

Creating Views Programmatically

Up to this point, you've probably used a variety of storyboards and Xib files to create your interfaces, but it's important to know that everything you can do in Interface Builder you can also do using code.

Another important concept to understand is that the files that we use to structure and organize our projects are just that, a way to structure and organize our code for human readability, but not really required by the computer.

For this project, you can only create interface elements programmatically and you can only use one file (*main.m*).

Read the following programming guide and make sure you understand the following concepts:

- *UIWindow*
- *UIScreen*
- *UIView*
- *makeKeyAndVisible* function
- *the difference between frame and bounds*
- *adding subviews to a view*

View Programming Guide for iOS

https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewPG_iPhoneOS/Introduction/Introduction.html

Read the following programming guide to understand Event Handling.

Event Handling Guide for iOS

<https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/Introduction/Introduction.html>

Assignment:

1. Explore the sample and fully understand it.
2. Recreate the sample project
3. Create a new project of your own design that has an entirely programmatic UI. Make it look good.
4. Add the following gestures to items in your interface (be creative):
 - a. *UITapGestureRecognizer* to implement single tap, double tap and two-finger tap on a view.
 - b. *UIPinchGestureRecognizer* to implement zoom in and out on an image
 - c. *UIPanGestureRecognizer* to implement pan/dragging of a view
 - d. *UISwipeGestureRecognizer* to implement a swipe on the screen in any direction to change background color of the screen
 - e. *UIRotationGestureRecognizer* to rotate a view or image view
 - f. *UILongPressGestureRecognizer* to implement a long press / touch-and-hold functionality on a view.
5. Add a simple calculator feature that allows the user to enter a number into each of two text fields and press a button to add them together. The sum should appear in a *UILabel*.
6. Add a square in the bottom right corner. (*UIView*)
7. No matter the orientation, the square should always appear in the bottom right corner. (you **MUST** use the view controller's callbacks for this, **NOT AUTOLAYOUT**)

Completion Checklist:

- Does the square in the bottom right corner stay in the bottom right corner regardless of the orientation?
- Code review with instructor