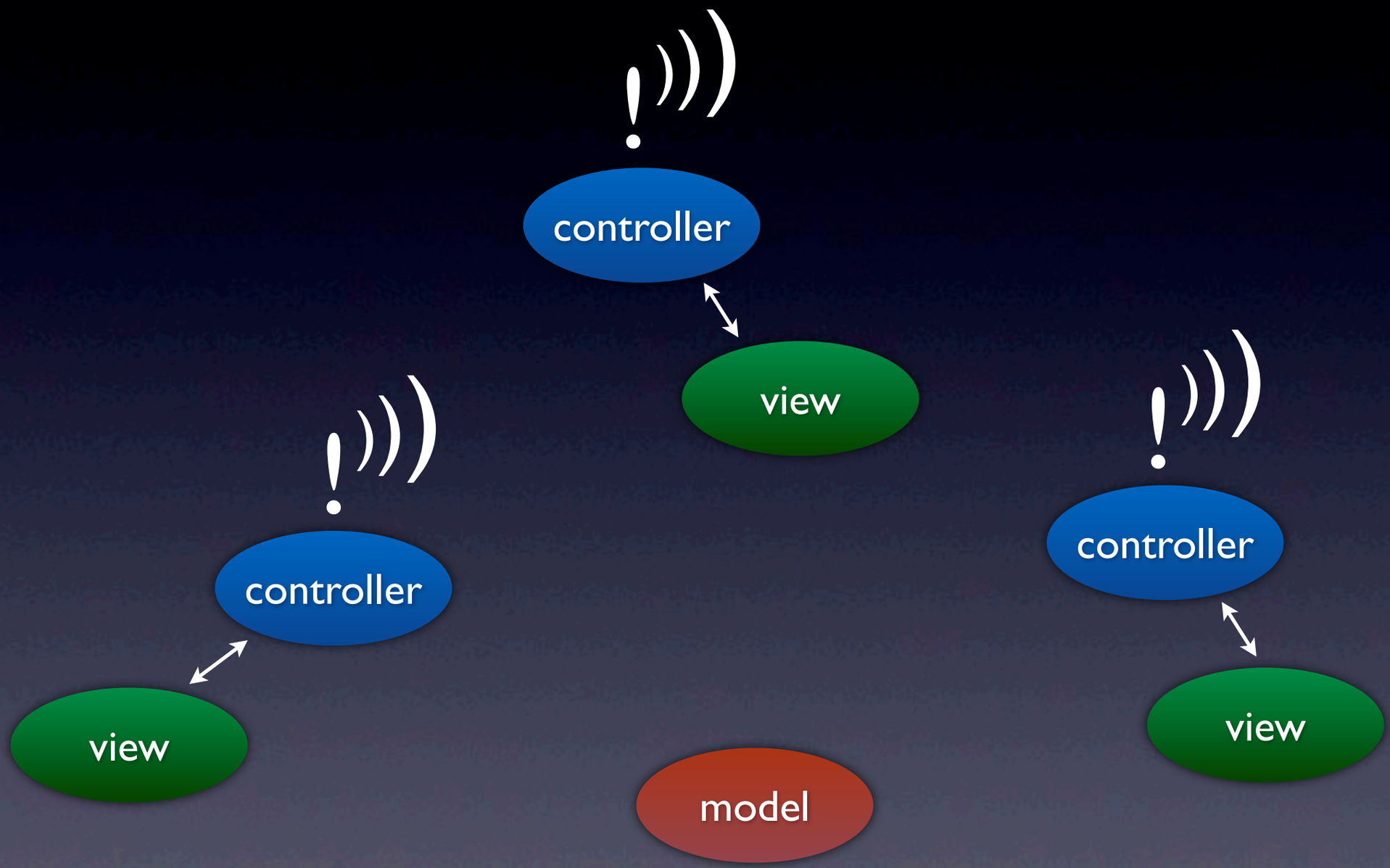
Notifications



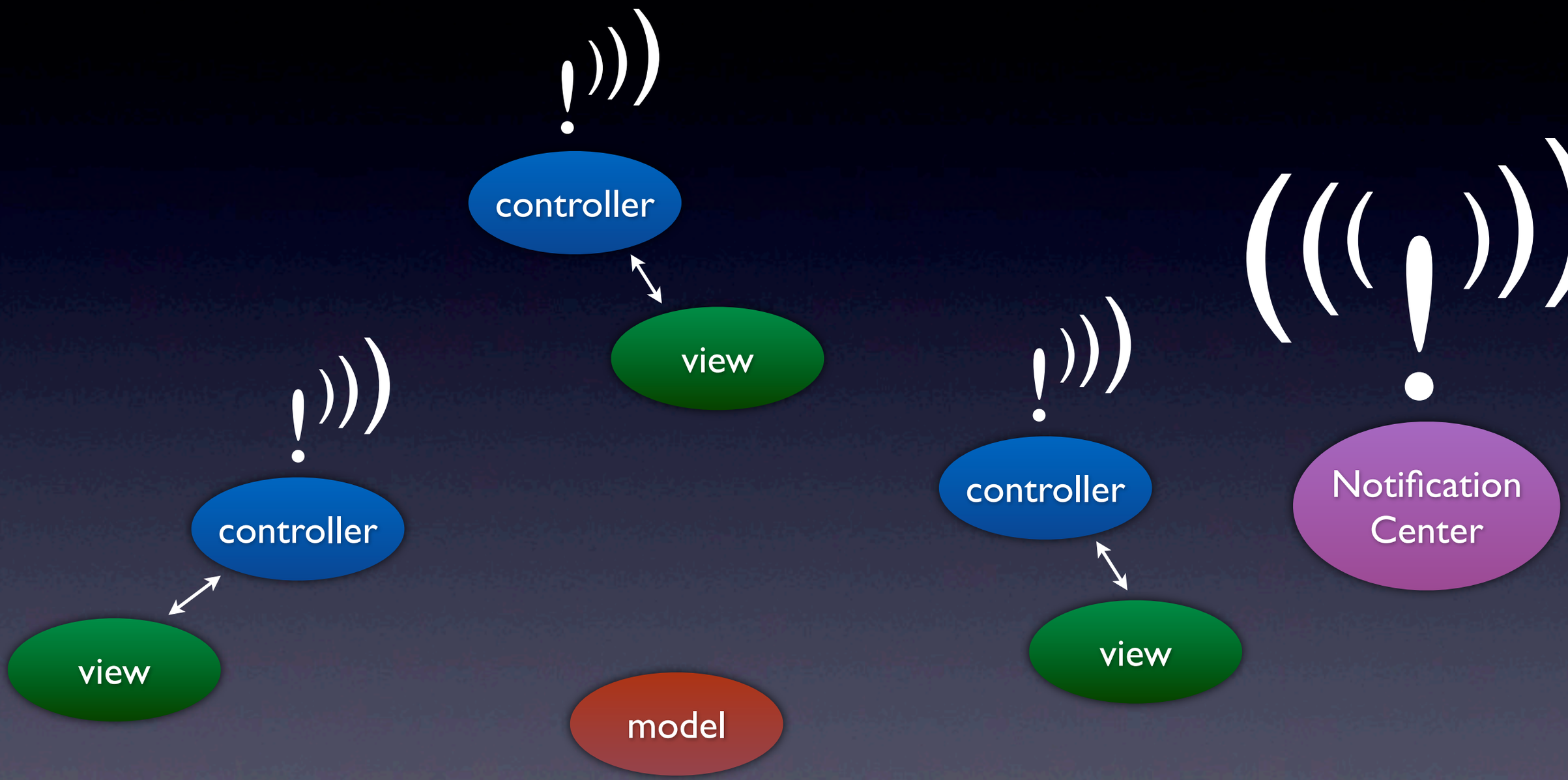
Notifications



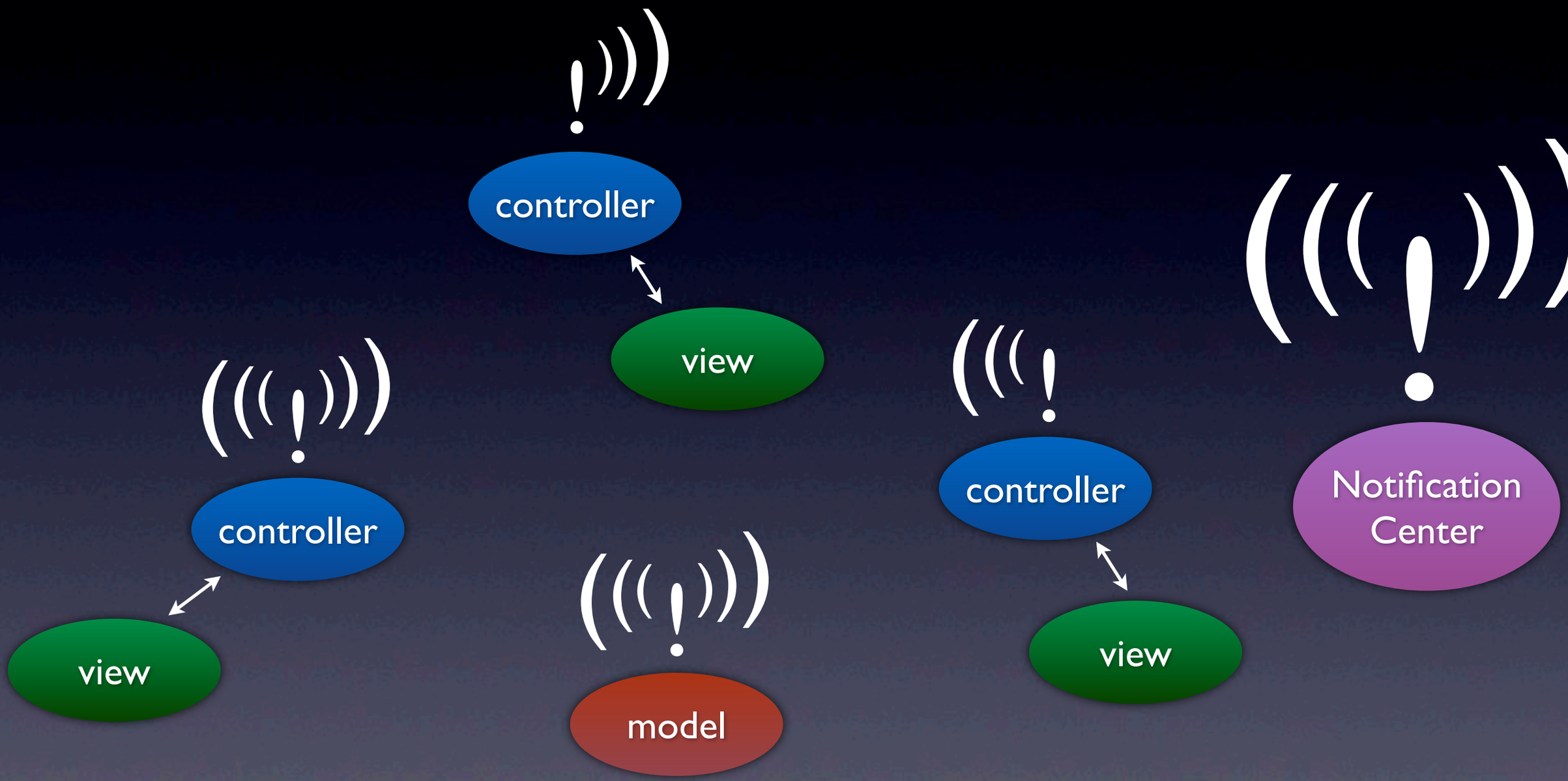
Notifications



Notifications



NotificationCenter



Concetti chiave

NSNotificationCenter

- abbonarsi alle notifiche con uno specifico nome, o di uno specifico oggetto
- inviare le notifiche con uno specifico nome, o allegando uno specifico oggetto. Opzionale: aggiungere un dizionario con altre informazioni utili per chi riceve la notifica

Ricordarsi di dis-abbonarsi nel dealloc!

