

Разработване на приложения за iOS



Петко Пенчев, TapBits.com

UISegmentedControl

- Представя списък от опции



- `@property UISegmentedControlStyle segmentedControlStyle;`
UISegmentedControlStylePlain/Bordered/Bezeled/Bar

UISegmentedControl

- typedef enum {
 UISegmentedControlStylePlain,
 UISegmentedControlStyleBordered,
 UISegmentedControlStyleBar,
 UISegmentedControlStyleBezeled
} UISegmentedControlStyle;
 - Масив от NSString или UIImage
NSArray *itemArray = [NSArray arrayWithObjects: @"One", @"Two", @"Three", nil];
 - Инициализация
UISegmentedControl *segmentedControl = [[UISegmentedControl alloc]
initWithItems:itemArray];
 - - (void)setImage:(UIImage *)image forSegmentAtIndex:(int)index;
- (NSString *)titleForSegmentAtIndex:(int)index;
 - Избран по подразбиране
@property NSInteger selectedIndex;
- Ако нищо не е избрано UISegmentedControlNoSegment

UISegmentedControl

- Установяване

```
segmentedControl.frame = CGRectMake(35, 200, 250, 50);  
segmentedControl.segmentedControlStyle = UISegmentedControlStylePlain;  
segmentedControl.selectedSegmentIndex = 1;
```

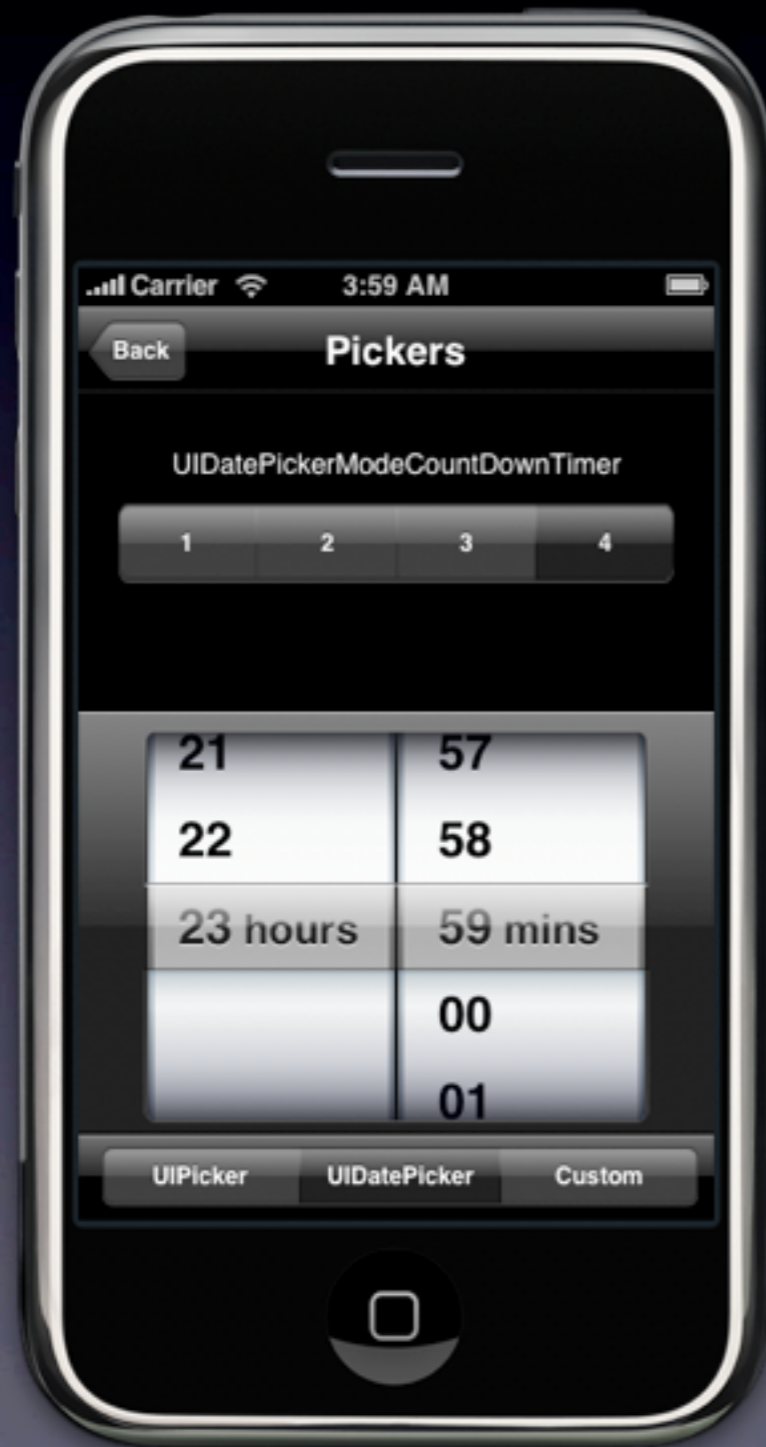
- Target/Action model

```
[segmentedControl addTarget:self action:@selector(pickOne:)  
                        forControlEvents:UIControlEventValueChanged];
```

