PROJECT #3

The Problem

Your task is to develop an application that allows you to place an 'unlimited' number of squares (UIViews) on screen. The user will start on a "landing page" where he/she can select to enter the "Red Square Arena" or the "Blue Square Arena". Every time you tap somewhere on the view in either the red or green arenas, you will place a new square on screen with that corresponding color. The images for what the project should look like are on the next page. In this assignment, you should tackle the challenge problems as this part shouldn't take you very long since it is basically a recap of what we did in lecture.

Challenge Problems

If you tackle the challenge problems, please make new "arenas" where specified.

1. The way the original problem is setup is that both arenas will be added with the navigation controller. Let's change this so that the view controller managing the blue squares will be presented modally. Be sure to include a way to get back!

2. (Please add a 3rd "arena" if you are tackling this problem). Create a new subclass of **`UIView`** whose background color is randomly chosen upon creation. Every new tap on the screen should then use this class to display a square with a random color.

3. Are you creating different classes for each arena? See if you can simplify this by having one **`ShapeViewController`** that can display different shapes depending on which button was clicked on the root view controller. Hint: You can pass information between view controllers via the **`prepareForSegue(_:sender:)`** method and accessing the segue's destination view controller.

4. Introduce a **`UISlider`** to the application that can control the size of the square that will be placed the next time you tap. You can even add sliders for the height and another for the width.

5. You might find that it's easy to add all these views, but how can we change their transparency (alpha value) or change all of their background colors after we've added them on screen? For this challenge, add the ability to show or hide all the subviews after they've been added.

6. Add an eraser functionality that will remove any view that is placed on screen when it is tapped. You can do this in either **`touchesBegan`** or **`touchesMoved`** if you want to get fancy. You should have a toggle or some way to activate the eraser and then reactivate the ability to add shapes.

7. (Super challenge problem) Instead of adding square views to the view, try and add triangles instead. Hint: you will still use a UIView to do this. (Again, make a 4th “arena” if you are going to add this feature)

Reference

- Lecture #3
- UIViewController Class Reference
- UIView Class Reference
- UIColor Class Reference
- UISlider Class Reference
- UIStoryboardSegue Class Reference
- (for the triangle problem) Either UIView’s drawRect: method or the CAShapeLayer class (there are many ways to achieve this)
- The Swift Programming Language book