(il notification center contiene una reference non retained degli abbonati)

Demo

- Balance: quando compro una pizza, voglio notificare la vista counter che ci sono dei soldi in più (e deve fare refresh!)

- Bottone buy: quando modifico la disponibilità di un ingrediente, voglio avvisare le viste che probabilmente la pizza non è ordinabile

Third-Party components

iOS Components

Moltissime sono le componenti open source di terze parti presenti online.
Il portale di riferimento è GitHub.com (non soltanto per iOS).

- **AFNetworking**: libreria complessa per il networking (https://github.com/AFNetworking/AFNetworking);
- **ShareKit**: libreria di facile utilizzo per condivisione contenuti sui social networks (https://github.com/ShareKit/ShareKit);
- **MBProgressHUD/SVProgressHUD**: vista heads up display mancante in Cocoa (https://github.com/jdg/MBProgressHUD, https://github.com/samvermette/SVProgressHUD);
- **SBJSON**: libreria per il parsing di contenuti JSON (https://github.com/stig/json-framework);
- **MGSplitViewController**: custom split view controller che può essere usato in tabbar su iPad (https://github.com/mattgemmell/MGSplitViewController)
- **ConciseKit**: insieme di macro e utility methods per scrivere codice meno verboso (https://github.com/petejkim/ConciseKit);
- **JASidePanels**: view controller stile applicazione Facebook (https://github.com/gotosleep/JASidePanels);

CocoaPods

http://cocoapods.org/

“The best way to manage library dependencies in Objective-C projects.”

Imitazione del GemFile di Ruby On Rails.

PodFile con specifiche di librerie da utilizzare.

Componenti automaticamente aggiunte al Workspace

```
$ [sudo] gem install cocoapods
$ pod setup

$ edit Podfile
platform :ios
pod 'JSONKit',        '~> 1.4'
pod 'Reachability',   '~> 3.0.0'

$ pod install
```