- Action

```
- (void) pickOne:(id)sender {  
    UISegmentedControl *segmentedControl = (UISegmentedControl *)sender;  
  
    label.text = [segmentedControl  
                titleForSegmentAtIndex: [segmentedControl selectedSegmentIndex]];  
}
```

UIDatePicker



August	13	2007
September	14	2008
October	15	2009
November	16	2010
December	17	2011

`UIDatePickerModeTime` - час, минута и опционално AM/PM

`UIDatePickerModeDate` - месец, ден, година

`UIDatePickerModeDateAndTime` - ден от седмицата, месец, ден, час

`UIDatePickerModeCountDownTimer` - час, минута като интервал от време

UIDatePicker

- Свойства

maximumDate property
minimumDate property
minuteInterval property
countDownDuration property

- Установяване

@property(nonatomic, retain) NSDate *date
@property(nonatomic, retain) NSTimeZone *timeZone
@property(nonatomic, copy) NSCalendar *calendar

- (void)setDate:(NSDate *)date animated:(BOOL)animated

UIDatePicker

```
#import <UIKit/UIKit.h>

@interface DatePicker : UIViewController
    <UIPickerViewDelegate> {
    ...
}
@end

...
#import "DatePicker.h"

@implementation DatePicker
UILabel *label;
UIDatePicker *datePicker;

- (void)changeDateInLabel:(id)sender{
    NSDateFormatter *df = [[NSDateFormatter alloc] init];
    df.dateFormat = NSDateFormatterMediumStyle;
    label.text = [NSString stringWithFormat:@"%s",
        [df stringFromDate:datePicker.date]];
    [df release];
}

- (void)dealloc {
    [datePicker release];
    [label release];
    [super dealloc];
}

- (void)viewDidLoad {
    [super viewDidLoad];

    //Create label
    label = [[UILabel alloc] init];
    label.frame = CGRectMake(10, 10, 300, 40);
    label.textAlignment = NSTextAlignmentCenter;

    NSDateFormatter *df = [[NSDateFormatter alloc] init];
    df.dateFormat = NSDateFormatterMediumStyle;
    label.text = [NSString stringWithFormat:@"%s",
        [df stringFromDate:[NSDate date]]];
    [df release];
    [self.view addSubview:label];
    [label release];

    datePicker = [[UIDatePicker alloc] initWithFrame:
        CGRectMake(0, 250, 325, 250)];
    datePicker.datePickerMode = UIDatePickerModeDate;
    datePicker.hidden = NO;
    datePicker.date = [NSDate date];
    [datePicker addTarget:self
        action:@selector(changeDateInLabel:)
        forControlEvents:UIControlEventValueChanged];
    [self.view addSubview:datePicker];
    [datePicker release];
}
@end
```

Модални известия

- **Action Sheets**

Типично се визуализират като въпрос с няколко възможни отговора.



