Mobile Application Programming: iOS

CS4962 Spring 2016
Project 1 - Brush Chooser
Due: 11:59PM Monday Feb 8

Abstract

Create a fully-functional user interface in Swift that allows the selection of the style settings for a brush for a paint program. That is, stroke width, stroke end caps, stroke join, and paint color. The brush chooser can use any method to do selection that satisfies the components section of this document, but must include a brush preview view that draws a stroke using the brush settings with the CoreGraphics API. It must also include at least one custom control that draws its content using the CoreGraphics API.

The set of controls should also be visually pleasing and target an iPhone 6 Plus size screen in portrait mode. Put some thought into the overall design. If you're not feeling particularly creative, building the example shown below will give 90%. Improve its visual appeal in some way (gradients, backgrounds, better borders, shadows, glass sheen, etc.) to get the last 10%.

Components

- Main Composite View - A grouping view that will contain the other views, inherit from UIView or UIControl, and include public custom properties to programmatically get or set values for:
  - Stroke Width - values between 0.5 and 50.0 points
  - Stroke End Caps - Butt, Round, Projecting
  - Stroke Join - Miter, Round, Bevel
  - Paint Color - 200 or more individual colors spanning the rainbow
- Stroke Width Chooser - A view that allows selection of width between 0.5 and 50.0 points
- Stroke End Cap Chooser - A view that allows the user to select among the 3 end cap options
- Stroke Join Chooser - A view that allows the user to select among the 3 line join options
- Brush Preview - A view that draws a sample stroke using the settings above and:
  - Updates dynamically when the properties are changed programmatically or by the user
  - Has variation sufficient in the sample stroke to see how the stroke width, join, and end cap settings affect the stroke (so don't use just a straight line)
  - Has properties for stroke width, end caps, join, and color

Considerations

- Your brush chooser UI should be composed of sub-views that are subclasses of UIView or UIControl and use the target-action mechanism or delegation to post "value changed" notifications to registered listener objects. When the main composite view creates its sub-views, it should set itself up to receive update messages from them, then forward the changes to the brush stroke preview view, causing it to redraw with the new settings.
- When the user adjusts the brush settings using the sub-views, the public properties the main composite view offers should return those values when queried. This can be done by updating a stored property to reflect the changed values, or using a computed property that queries the appropriate sub-view for the value.

- At least one of the 4 property chooser views needs to be completely custom, with its content being drawn using the CoreGraphics API. The color choosing element is the most straightforward to implement, but the other choosers are also good candidates. The other choosers can use standard UI element(s) in their construction.
- 20% extra credit will be given to anyone with a physical-analog themed brush chooser. Think of a 70's guitar amp knob-based brush selector, steampunk-style brass-and-gear brush selector, Star Trek console brush selector, cave-man rock-and-twig brush selector, or contemporary brushed-steel and glass brush selector. Look around Lowe's or Home Depot in the appliance section for contemporary examples (physically or online).

Handin

You should hand in a zip file containing your project folder, including any image resources (please check that they are in the folder). Hand this zip into:

handin cs4962 project1 your_zip_file.zip