MapKit

CoreLocation e MapKit

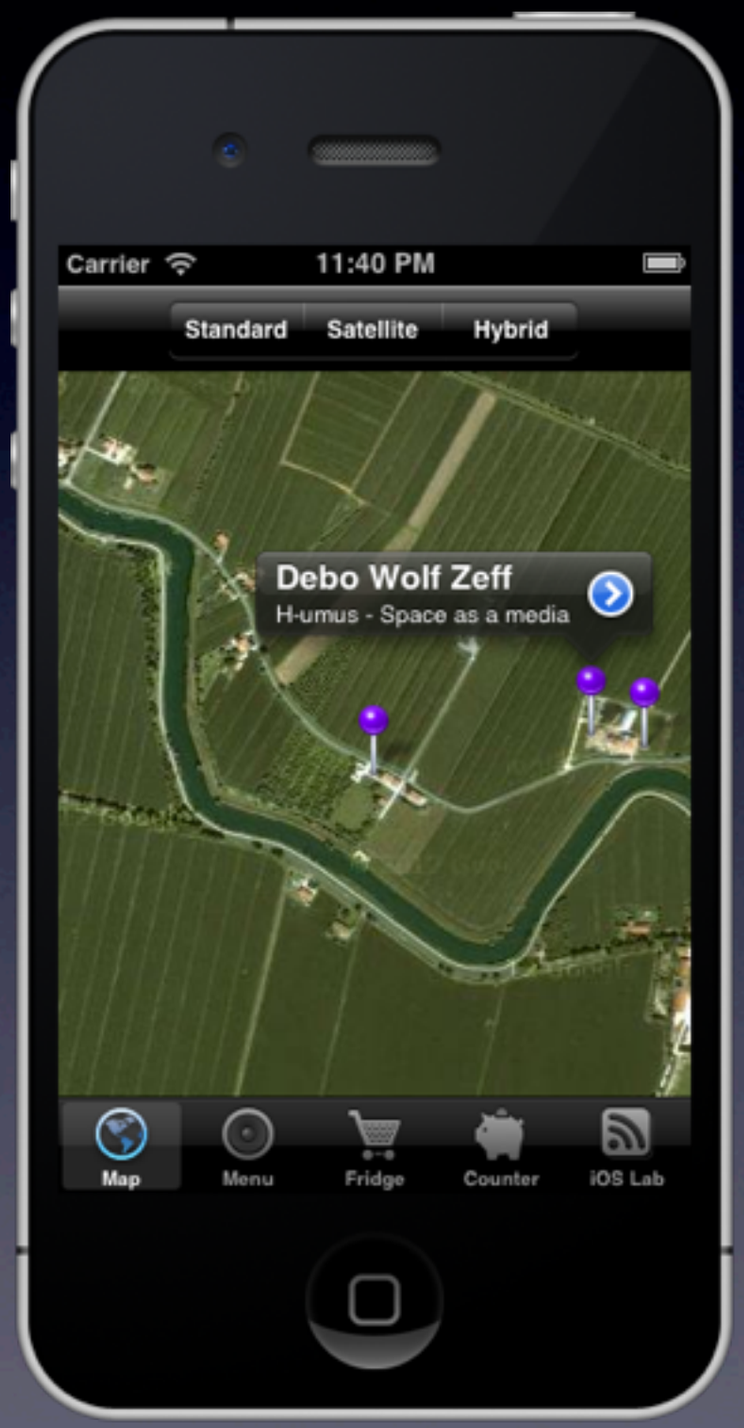
CoreLocation: GPS, altitudine, velocità istantanea... Purtroppo non si può provare sul simulatore

MapKit: mostra mappe di Google. Il delegato è informato di ciò che fa l'utente (zoom, panning, ...)

MapKit mostra un array di id<MKAnnotation>. Protocollo:

```
@property (nonatomic, readonly) CLLocationCoordinate2D coordinate
@property (nonatomic, readonly, copy) NSString *title
@property (nonatomic, readonly, copy) NSString *subtitle
```

MapKit Demo


Carrier
11:40 PM
Standard
Satellite
Hybrid
Debo Wolf Zeff
H-umus - Space as a media
Map
Menu
Fridge
Counter
iOS Lab

Gestures

UIGestureRecognizer

Le UIGestureRecognizer forniscono la possibilità di aggiungere il riconoscimento di alcune gesture su qualsiasi UIView.
E' possibile creare dei nuovi UIGestureRecognizer (tramite subclassing), ma di default esistono questi:

UIGestureRecognizer
UITapGestureRecognizer
UISwipeGestureRecognizer
UIPinchGestureRecognizer
UIRotationGestureRecognizer
UIPanGestureRecognizer
UILongPressGestureRecognizer

UIGestureRecognizer

La sintassi è la seguente:

```
UITapGestureRecognizer *tap = [[[UITapGestureRecognizer alloc]
                                initWithTarget:self
                                        action:@selector(method:)]
                                               autorelease];
[tap setNumberOfTouchesRequired:1];
[tap setNumberOfTapsRequired:1];
[myView addGestureRecognizer:tap];
```

Signature del selettore:

```
- (void)method:(UIGestureRecognizer *)recognizer;
```

La view alla quale viene aggiunta la gesture farà retain della gesture stessa e si occuperà di rilasciarla nella propria fase di dealloc (automatico).