- **Alerts**

Типично представя известие за нещо под формата на pop-up с две възможности (или без) за продължение



UIAlertSheet

- **Инициализация**

```
-(id)initWithTitle:(NSString *)title delegate:(id <UIAlertSheetDelegate>)delegate  
cancelButtonTitle:(NSString *)cancelButtonTitle destructiveButtonTitle:(NSString *)  
destructiveButtonTitle otherButtonTitles:(NSString *)otherButtonTitles, ...;
```

- **Добавяне на още бутони**

```
-(void)addButtonWithTitle:(NSString *)buttonTitle;
```

- **Визуализиране**

```
UIAlertSheet *actionSheet = [[UIAlertSheet alloc] initWithTitle:...];
```

```
[actionSheet showInView:(UIView*)];
```

```
[actionSheet showFromRect:(CGRect)inView:(UIView*)animated:(BOOL)];
```

```
[actionSheet showFromBarButtonItem:(UIBarButtonItem *) animated:(BOOL)];
```

UIAlertSheet

- UIAlertController методи

- (void)actionSheet:(UIAlertSheet *)actionSheet didDismissWithButtonIndex:(NSInteger)buttonIndex

- (void)actionSheet:(UIAlertSheet *)actionSheet willDismissWithButtonIndex:(NSInteger)buttonIndex

- (void)willPresentActionSheet:(UIAlertSheet *)actionSheet

- (void)didPresentActionSheet:(UIAlertSheet *)actionSheet

- (void)actionSheetCancel:(UIAlertSheet *)actionSheet

- Установяваме избора на потребителя, чрез Delegate метод

- (void)actionSheet:(UIAlertView *)sender clickedButtonAtIndex:(NSInteger)index;

UIAlertSheet + Delegate

- ```
@interface... UIViewController <UIAlertSheetDelegate>
...
@implementation

- (IBAction)doSomething:(id)sender
{
 UIAlertController *actionSheet = [[UIAlertSheet alloc] initWithTitle:@"Сигурен ли сте?"
 delegate:self
 cancelButtonTitle:@"Не, колебая се"
 destructiveButtonTitle:@"Да, напълно!"
 otherButtonTitles:nil];

 [actionSheet showInView:self.view];
 [actionSheet release];
}

- (void)actionSheet:(UIAlertSheet *)actionSheet clickedButtonAtIndex:(NSInteger)buttonIndex
{
 // the user clicked one of the OK/Cancel button
 if (buttonIndex == 0)
 {
 NSLog(@"ok");
 }
}

@end
```

# UIAlertView

- **Инициализация**

```
-(id)initWithTitle:(NSString *)title message:(NSString *)message
 delegate:(id <UIActionSheetDelegate>)delegate
 cancelButtonTitle:(NSString *)cancelButtonTitle
 otherButtonTitles:(NSString *)otherButtonTitles, ...;
```

- **Добавяне на още бутони програмно**

```
-(void)addButtonWithTitle:(NSString *)buttonTitle;
```

- **Визуализиране**

```
UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:...];
[alertView show]; // винаги се визуализира в центъра на екрана
```

# UIAlertView

```
@interface MyClass : NSObject <UIAlertViewDelegate>
..
@implementation

- (IBAction)doSomething:(id)sender {
 UIAlertView *alert = [[UIAlertView alloc]
 initWithTitle:@"Пауза"
 message:@"Желаете ли да продължите?"
 delegate:nil
 cancelButtonTitle:@"OK"
 otherButtonTitles:nil];

 [alert show];
 [alert release];
}

- (void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)buttonIndex {
 if (buttonIndex == 0) {
 NSLog(@"потребителя избра OK");
 }
 else {
 NSLog(@"потребителя избра Cancel");
 }
}
@end
```

# UIActivityIndicatorView



- Вид и стил

```
UIActivityIndicatorViewStyleGray
UIActivityIndicatorViewStyleWhite
UIActivityIndicatorViewStyleWhiteLarge
```

