Workshop Lesson Plan

1. Introduce self, ESW, and CSES. Give thanks.
2. Ask audience how many have coded with swift or XCode before (Make sure mentors are all in positions to help students)
3. Introduce how I got into swift; history of it and why it is going to be useful in the future
4. Open up XCode and go over the layout of it.
5. Describe what we will be doing for the workshop and how this is just a step by step tutorial to give the basics.
6. Create a new project and navigate to the storyboard.
   1. Should have appdelegate and viewController.swift files
   2. Change top simulator setting to iPhone 6s
   3. Change bottom layout setting to w-Compact, h-any (for iPhone app)
7. Introduce toolbars, have everyone navigate to bottom right toolbar to add first label.
8. Make sure everyone is zoomed in properly so they can actually drag label onto storyboard
9. Have everyone drag a button from the same toolbar to the middle of the screen. Change button name to “Let’s play!”
10. Add a second view controller from the toolbar to the right of the first one.
11. Right click and drag from the button to the second View Controller and hit present modally.
    1. This makes it so our first view controller acts as a title screen.
12. Highlight all objects and go to constraint panel. Hit “add missing constraints.”
    1. Can be tedious to go through each object individually and add constraints.
    2. Tool makes it quick and easy to add constraints for all objects in a view.
    3. If a constraint is not proper, you can always click on that object, view the constraints on it and change them individually later.
13. Add two buttons to second view controller
    1. Top named Counter Button
    2. Button below named Reset
14. Add a Label above Counter Button.
    1. Change label to 0.
    2. Change size of label to fit whole screen.
15. Add 2 Labels in top left side.
    1. One for Highscore
    2. One for actual score (make it 0)
16. Highlight all objects and add missing constraints.
17. Go to File…Create and create a new Cocoa Touch File.
    1. Class: gameViewController
    2. subclass: UIViewController
    3. language: Swift
18. Make sure when file is created:
    1. Import UIKit
    2. Override viewDidLoad(){ super.viewDidLoad()}
19. Navigate to Storyboard and click secondViewController. Change type to gameViewController.
20. Change to dualScreen mode and have coding screen on the right.
21. Connect all 3 changing objects to the gameViewController file
    1. Highscore for highscore label
    2. Counter for the counter label
    3. Timerlabel for timer label
22. Create two button actions by right-clicking the button and dragging to the file.
    1. makeCount
    2. ResetCount
23. Add a variable to keep counts named buttonPresses = 0
24. To makeCount Action:
    1. buttonPresses+=1 and counter.text = String(buttonPresses)
25. To resetCount Action:
    1. buttonPresses = 0 and counter.text = String(buttonPresses)
26. Now we want to set the Highscore to update!
    1. Create variable for scoreVal and set equal to 0
    2. Add 3 lines to makeCount Button method

if(scoreVal < buttonPresses){){

scoreVal = buttonPresses

score.text = String(scoreVal)

}

1. Next is saving our data when we close the app.
   1. Add following code to makeCount method
      1. let scoreDefault = UserDefaults.standard
      2. scoreDefault.setValue(scoreVal,forKey:"scoreVal")
      3. scoreDefault.synchronize()
   2. Add following code to viewDidLoad() method
      1. let scoreDefault = UserDefaults.standard
      2. if(scoreDefault.value(forKey:"scoreVal") != nil){
      3. scoreVal = scoreDefault.value(forKey:"scoreVal") as! NSInteger
      4. score.text = String(scoreVal)}
2. Finally is timer implementation.
3. Start by creating timer object globally.
4. Within the makeCount method
   1. if(buttonPresses == 0){
   2. timer = Timer.scheduledTimer(timeInterval: 1, target: self, selector: #selector(gameViewController.updateTimer), userInfo: nil, repeats: true)}
5. create updateTimer method.
   1. Timer.invalidate() if time is 0
   2. timeVal-=1 and timeLabel.text = String(timeVal)
6. Within reset method add timer.invalidate()
7. Finally add some color
   1. Self.view.backgroundColor = UIColor.red (or green)