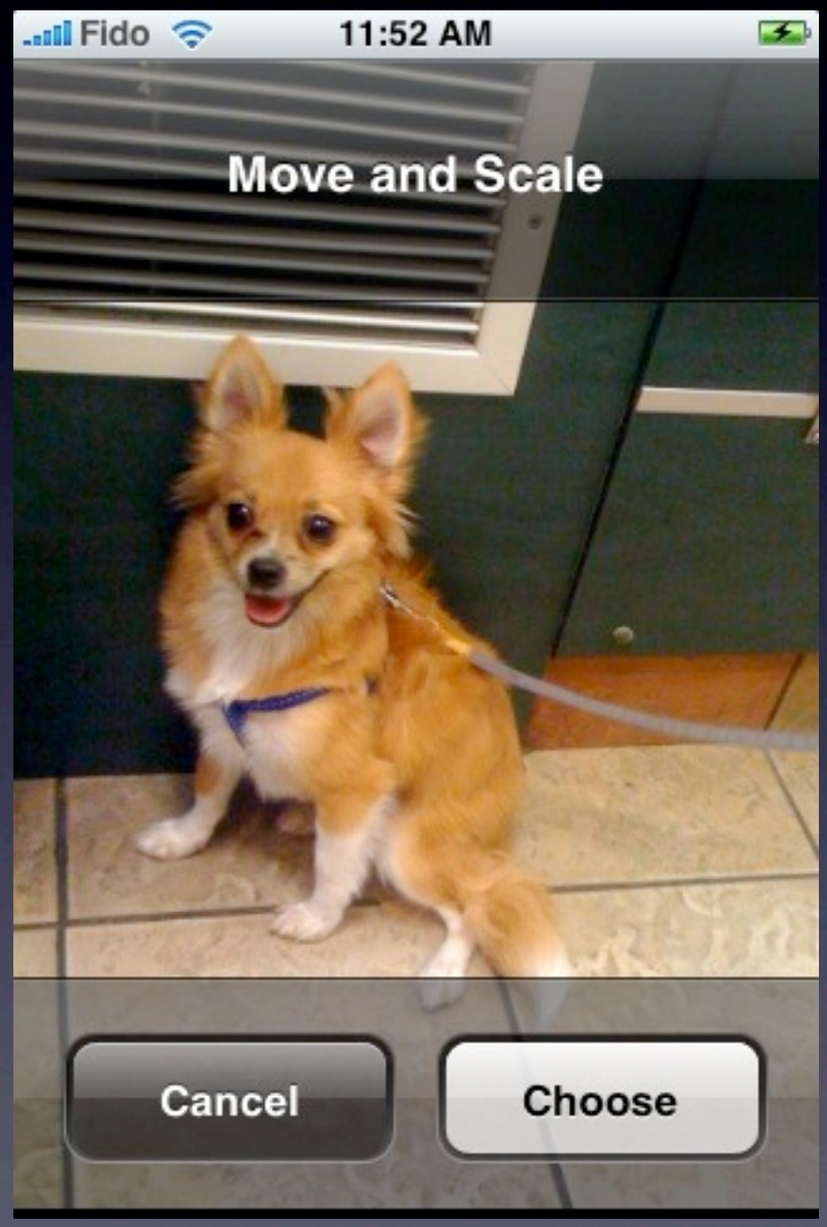
UIGestureRecognizer

Il protocollo UIGestureRecognizerDelegate definisce questi metodi (opzionali):

```
- (BOOL)gestureRecognizerShouldBegin:(UIGestureRecognizer
*)gestureRecognizer;

- (BOOL)gestureRecognizer:(UIGestureRecognizer *)gestureRecognizer
shouldRecognizeSimultaneouslyWithGestureRecognizer:
(UIGestureRecognizer *)otherGestureRecognizer;

- (BOOL)gestureRecognizer:(UIGestureRecognizer *)gestureRecognizer
shouldReceiveTouch:(UITouch *)touch;
```

UIImagePickerController

L'UIImagePickerController è un view controller che facilita l'accesso alle immagini salvate nella galleria del device e alla fotocamera.
Fornisce un'interfaccia predefinita da Apple e un protocollo UIImagePickerControllerDelegate.
E' presentabile come un qualsiasi altro UIViewController e bisogna configurarne la proprietà sourceType.