- Показване

```
UIActivityIndicatorView *myIndicator = [[UIActivityIndicatorView alloc]
initWithActivityIndicatorStyle:UIActivityIndicatorViewStyleWhiteLarge];
myIndicator.center = CGPointMake(160, 240);
myIndicator.hidesWhenStopped = NO;
```

- Старт / Стоп

```
[myIndicator startAnimating];
[myIndicator stopAnimating];
```

# UIActivityIndicator + Status bar

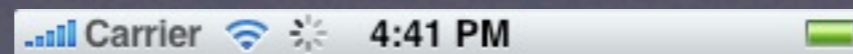
- Статус

```
if ([myIndicator isAnimating])
 NSLog(@"animating");
else

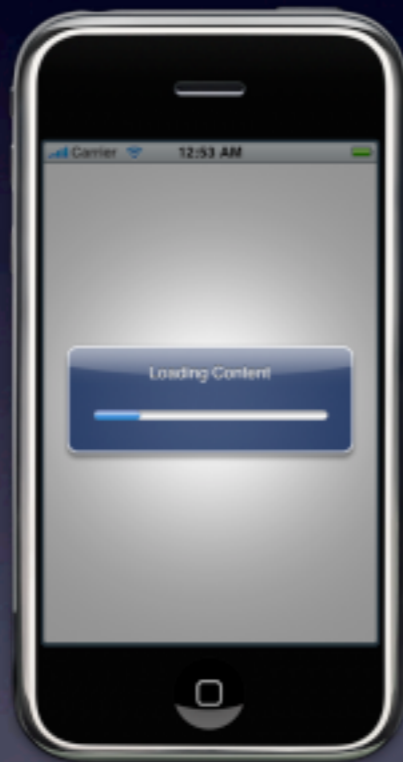
 NSLog(@"not animating");
```

- Управление на състоянието

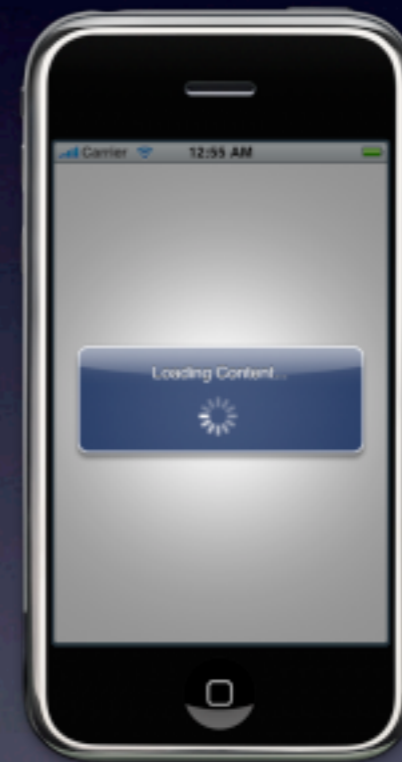
```
[UIApplication sharedApplication].networkActivityIndicatorVisible
```



# UIAlertView и UIActivityIndicatorView или UIProgressView



UIProgressView



UIActivityIndicatorView

# UIAlertView и UIActivityIndicatorView или UIProgressView

```
- (void) createProgressionAlertWithMessage:(NSString *)message withActivity:(BOOL)activity
{
 progressAlert = [[UIAlertView alloc] initWithTitle: message
 message: message
 delegate: self
 cancelButtonTitle: nil
 otherButtonTitles: nil];

 // Create the progress bar and add it to the alert
 if (activity) {
 activityView = [[UIActivityIndicatorView alloc]
 initWithActivityIndicatorStyle:UIActivityIndicatorViewStyleWhite];
 activityView.frame = CGRectMake(139.0f, 18.0f, 80.0f, 37.0f, 37.0f);
 [progressAlert addSubview:activityView];
 [activityView startAnimating];
 } else {
 progressView = [[UIProgressView alloc] initWithFrame:CGRectMake(30.0f, 80.0f, 225.0f, 90.0f)];
 [progressAlert addSubview:progressView];
 [progressView setProgressViewStyle: UIProgressViewStyleBar];
 }
 [progressAlert show];
 [progressAlert release];
}
```

