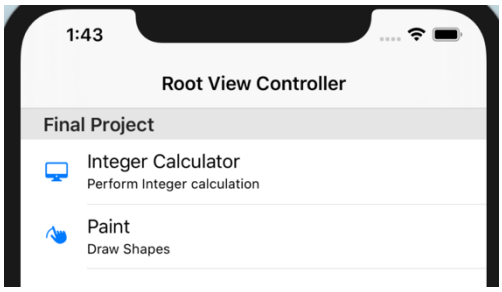
CSC 471 iOS development
Professor: Xiaoping Jia
Name: Frank Wang
Date: 03/15/2020

Final Project Documentation

A. Final Project Description and Documentation



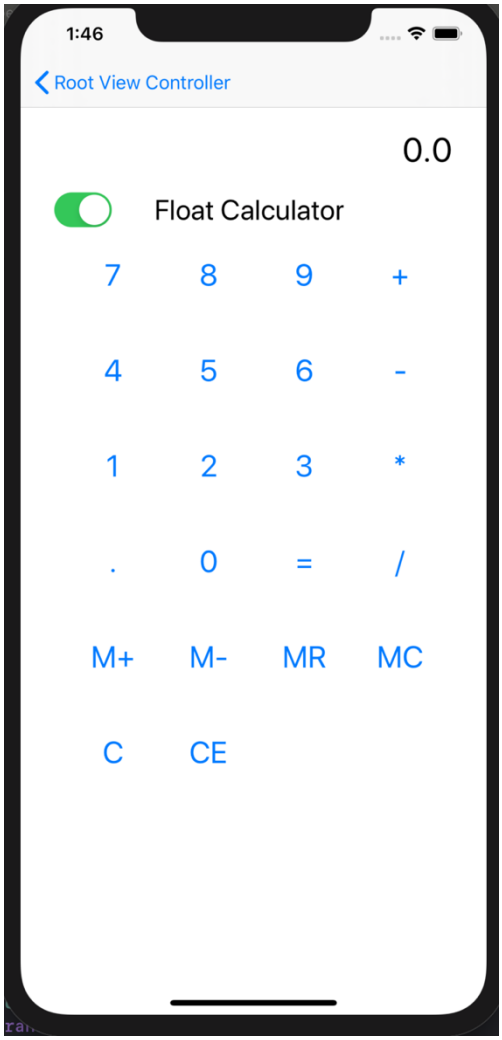
My final project has 2 separate parts:

1. Calculator and 2. Paint

1. Main Screen

a. Using navigation view to select from different programs.

b. Add different icon image to indicate different apps



2. Calculator

a. Based on assignment3, now not only + but also -, *, /

b. Add M+, M-, MR, MC function

c. Add C, CE function

d. Add float calculator, using one switch to change between int and float calculator

3. Paint (Screen Shot next page)

a. Use a picker to pick from different shapes, drawing type, or color.

b. If the drawing type has outline, I will ask the user to define an outline color.

c. Use enum to create list.

d. Use dictionary to look for different type of values.

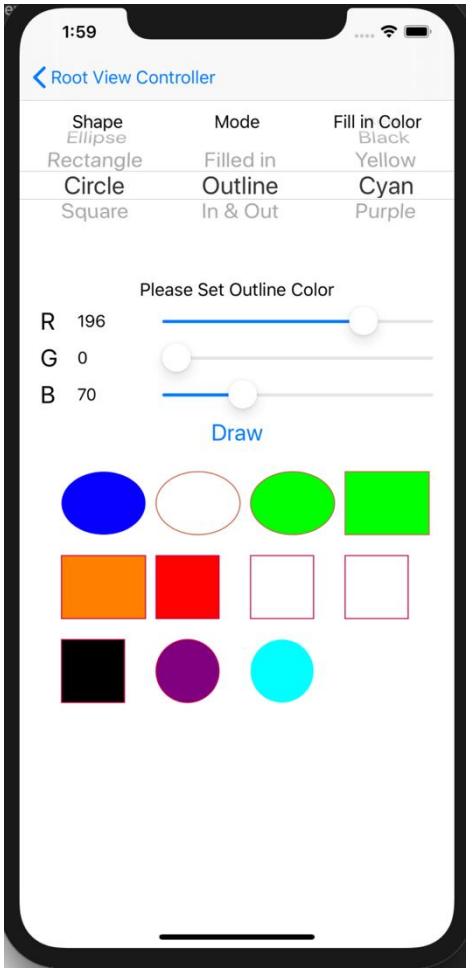
e. Use factory design pattern to define different shapes.

f. Use strategy design pattern to define different draw type.

g. Every time click “draw” button, it will draw at designed position.

h. Detailed discussion will be in the video

B. Final Project Discussion



1. API features:

Navigation View : main screen
Static Table : main screen
Picker : paint scene
View: paint scene / canvas
Switch : calculator scene
Slider : paint scene
Alert: alert message when shape is drawn out of canvas

2. Biggest challenges:

a. Dynamically change type from Int to Float, I believe using generic type will be a good solution, I just used a Boolean value to indicate if the value should be int or float. This ugly method gave me some plan on the paint app.

b. In the paint app, I want to draw shapes and keep what I have drawn. All the demo code does not have this feature. In order to do this, I have to save all different shapes in a list. So that I create a shape prototype, and all different shapes are its subclass.

c. When drawing the shape, I want to iterate through the shape list and call some drawing function. Since it has different drawing type, I want to pass the type to a common function, and that function will return the real function that is drawing. This is the strategy pattern, and I learned how to let a function return a function.

d. When Create a shape, instead of calling different constructors, I want to create different shapes with one function, so this is the factory pattern, and I learned how to let a function return an object.

3. Limitations of my app

a. I planned to add scientific calculator, but I notice it is just adding buttons and calling functions, which is redundant work. So that I did not do that part. If I have more time, I will do that part.

b. For the painting app, I planned to let the user draw with figure, however, the demo code already has that feature, and I do not want to copy and paste code from the demos.

c. When picking up a color, I used 3 sliders for rgb value, however, if I can use a color panel, it would be nicer.

4. Limitations of iOS SDK

a. When doing weekly assignment, I find that for the stepper, it is too hard to put it vertically. Even though we can use some rotation function to do that, the layout will also change. And also, the label on the stepper can only be + and -, which is limited in the design.

b. No automatic type converter is really a problem for me, especially int to float.

5. Overall experience

It is a great experience to learn iOS develop, and I really appreciate prof Jia always explain everything in to detail, the is one of the best courses that I have experienced in DePaul.

iOS SDK provides some wonderful features that other programming language does not have. When developing the final project, I really read more documentations that I expected, some features would be so hard to implement in other language, but in swift, it is just one line of code, the only thing is that, I have to know there is such a function.