



# Въведение в UIKit





# За какво ще си говорим

- Събития: обобщение
- Класът UIControl
- Основни елементи за потребителския интерфейс
- Специални видове изгледи и приложенията им



# Събития в iOS

- Събития и контроли
- Няколко основни механизма:
  - събития на ниско ниво - **обработка на потребителски докосвания**
  - събития на високо ниво
    - **actions**
    - **gesture recognizers**
  - **делегати**
  - **notifications**
    - събития извън изгледите и контролите
    - много реагиращи
    - ще ги разгледаме в друга лекция



# Модел на събитията

- Работи на ниско ниво
- Основен компонент на събитието: потребителското докосване
- Визуалните елементи реагират на докосванията
- Кой ще реагира, се определят от т.нар. *Responder chain*
- На визуалните елементи се изпращат съобщения, дефинирани в UIResponder

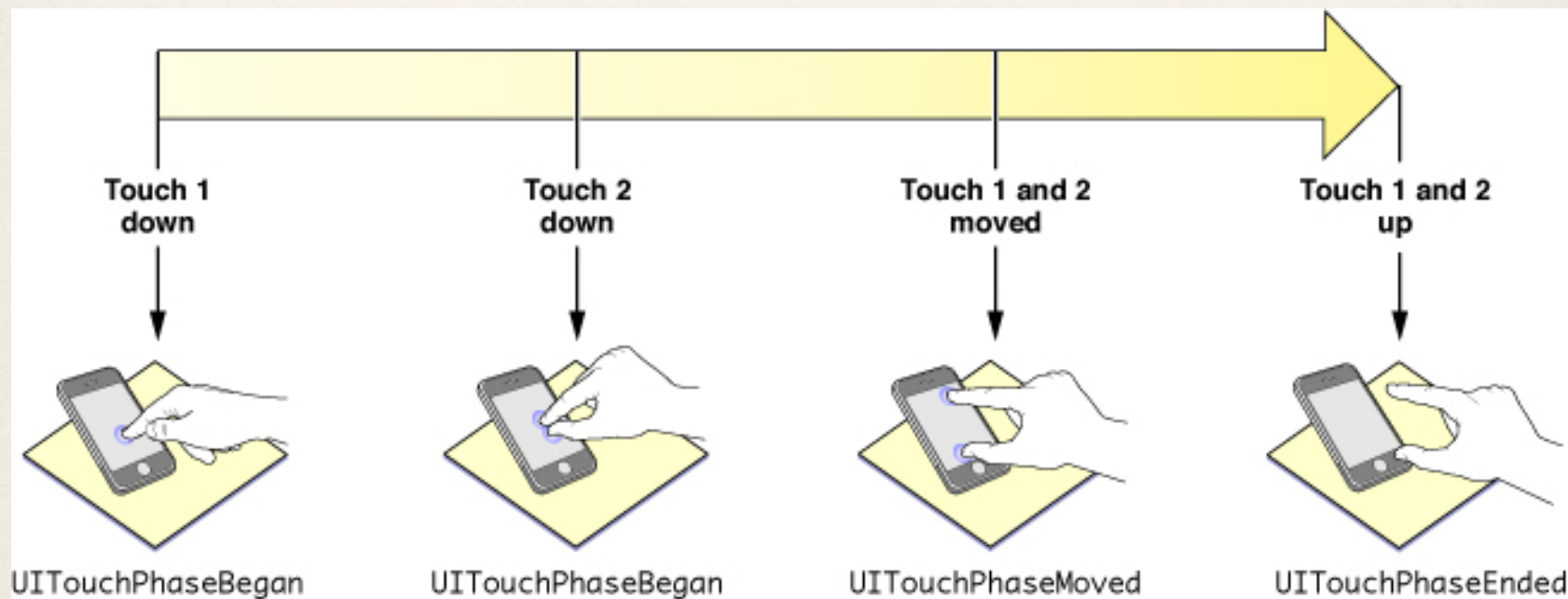




# Обработка на докосвания

- Методите

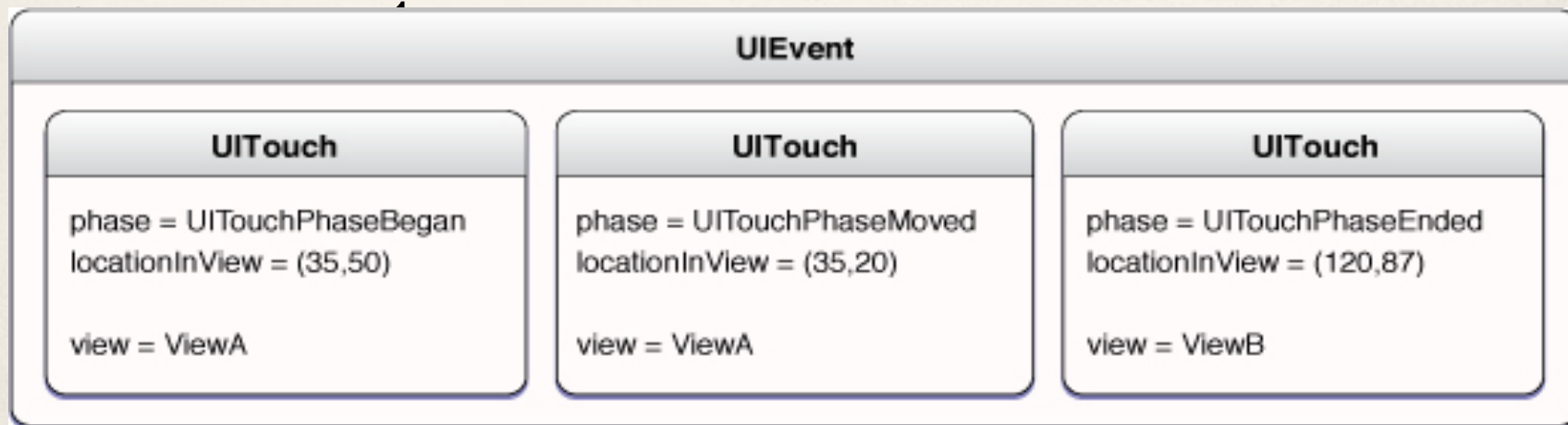
- touchesBegan:(NSSet\* touches) withEvent:(UIEvent\*)
- touchesMoved, touchesEnded, touchesCanceled



# UITouch & UIEvent



- UITouch
  - представя "пръст" по екрана
  - полезна информация за: кога, къде, колко
  - **locationInView:** - координати в произволен изглед
- UIEvent
  - съдържа множество докосвания