# UIImagePickerController

- Използване на изображения от Photo Library
- Заснемане на изображения

Нужно е имплементиране на Delegate методите

`UIImagePickerControllerDelegate`

# UIImagePickerController

- Визуализиране

```
UIImagePickerController * picker = [[UIImagePickerController alloc] init];
picker.delegate = self;
```

- Известяване на delegate, когато е приключил потребителя

```
- (void)imagePickerController:(UIImagePickerController *)picker
 didFinishPickingMediaWithInfo:(NSDictionary *)info {

 // extract image/movie data/metadata here
 [self dismissModalViewControllerAnimated:YES];
}
```

# UIImagePickerController

- Какво се “крие” в

(NSDictionary \*)info ?

UIImagePickerControllerMediaType // kUTTypeImage or kUTTypeMovie

UIImagePickerControllerOriginalImage // UIImage

UIImagePickerControllerEditedImage // UIImage

UIImagePickerControllerCropRect // CGRect

UIImagePickerControllerMediaMetadata // Can be stored in saved photo album with data

# UIImagePickerController

- Overlay View

Винаги е full screen (модален в iPhone/iPod)

Удобен в случай, че имаме нужда да добавим built-in контроли.

Използва се за Augmented reality

```
@property UIView *cameraOverlayView;
```

- Скриване на стандартните бутони

```
@property BOOL showsCameraControls;
```

В този случай следва да имплементираме over view с поне един бутон който да изпраща на picker-а съобщение

```
- (void)takePicture
```

# UIImagePickerController

```
#import <UIKit/UIKit.h>
```

```
@interface PhotoAppViewController : UIViewController
 <UIImagePickerControllerDelegate,
 UINavigationControllerDelegate> {
 UIImageView * imageView;
 UIButton * choosePhotoBtn;
 UIButton * takePhotoBtn;
}
```

```
@property (nonatomic, retain) IBOutlet
 UIImageView * imageView;
@property (nonatomic, retain) IBOutlet
 UIButton * choosePhotoBtn;
@property (nonatomic, retain) IBOutlet
 UIButton * takePhotoBtn;

-(IBAction) getPhoto:(id) sender;
```

```
@end
```

```
@implementation PhotoAppViewController
@synthesize imageView,choosePhotoBtn, takePhotoBtn;
```

```
-(IBAction) getPhoto:(id) sender {
 UIImagePickerController * picker =
 [[UIImagePickerController alloc] init];

 picker.delegate = self;

 if((UIButton *) sender == choosePhotoBtn) {
 picker.sourceType =
 UIImagePickerControllerSourceTypeSavedPhotosAlbum;
 } else {
 picker.sourceType =
 UIImagePickerControllerSourceTypeCamera;
 }
 [self presentViewController:picker animated:YES];
}
...
- (void)imagePickerController:(UIImagePickerController *)picker
 didFinishPickingMediaWithInfo:(NSDictionary *)info {

 [picker dismissModalViewControllerAnimated:YES];
 imageView.image =
 [info objectForKey:@"UIImagePickerControllerOriginalImage"];
 ...
 // UIImageWriteToSavedPhotosAlbum (imageToSave, nil, nil , nil);
}
@end
```

# UIImagePickerController

Проверка дали устройството разполага с камера

```
if ([[UIImagePickerController isSourceTypeAvailable: UIImagePickerControllerSourceTypeCamera] == NO)
 || (delegate == nil)
 || (controller == nil))
 return NO;
```

...

Delegate метод в случай че потребителя откаже действието

```
- (void) imagePickerControllerDidCancel: (UIImagePickerController *) picker {
 [[picker parentViewController] dismissModalViewControllerAnimated:YES];
 [picker release];
}
```