Categories and Class Extension

Objective-C Categories

Concettualmente simile al paradigma della ‘class extension’.
Permette di ‘aggiungere’ metodi a classi esistenti. Esempio:

```
//  NSArray+Additions.h

@interface NSArray (Additions)

- (void)shuffle;

@end

// NSArray+Additions.m

@implementation NSArray (Additions)

- (void)shuffle
{
    for (NSInteger i = [self count] - 1; i > 0; i--) {
        [self exchangeObjectAtIndex: arc4random() % (i+1) withObjectAtIndex:i];
    }
}
```

alcuni esempi: https://github.com/albertodebortoli/ADBCategories

Objective-C
Class Extension

Solo con compilatore CLANG/LLVM 2.0 (default in Xcode >= 4.3).
L'implementazione dei metodi deve essere fatta nel blocco di implementazione principale.
L'uso più comune è quello di dichiarare property in .h con accesso readonly, mentre in .m con accesso readwrite.

Esempio:

```
//  BankAccount.m

@interface BankAccount ()

NSString *_IBAN;

- (NSInteger)numberOfTransfers;

@property (nonatomic, readwrite) NSString *IBAN;

@end
```

```
//  BankAccount.m

@implementation BankAccount

@synthesize IBAN = _IBAN;

- (NSInteger) numberOfTransfers
{
    ...
}

@end
```

Modern Objective-C

Modern Objective-C

- Default synthesis of @property instance variables and accessor methods
- Instance variables in class extensions
- Instance variables in @implementation block
- NSNumber, NSDictionary and NSArray literals
- @YES and @NO literals
- NSDictionary and NSArray subscripting

Modern Objective-C

- Default synthesis of @property instance variables and accessor methods

OLD

```
// interface
NSObject *_myObject;
@property (nonatomic) NSObject *myObject;

//implementation
@synthesize myObject = _myObject;
```

NEW

```
// interface
NSObject *_myObject;
@property (nonatomic) NSObject *myObject;
```

OR

```
// interface
@property (nonatomic) NSObject *myObject;
```

Modern Objective-C

- Instance variables in @implementation block

```
@implementation Thing
{
    NSString  *title;
    float  radius;
    id  delegate;
}
```

Modern Objective-C

- NSNumber, NSDictionary, NSArray, @YES and @NO literals

```
NSNumber *aNumber = [NSNumber numberWithFloat:2.3];

NSNumber *anotherNumber = [NSNumber
numberWithFloat:x];

NSArray *anArray = [NSArray arrayWithObjects:aThing,
@"A String", [NSNumber numberWithFloat:3.14], nil];

NSDictionary *aDictionary = [NSDictionary
dictionaryWithObjectsAndKeys:value, @"Key", [NSNumber
numberWithBOOL:YES], @"OtherKey", nil];
```

Modern Objective-C

- NSNumber, NSDictionary, NSArray, @YES and @NO literals

```
NSNumber *aNumber = @2.3f;

NSNumber *anotherNumber = @(x);

NSArray *anArray = @[ aThing, @"A String", @3.14 ];

NSDictionary *aDictionary = @{ @"Key" : value,
@"OtherKey" : @YES };
```

Modern Objective-C

- NSDictionary and NSArray subscripting

```
NSString *value1 = [dict objectForKey: @"Key"];

NSString *value1 = dict[@"Key"];

NSString *value2 = [array objectAtIndex:2];

NSString *value2 = array[2];
```

Modern Objective-C