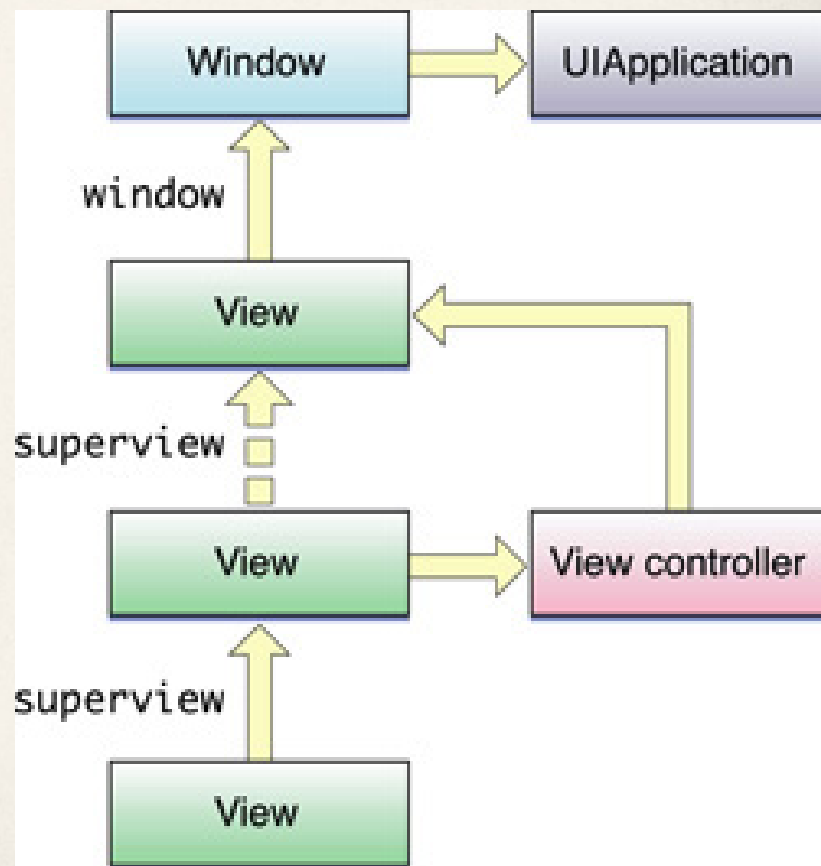
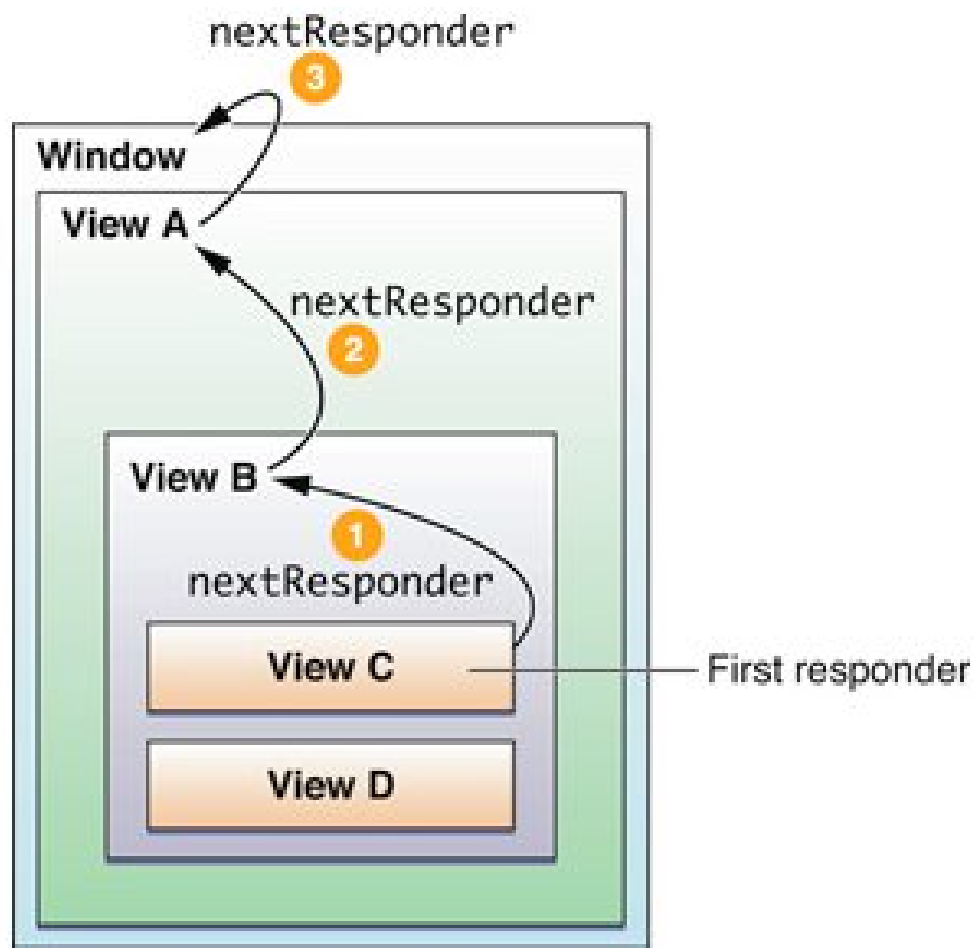


# Responder chain

- Всички изгледи са подредени в йерархия
- Един от тях е ***first responder*** - по подразбиране този, който е най-отгоре на мястото на докосването в key window
- Ако той не реагира, събитието се предава на неговия родител в йерархията
- Ако никой, включително най-горният UIWindow не реагира, UIApplication поема събитието



# Картинка







# Особености

- Някои докосвания и събития имат конкретна семантика (*пример*: избирам ред от таблица)
- Изгледите често делегират обработката на такива събития на свой **делегат** (обикновено техния `UIViewController`)
- Събитието се предава нагоре ако няма обработка на докосването и няма деклариран делегат
- Напомняме: делегатите имплементират съобщения-кукички



# Още особености

- Сложните жестове от много докосвания се разпознават с приоритет от Gesture Recognizers
- `resignFirstResponder/becomeFirstResponder`  
**не са** каквото си мислим
- Можем да променим началото на веригата с ***hitTest*** и ***pointInside***
- Можем да изключим докосванията
  - за изглед - `userInteractionEnabled`
  - за целия application - `beginIgnoringInteractionEvents`
- И още много



