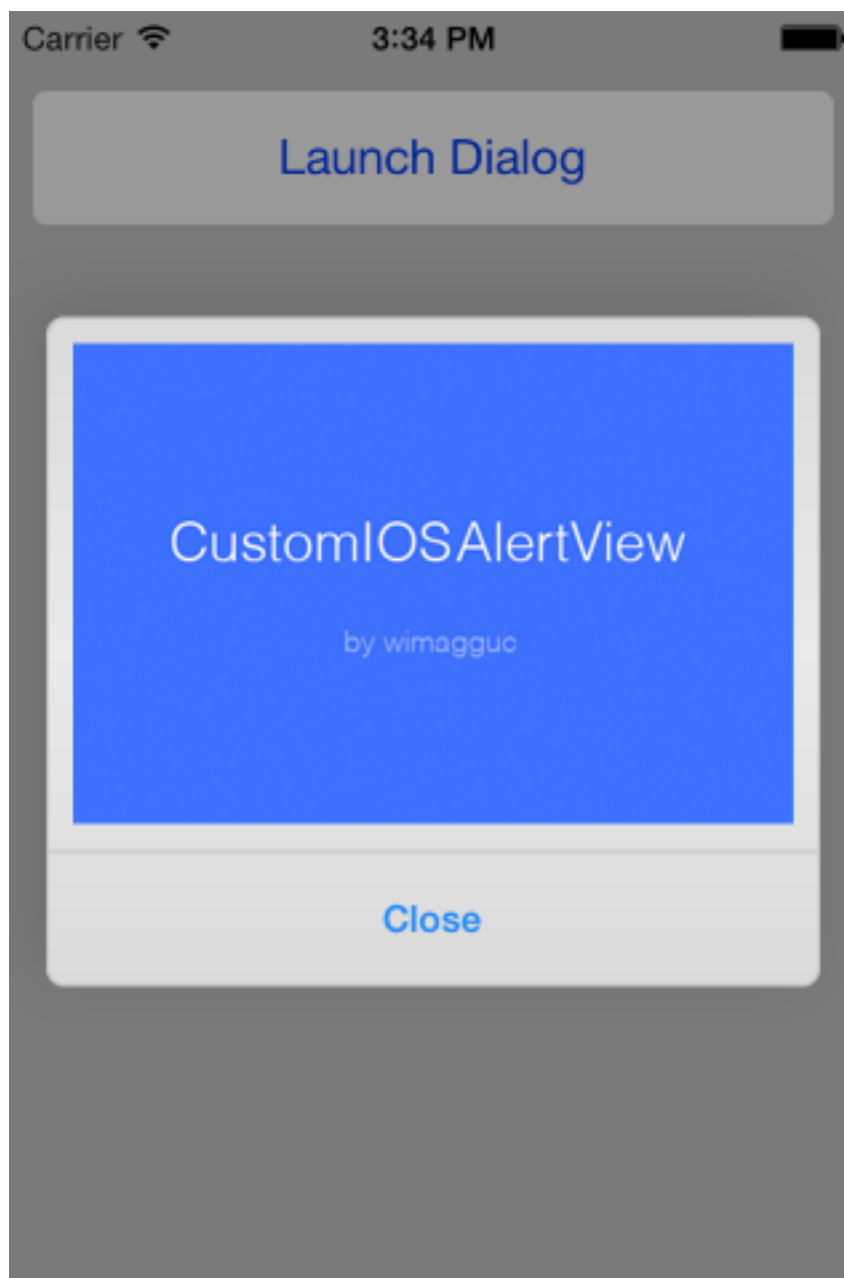

Custom iOS UIAlertView

v0.9.5

support **for** iOS7+

The addSubview is not available in UIAlertView since iOS7. The view hierarchy for this class is private and must not be modified.

As a solution, this class creates an iOS-style dialog which you can extend with any UIViews or buttons. The animations and the looks are copied too and no images or other resources are needed.



Install

As simple as adding the following files to your project:

- CustomIOSAlertView.h
- CustomIOSAlertView.m

Or use Cocoapods:

```
1 pod 'CustomIOSAlertView', '~> 0.9.5'
```

Change notes

- Fixed rotation for IOS8
- Removed 7 from the class name. Just use CustomIOSAlertView from now on, like: `[[CustomIOSAlertView alloc] init];`
- The `initWithParentView` method is now deprecated. Please use the `init` method instead, where you don't need to pass a parent view at all. **In case the `init` doesn't work for you, please leave a note or open an issue here.**

Quick start guide

1. Create the UIView object `changed`

```
1 CustomIOSAlertView *alertView = [[CustomIOSAlertView alloc] init];
```

2. Add some custom content to the alert view (optional)

```
1 UIView *customView ..;  
2  
3 [alertView setContainerView:customView];
```

3. Display the dialog

```
1 [alertView show];
```

More functions

- Close the dialog

```
1 [alertView close];
```

- To add more buttons, pass a list of titles

```
1 [alertView setButtonTitles:[NSMutableArray arrayWithObjects:@"  
    Button1", @"Button2", @"Button3", nil]];
```

- You can remove all buttons by passing nil

```
1 [alertView setButtonTitles:NULL];
```

-
- You can enable or disable the iOS7 parallax effects on the alert view

```
1 [alertView setUseMotionEffects:TRUE];
```

- Handle button clicks with a custom delegate

First, set the delegate:

```
1 [alertView setDelegate:self];
```

Then add the delegate methods:

```
1 - (void)customIOS7dialogButtonTouchUpInside: (CustomIOSAlertView *  
    *)alertView clickedButtonAtIndex: (NSInteger)buttonIndex  
2 {  
3     NSLog(@"Button at position %d is clicked on alertView %d.",  
        buttonIndex, [alertView tag]);  
4 }
```

- Handle button clicks with a code block

```
1 [alertView setOnButtonTouchUpInside:^(CustomIOSAlertView *  
    alertView, int buttonIndex) {  
2     NSLog(@"Block: Button at position %d is clicked on alertView %  
        d.", buttonIndex, [alertView tag]);  
3     [alertView close];  
4 }];
```

You can also disable all other delegates by:

```
1 [alertView setDelegate:self];
```

Todos

This is a really quick implementation, and there are a few things missing:

- Adding more buttons: they don't exactly match the look with that of on iOS7
- Rotation: rotates wrong with the keyboard on

Special thanks to

- @tamasdancsi for his support with the initial code
- @dingosky for his work on the parallax effects code

-
- @raspu for his work on the protocol delegates, iOS6 support and onClick blocks
 - @sbandol for his idea on adding the UIAlertView as the top most view in the hierarchy
 - @scorpiozj for his work on the rotation code
 - @kwent for adding performance optimisations
 - @thomasaw for the refactor on the deprecated initWithParentview
 - @yiboyu for the multi-line button fix
 - @logicxd for the close-on-touchup-outside code

License

MIT. Please read LICENSE.md for more info.

Other projects

Check out Appwoodoo for push notifications and remote app control service. Open source SDKs are available on Github: github.com/appwoodoo.

Some more of my free stuff for web devs at Github.

Project updates newsletter

About

Richard Dancsi

www.wimagguc.com

twitter: @wimagguc

linkedin: [linkedin.com/in/richarddancsi](https://www.linkedin.com/in/richarddancsi)