Saving Data

You'll need:

- The third textbook, **Data Collections**, Lesson 1.7
- The **Random Colors in Swift** project from Canvas, completed
- (The video for the Random Colors project is about 17 minutes long)

Part 1: Add a scene to show saved color schemes

1. Add a second **View Controller** onto the storyboard
2. Add a **Horizontal Stack View** into the new view controller
3. Add a **UIView** and a **UIButton** into the stack view
4. Duplicate the UIView until there are 5 of them
5. Put a light gray color on all the 5 views (these will be the color wells)
6. Duplicate the UIButton so there are 2 of them
7. Put a **magnifying class** on the first button and a **red X** on the second button
8. Put any light background color on both the buttons; light blue works well
9. Set the Stack View to **Alignment: Fill** and **Distribution: Fill Equally**
10. Embed the Horizontal Stack View into a **Vertical Stack View**
11. Give the Vertical Stack View constraints to
    a. Center it vertically
    b. Make it 300 high
    c. Make it go to 20 pixels from the right and left edges (using I-beams)
12. Set the Vertical Stack View to **Alignment: Fill** and **Distribution: Fill Equally**
13. Duplicate the Horizontal Stack View until there are 8 rows

Part 2: Add a tab bar controller

1. Embed the first view into a Tab Bar Controller
2. Add the second view as the second tab
3. Change the title of the first Tab to **New Color** and give it a paint brush icon
4. Change the title of the first tab to **Saved Colors** and give it a 3x2 grid icon

Part 3: Add the code for saving color schemes

1. Add a new **Swift File** and call it **Color.swift**
2. Add a **Codable** structure for holding a color

```
struct Color : Codable {
      var red : CGFloat
      var green : CGFloat
      var blue : CGFloat
}
```

3. Add a Codable structure for holding a color scheme

```
struct ColorScheme : Codable {
      var colors : [Color]
}
```

4. Add an array of color schemes

```
var colorSchemes : [ColorScheme] = []
let maxColorSchemes = 8
```

5. Add a function to add a color scheme to the array

```
func addColorScheme (_ colorScheme : ColorScheme) {
      if colorSchemes.count >= maxColorSchemes {
            return
      }
      colorSchemes.append(colorScheme)
}
```

6. Add a function to remove a color scheme from the array

```
func removeColorScheme (at index : Int) {
      if index >= 0 && index < colorSchemes.count {
            colorSchemes.remove(at: index)
      }
}
```

7. Add a function to save the color schemes to a file

```
var archiveURL : URL {
      let documentsDirectory = FileManager.default.urls(
           for: .documentDirectory, in: .userDomainMask).first!
      return documentsDirectory .appendingPathComponent(
          "myColorSchemes").appendingPathExtension("plist")
}

func saveColorSchemes() {
      let propertyListEncoder = PropertyListEncoder()
      if let encodedColorSchemes = try?
            propertyListEncoder.encode(colorSchemes) {
            try? encodedColorSchemes.write(
               to: archiveURL, options: .noFileProtection)
      }
}
```

8. Add a function to load the color schemes from a file

```
func loadColorSchemes() {
      let propertyListDecoder = PropertyListDecoder()
      if let retrievedColorSchemes = try? Data(
              contentsOf: archiveURL),
         let decodedColorSchemes = try?
      propertyListDecoder.decode([ColorScheme].self,
          from: retrievedColorSchemes) {
```

```
            colorSchemes = decodedColorSchemes
        }
    }
```

Part 4: Add the code for displaying the saved colors

1. Add a button and action method to the **first screen** to add and save a color scheme

```
    @IBAction func save(_ sender: UIButton) {
        addColorScheme(currentColorScheme)
        saveColorSchemes()
        tabBarController?.selectedIndex = 1
    }
```

2. Add a new UIViewController and call it **SavedColorsViewController**
3. Using the Identity Inspector, connect the SavedColorsViewController to the second scene
4. In the **SavedColorsViewController,** make an outlet for the master stack view and call it **masterStackView**
5. Add code to retrieve the saved colors and display them in the color wells

```
    func showColorSchemes() {
        for index in 0..<maxColorSchemes {
            showColorScheme(index: index)
        }
    }

    func showColorScheme(index: Int) {
        if let stackRows =
            masterStackView.subviews as? [UIStackView] {
            let stackRow = stackRows[index]
            let cells = stackRow.subviews
            if index < colorSchemes.count {
                for i in 0...4 {
                    cells[i].backgroundColor =
                        colorToUIColor(
```

```
                            colorSchemes[index].colors[i])
                }
            }
            else {
                for i in 0...4 {
                    cells[i].backgroundColor =
                        colorToUIColor(
                            neutralColorScheme.colors[i])
                }
            }
        }
    }
```

6. Add a viewWillAppear function and load and show the saved color schemes each time

```
    override func viewWillAppear(_ animated: Bool) {
        loadColorSchemes()
        showColorSchemes()
    }
```

7. In the SavedColorsViewController, add code for the X button to delete a color

```
    @IBAction func deleteColorScheme(_ sender: UIButton) {
        removeColorScheme(at: sender.tag)
        saveColorSchemes()
        showColorSchemes()
    }
```

8. In the ViewController file, add code to display a particular color scheme

```
func magnifyColorScheme (index : Int) {
      currentColorScheme = colorSchemes[index]
      for i in 0...4 {
          views[i].backgroundColor = colorToUIColor(
```

```
                currentColorScheme.colors[i])
        }
    }
```

9. In the SavedColorsViewController, add code for the magnifying glass button to display that color on the first screen, by calling its magnifyColorScheme method.

```
    @IBAction func magnify(_ sender: UIButton) {
        if let vc = tabBarController?.viewControllers?[0]
            as? ViewController {
            vc.magnifyColorScheme(index: sender.tag)
            tabBarController?.selectedIndex = 0
        }
    }
```