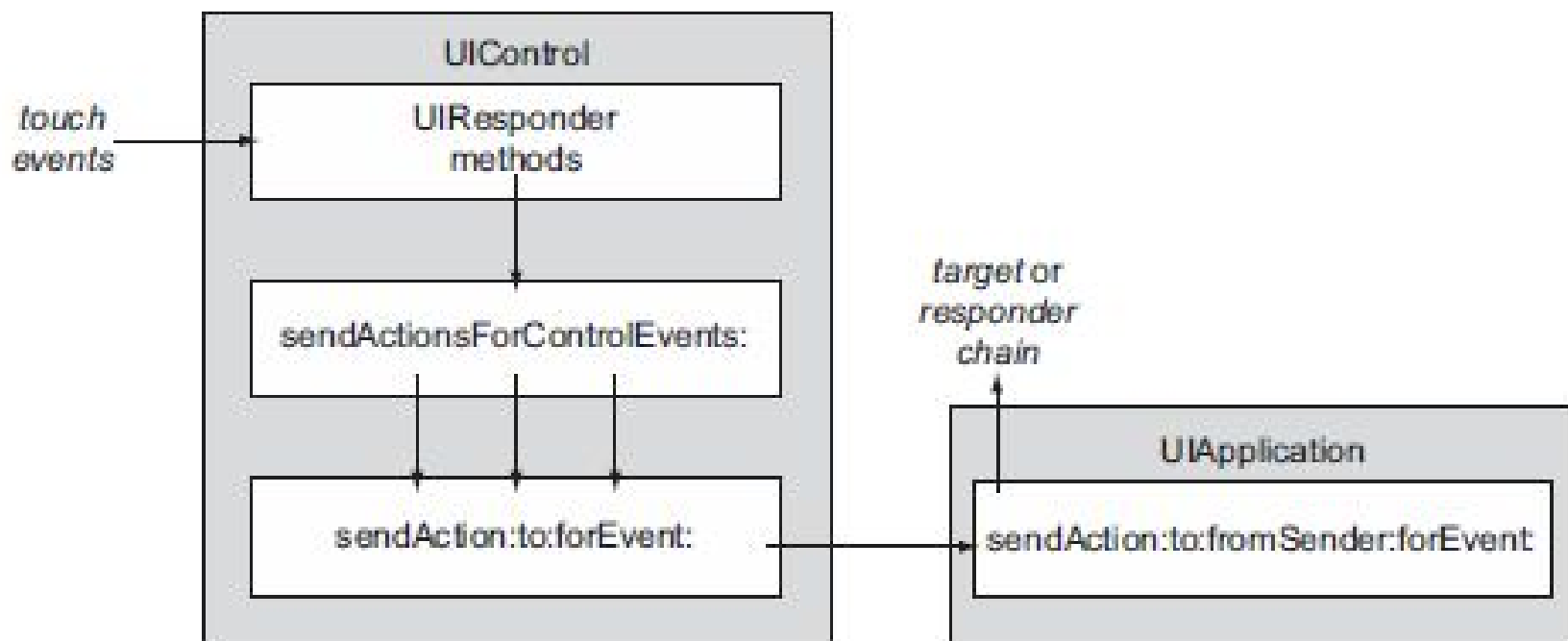
# Actions & Targets

- Контролите предлагат по-лесен начин за реакция
- Обработват докосванията вместо нас и им придават семантика, създавайки *control events* (пример: touch => button push)
- Ние указваме от кое контролно събитие, какъв *action* следва
- ***action = send a message to a target***
- ***Важно:*** UIControl не предава събития по responder chain



# Картинка

- Без actions ние трябва да си пишем цялата лява част на картинката и да разпознаваме докосванията като събитие







# Задаване на actions

- Програматично

- ако създаваме контролата програматично
- ако искаме да променяме поведението runtime
- става с метода:

addTarget: **обработчик**  
action:@selector( **метод на обработчика** ,  
forControlEvents: **кои събития**

- Чрез Interface Builder

- IBAction = void



# Класът UIControl

- Е наследник на UIView
- Всички контроли го наследяват (почти)
- Добавя се като subview
- Дефинира задаването на actions
- И следните свойства
  - enabled
  - highlighted
  - selected
  - state
- Ако искаме все пак да следим докосвания: *beginTrackingWithTouch:withEvent:*



# На фокус: UITextField

- Едноредово текстово поле
- Реагира на touch actions и editing actions
- Разни свойства:
  - NSString\* text
  - NSString\* placeholder
  - UITextFieldViewMode clearButtonMode
  - всякакви други стилизираци свойства
- Вметка
  - [UIColor whiteColor]
  - [UIFont fontWithName:size:]



# UITextField: клавиатурата

- За работа с клавиатурата се грижи протоколът `UITextFieldInputTraits`
  - `UIKeyboardType` **keyboardType** (напр. `UIKeyboardTypeEmailAddress`)
  - `BOOL` **secureTextEntry**
  - `UIReturnKeyType` **returnKeyType**
  - и други
- Връзката с `resignFirstResponder` / `becomeFirstResponder`





