Creating custom iOS UIButtons

Introduction

Not only functionality is the driving factor of good applications, also the look and feel is very important. This tutorial will guide you through some steps of creating custom buttons, this are buttons with a custom look and feel. The starting point of this tutorial is the StopWatch application. You can download the sample project file: StopWatch

What will we do:

- Change the color of a UIButton

SEARCH

Search GO

CATEGORIES

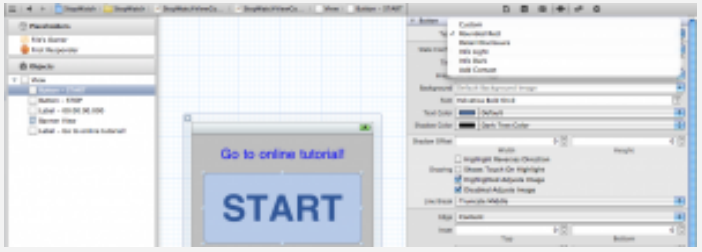
- AdMob (1)
- Drupal (1)
- iAd (1)
- iOS (20)
- iOS Tutorial (10)
- iPad (6)
- iPhone (15)
- MySQL (1)
- Obj-C (1)

- Change the label of a UIButton
- Use an image as UIButton background
- Use gradients to create shiny buttons
- Extras

If you like this tutorial check out the application itself and increases my iAd revenue! 😉

Changing the color of a UIButton

Before changing anything in code we need to change the Button type from Rounded Rect to Custom as shown inside the following screenshot:



Changing the button type from "Rounded Rect" to "Custom"

Do this for both the start and stop button as we will be changing both buttons.

Of course we also need a reference to the buttons we want to change. To do this open the Assistant View and let it display the header file StopWatchViewController.h. Select the start button and Control-drag to the header file and insert an IBOutlet called startButton, do the same for the stop button. Your header file should be updated and included following extra lines of code:

```
1  @property (nonatomic, retain) IBOutlet UIButton *startButton;
2  @property (nonatomic, retain) IBOutlet UIButton *stopButton;
```

Now open the source file StopWatchViewController.m and locate the method viewDidLoad. Add the following code:

```
1  - (void)viewDidLoad
2  {
```

```
3       [super viewDidLoad];
4
5       [[startButton layer] setCornerRadius:8.0f];
6       [[startButton layer] setBorderWidth:1.0f];
7       [startButton setBackgroundColor:[UIColor greenColor]];
8
9       [[stopButton layer] setCornerRadius:8.0f];
10      [[stopButton layer] setBorderWidth:1.0f];
11      [stopButton setBackgroundColor:[UIColor redColor]];
12  }
```

Lets discuss this line by line:

1. The method setCornerRadius specifies the radius used to draw the rounded corners of the background of the UIButton
2. The method setBorderWidth sets the border width
3. The method setBackgroundColor sets the background color of the button to green
4. The lines 9 - 11 apply the same changes to the stop button

Before trying the changes you will need to import the header QuartzCore.h:

```
1  #import <QuartzCore/QuartzCore.h>
```

Feel free to compile and run the application. The output should look like the following screenshot:

Start and stop button with updated background colors.

Changing the label of a UIButton

The first thing we will do is change the label of the stop button so that it no longer fits. To do this open the file called StopWatchViewController.m and locate the method viewDidLoad. Inside this method add the following code to update the start button title label:

```
1  [startButton setTitle:@"START BUTTON" forState:UIControlStateNormal];
```

If you run the application again the output should be like this:

The button title gets truncated in the middle

Now we are going to add a few lines of code that will change the look and feel of the label completely. Again inside the method viewDidLoad add the following lines of code:

```
startButton.titleLabel.lineBreakMode = UILineBreakModeWordWrap;
startButton.titleLabel.numberOfLines = 0;
startButton.titleLabel.textAlignment = UITextAlignmentCenter;
startButton.titleLabel.font = [UIFont fontWithName:@"MarkerFelt-Wide" size:42];
[startButton setTitleColor:[UIColor whiteColor] forState:UIControlStateNormal];
[startButton setTitleShadowColor:[UIColor redColor] forState:UIControlStateNorm
startButton.titleLabel.shadowOffset = CGSizeMake(3.0f, 3.0f);
```

Lets again discuss line by line:

1. The line break mode was set to wrap or clip on the boundaries of words by setting the property lineBreakMode to UILineBreakModeWordWrap
2. The number of lines was set to 0, this forces the control to use as many lines as needed to display the string
3. The alignment was set to center by setting the property textAlignment to UITextAlignmentCenter
4. Setting a new font is also very easy and can be done by setting the font property to a new UIFont with a

predefined font name and size. If you want to keep the default font use systemFontOfSize instead of fontWithName. TIP: look at http://iosfonts.com/ for the fonts installed on iOS

5. The next call will change the font color to white for the normal button states. This is done with the call setTitleColor
6. A shadow is always nice and this can be done with just 2 lines of code. The first line of uses setTitleShadowColor to set the shadow color and is followed by updating the property shadowOffset to set the shadow offset.

