## 4-Controllers of View Controllers

- Multiple MVCs
- Aggiungere un view controller
  - UIViewController
  - <u>UINavigationController</u>
  - UINavigationBar
  - Segues
  - UITabBarController
    - Le viste di un UITabBarController

Applicazioni iOS complesse tipicamente utilizzano multiple viste per mostrare i content. Per un istanza singola potrebbe non essere possibile mostrare tutti i contenuti all'interno di una singola vista, quindi un altra vista è necessaria per mostrare content aggiuntivi.

#### Nota

Più viste rendono il flusso dell'applicazione più naturale ed intuitiva per l'utente.

La gestione di più viste significa avere più view-controllers che devono coordinare il flusso dell'applicazione.

### Ricorda

Ogni vista è controllata da il suo controller, quindi per un applicazione sono presenti MVCs multipli.

# Multiple MVCs

Come per la programmazione OOP anche i view controller devono essere altamente specializzati ed indipendenti uno dall'altro. Ognuno dovrebbe essere responsabile di gestire una vista che mostra qualche semplice contenuto.

#### Nota

Alcune applicazioni Apple presentano MVCs multipli, queste sono:

- App Contatti
- App Calendario
- App Musica

Differenti view controller possono essere rappresentati in modi differenti, in base a cosa il contenuto deve mostrare. La gestione delle transazioni tra differenti MVCs è fornita da dei controllers dedicati:

- UINavigationController
- UITabBarController

Le transazioni tra MVC sono chiamate segues, e sono attivate dai controlli contenuti in una vista o da determinati eventi che possono avvenire nell'applicazione (dipende dalla logica di quest'ultima).

# Aggiungere un view controller

Al rio view controllore sono aggiunti nella storyboard in 3 modi:

- 1. trascinando un (UIViewController) dal palette degli oggetti alla storybard.
- 2. Creando una sotto classe di UIViewController utilizzando il menu di creazione di un nuovo file.
- ${\tt 3.} \ \ {\tt Nell'identity inspector impostando una classe di } \ \overline{\tt UIViewController} \ \ a \ \ nuova \ \ sotto-classe \ \ creata.$

### **UIViewController**

La classe  $\overline{\mathtt{UIViewController}}$  è una classe che implementa uno view controller specializzato che gestisce la navigazione gerarchica di contenuti. L'utilizzo tipico è drill-down to detailed content (year  $\rightarrow$  month  $\rightarrow$  day).

Il navigation controller gestisce uno stack di view controllers, la classe che implementa il navigation controller fornisce metodi per la gestione dello stack a tempo di esecuzione

Metodo	Descrizione
((void)pushViewController:(UIViewController *)viewController animated:(BOOL)animated	È utilizzato per mettere una lista in cima allo stack

Metodo	Descrizione
((UIViewController *)popViewControllerAnimated:(BOOL)animated	È utilizzato per rimuovere una vista dalla cima dello stack
((NSArray *)popToViewController:(UIViewController *)viewController animated:(BOOL)animated)	Rimuove dalla cima dello stack tutti view controllers finché non viene trovato quello desiderato
<pre>((NSArray *)popToRootViewControllerAnimated:(BOOL)animated)</pre>	Rimuove tutti i view controller dallo stack escludendo il view controller root

#### Nota

Tutti gli (UIViewController) presentano la proprietà (navigationController) che può essere usata per accedere al (UINavigationController) che la incorpora.

## **UINavigationController**

La vista per un navigation controller è solo un container per diverse altre viste. Alcuni esempi di navigation controller sono:

- Barra di navigazione
- Barra degli strumenti
- Vista con contenuti personalizzati

Quando ci si muove da un MVC ad un altro, il contenuto della vista personalizzata cambia, come cambierà anche la barra di navigazione e la barra degli strumenti.

### UINavigationBar

La (UINavigationBar) mostra:

- " Un titolo, per il view controller corrente. Può essere imposta to mediante la proprietà (title del (UIViewController) corrispondente
- Un bottone per tornare indietro (back button), che mostra il (title) del (UIViewController) precedente nel navigation stack
- Un array di oggetti (UIBarButtonItem) accessibile con la proprietà (navigationItem.rightBarButtonItems) del (UIViewController)

### Nota

Un UINavigationControlle inizia mostrando nessun contenuto, poi quando la proprietà rootViewController è impostata il UINavigationController mostrrà le viste gestite dal view controller.

### Segues

Le segues sono responsabili di svolgere la transazione (attivate mediante delle segue o mediante codice) visiva tra 2 view controllers.

Tipi di segues
Push
Modal
Custom

Un view controller può svolgere azioni aggiuntive prima di performare una segue, che necessita il tempo di essere preparata, ovvero necessita di passare i dati importanti alla nuovo view controller.

Ogni view controller può sovrascrivere il metodo (void)prepareForSegue:(UIStoryboardSegue \*)segue sender:(id)sender per prepararsi per la segue. Il metodo è chiamato prima he il nuovo view controller (destinationViewController) della segue) è presentato.

Esempio preparazione per una segue

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
        if([segue.identifier isEqualToString:@"GoToSecondView"]) {
            if([segue.destinationViewController isKindOfClass:[SecondViewController
class]]) {
            SecondViewController *sVC =
            (SecondViewController*)segue.destinationViewController; sVC.data = @"Some data";
            }
        }
    }
}
```

Instanziazione di un view controller programmaticamente

Un view controller può essere instanzializzato a mano (questo può avvenire quando il tipo di destinazione del view controller viene risolto a tempo di esecuzione).

```
NSString *vc = @"SecondViewController"; UIViewController *controller = [self.storyboard
```

### instantiateViewControllerWithIdentifier:vc];

Il view controller, dopo essere stato creato, viene inserito in cima al navigation stack con il metodo [pushViewController:animated:]

### **UITabBarController**

La classe UITabBarController che implementa un view controller specializzato che gestisce una selezioni di interfacce radio-style. Un utilizzo tipico è quello delle multiple sections (es. App Orologio).

Le schede sono mostrate sul fondo della finestra e permettono di scegliere tra differenti modalità.

L'interfaccia di ogni scheda di una tab bar controller è associata a un view controller personalizzato, quando un utente seleziona una scheda specifica, la barra delle schede mostra la root view del corrispondente view controller, rimpiazzando ogni view precedente. I view controller sono assegnati sul tab bar controller impostando la proprietà (viewControllers) del (UITabBarController).

### Le viste di un UITabBarController

I view controller incorporati definiscono l'aspetto della loro tab bar item corrispondente. Il titolo per la scheda che correntemente mostrata il proprio view controller può essere impostata utilizzando la proprietà title del (UITabController) incorporato.

### Nota

L'aspetto del titolo e dell'icona delle shceda della barra pososno essere configurate nello storyboard. In alternativa è possibile farlo programmaticamente creando un istanza di (UITabBarItem) per configurare imamgine e titolo, ed impostare la proprietà (tabBarItem) del controller incorporato con l'oggetto creato.

#### Nota

Lo spazio dedicato al tab bar controller è limitato, quindi quando sono presenti più di 6 view controllers viene aggiunta in automatico la scheda "morte"