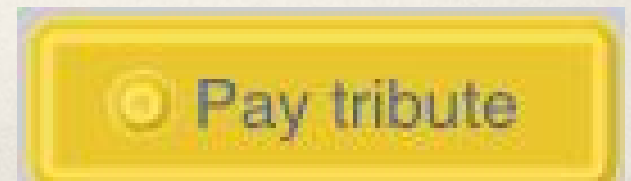
# UITextFieldDelegate

- Вметка английска филология: will, did, should
- **textFieldShouldReturn** и работа с клавиатурата
- **textFieldShouldBeginEditing & Co. textFieldDidBeginEditing** и кой има приоритет между делегат и action



# UIButton

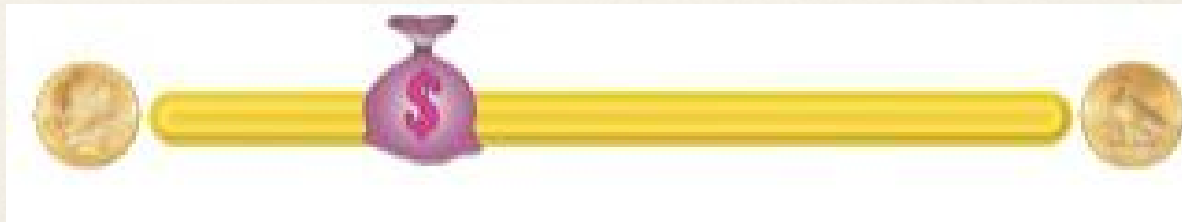
- Създава се с factory метод `buttonWithType`
  - свойството `buttonType`
  - не трябва да наследяваме (class cluster)
- Стилизация и скининг
  - състояния
  - ***`setTitle:(NSString*) forState:(UIControlState)`***
- Actions - всички touch actions  
**& UILabel**
  
- Статичен текст
- `adjustsFontSizeToFitWidth`
- `lineBreakMode`





# UISlider & UIPageControl

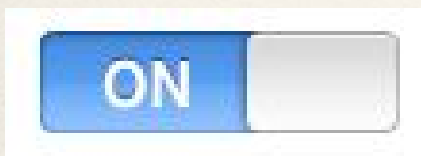
- Actions, свързани с промяна на стойност
- UISlider
  - value, minimumValue, maximumValue
  - continuous
- UIPageControl
  - numberOfPages
  - currentPage





# UISwitch & UISegmentedControl

- Отговарят на checkbox & radio buttons
- Промяната им може да се анимира
  - `setOn:YES animated:YES`



- UISegmentedControl
  - `selectedSegmentIndex`







# UIImage & UIImageView

- Интерлюдия: ресурси
  - достъпни директории
  - NSBundle
  - методи за откриване на ресурси
- UIImage
  - imageNamed и кеширането
- UIImageView



# UIScrollView

- Изглед, по-голям от екрана, който позволява обработка на скалиране
- Основни свойства
  - `contentSize`, `contentOffset`, `bounces`
  - `showHorizontalScrollIndicator`
- Работа с делегата
  - `scrollViewDidScroll:` `scrollViewDidEndDragging:`  
`willDecelerate:`
- Мащабиране
  - `viewForZoomingInScrollView:`



# UIAlertView

- Показва модално съобщение на екрана

```
UIAlertView* alert = [[UIAlertView alloc] initWithTitle:@"Not So Fast!"  
message:@"Do you really want to do this tremendously destructive thing?"  
delegate:self cancelButtonTitle:@"Yes" otherButtonTitles:@"No", @"  
Maybe", nil];  
[alert show];  
[alert release];
```





# UIWebView

- Представява мини-браузър в приложението
- Може да зарежда:

- HTML низ

***loadHTMLString:baseURL:***

- данни с произволен MIME тип

***loadData:MIMETYPE:textEncodingName:baseURL:***

- интернет/локален ресурс

***loadRequest:***

- NSData & NSURL & NSURLRequest





# UIWebViewDelegate

- Позволява да следим какво се зарежда
- Проверка за състояние
  - webViewDidStartLoad:
  - webViewDidFinishLoad:
  - webView:didFailLoadWithError:(**NSError\***)
- Зареждане на линкове
  - webView:shouldStartLoadWithRequest:navigationType:
  - вторият параметър ни казва какво се е случило



# JavaScript в UIWebView

- ObjectiveC => JavaScript
  - stringByEvaluatingJavaScriptFromString:
- JavaScript => ObjectiveC
  - измисляме си заявка и изпълняваме:  
***document.location='play:me'***
  - в обработката в делегата проверяваме схемата  
***if ([request.URL.scheme isEqualToString: @"play"])***
  - и реагираме



# Други контроли и специални изгледи

- UISearchBar
- UITextView
- UIDatePicker
- UIPickerView
- UIActivityIndicatorView
- UIProgressView
- UIActionSheet
- **UITableView**