Running the application should result in the following button look:

The start button with custom label and fonts

If this is esthetically correct is an open question 😉

Using a background image with UIButton

Setting your own image as button background can be done very easily. The only steps required are:

1. Adding the image to the project, so that it gets included in the application bundle and becomes available

2. Creating an UIImage from you own and setting it as background image

The image that we will be using will look like this:

The button background (280 width x 125 height)

If you want you can download this image and add it to the “Supporting Files” group of you project.

Once this is done we will be adding some code to the method viewDidLoad and this time we will also comment some code. The snippet from viewDidLoad looks like:

```
UIImage *backgroundImage = [UIImage imageNamed:@"buttonbackground.png"];
[startButton setBackgroundImage:backgroundImage forState:UIControlStateNormal];
/*
  [[startButton layer] setCornerRadius:8.0f];
  [[startButton layer] setBorderWidth:1.0f];
  [startButton setBackgroundColor:[UIColor greenColor]];
*/
```

What is done here:

1. An UIImage is created from the background button
2. This image is set as the background image for the start button
3. The code that colored the start button green and gave it a border should be removed

The result should look like this:

An image as button background

So if you are good with some drawing program you can create whatever button you want.

Using gradients and shadows

This time we start with a big block off code! Add a new method called makeButtonShiny to the source file StopWatchController.m

```
- (void)makeButtonShiny:(UIButton*)button withBackgroundColor:(UIColor*)backgr
{
    // Get the button layer and give it rounded corners with a semi-transparan
    CALayer *layer = button.layer;
    layer.cornerRadius = 8.0f;
    layer.masksToBounds = YES;
    layer.borderWidth = 4.0f;
    layer.borderColor = [UIColor colorWithWhite:0.4f alpha:0.2f].CGColor;

    // Create a shiny layer that goes on top of the button
    CAGradientLayer *shineLayer = [CAGradientLayer layer];
    shineLayer.frame = button.layer.bounds;
    // Set the gradient colors
```

```
    shineLayer.colors = [NSArray arrayWithObjects:
                         (id)[UIColor colorWithWhite:1.0f alpha:0.4f].CGColor,
                         (id)[UIColor colorWithWhite:1.0f alpha:0.2f].CGColor,
                         (id)[UIColor colorWithWhite:0.75f alpha:0.2f].CGColor
                         (id)[UIColor colorWithWhite:0.4f alpha:0.2f].CGColor,
                         (id)[UIColor colorWithWhite:1.0f alpha:0.4f].CGColor,
                         nil];
    // Set the relative positions of the gradien stops
    shineLayer.locations = [NSArray arrayWithObjects:
                            [NSNumber numberWithFloat:0.0f],
                            [NSNumber numberWithFloat:0.5f],
                            [NSNumber numberWithFloat:0.5f],
                            [NSNumber numberWithFloat:0.8f],
                            [NSNumber numberWithFloat:1.0f],
                            nil];

    // Add the layer to the button
    [button.layer addSublayer:shineLayer];

    [button setBackgroundColor:backgroundColor];
}
```

Lets discuss the code in more details:

- We first create a semi transparent border around the button. The border width is set to 4 pixels by updating the property borderWidth. The color itself for the border is set by updating the property borderColor. By using the method colorWithWhite it is possible to get a transparent like border that takes the background color.
- Next we create a new layer with the same bounds as the button itself. The CAGradientLayer can be created easily by specifying the different “stop colors” and their relative locations. Setting the stop colors is done by updating the property colors and setting the relative locations is done by updating the property locations. So in this case a gradient with 5 transparent white flavors is created with 5 relative locations.
- Next we add a new layer to the button to have the transparent overlay gradient
- Finally the passed in backgroundColor is set as background color for the button
- **Remark**: Don’t forget to add the Framework “QuartzCore.framework”!

Now that the new method is in place we are going to use it to change the look of the stop button. Inside the method viewDidLoad add the following code:

```
[self makeButtonShiny:stopButton withBackgroundColor:[UIColor redColor]];
```

Compile and run your application. The result should look like this:

The Stop button with a shiny gradient as overlay

I would say experiment with the different methods of changing the look and feel until you find something that's your liking!

Extras

You can download the sample project file: StopWatch Custom Buttons Project

If you wan't to sponsor these tutorials you can download the StopWatch Full edition from the App Store => COMING SOON

Print article

This entry was posted by Luc Wollants on May 3, 2011 at 08:21, and is filed under iOS, iOS Tutorial, iPhone. Follow any responses to this post through RSS 2.0. You can skip to the end and leave a response. Pinging is currently not allowed.

COMMENTS (1)

#1 written by best auto insurance rates
about 6 months ago

Thanks very nice blog!

Name (required)

E-mail (required, will not be published)

Website

Submit Comment