Stanford CS193p

Developing Applications for iOS
Fall 2017-18

Today

- Emoji Art
  Demo continued ...
- UITextField
  Editable text input control
  Demo: Add a text field to Emoji Art

Demo

- Emoji Art
  - Make our Emoji Art scrollable/zoomable/centered
  - Add a Collection View to choose emoji from
  - Drag and Drop from the Collection View

UITextField

- Like UILabel, but editable

  Typing things in on an iPhone is secondary UI (keyboard is tiny).

  More of a mainstream UI element on iPad.

  Don't be fooled by your UI in the simulator (because you can use physical keyboard!).

  You can set attributed text, text color, alignment, font, etc., just like a UILabel.

- Keyboard appears when UITextField becomes "first responder"

  It will do this automatically when the user taps on it.

  Or you can make it the first responder by sending it the becomeFirstResponder message.

  To make the keyboard go away, send resignFirstResponder to the UITextField.

- Delegate can get involved with Return key, etc.

  `func textFieldShouldReturn(sender: UITextField) -> Bool` // when "Return" is pressed

  Oftentimes, you will `sender.resignFirstResponder()` in this method.

  Returns whether to do normal processing when Return key is pressed (e.g. target/action).

UITextField

- Finding out when editing has ended

  Another delegate method …

  `func textFieldDidEndEditing(sender: UITextField)`

  Sent when the text field resigns being first responder.

- UITextField is a UIControl

  So you can also set up target/action to notify you when things change.

  Just like with a button, there are different UIControlEvents which can kick off an action.

  Right-click on a UITextField in a storyboard to see the options available.

Keyboard

Controlling the appearance of the keyboard

Remember, whether keyboard is showing is a function of whether its first responder.

You can also control what kind of keyboard comes up.

Set the properties defined in the `UITextInputTraits` protocol (`UITextField` implements).

```
var autocapitalizationType: UITextAutocapitalizationType   // words, sentences, etc.
var autocorrectionType: UITextAutocorrectionType           // .yes or .no
var returnKeyType: UIReturnKeyType                         // Go, Search, Google, Done, etc.
var isSecureTextEntry: Bool                                // for passwords, for example
var keyboardType: UIKeyboardType                           // ASCII, URL, PhonePad, etc.
```

Other Keyboard functionality

Keyboards can have accessory views that appear above the keyboard (custom toolbar, etc.).

```
var inputAccessoryView: UIView   // UITextField method
```

Keyboard

- The keyboard comes up over other views

So you may need to adjust your view positioning (especially to keep the text field itself visible).
You do this by reacting to the UIKeyboard{Will,Did}{Show,Hide} Notifications sent by UIWindow.
We have not talked about what a Notification is yet, but it's pretty simple.
You register a method to get called when a named "event" occurs like this …

```
NotificationCenter.default.addObserver(self,
                             selector: #selector(theKeyboardAppeared(_:)),
                                 name: Notification.Name.UIKeyboardDidShow,
                               object: view.window)
```

The event here is Notification.Name.UIKeyboardDidShow.
The object is the one who is causing the even to happen (our MVC's view's window).
func theKeyboardAppeared(_ notification: Notification) will get called when it happens.
The notification.userInfo is a Dictionary that will have details about the appearance.
Almost always the reaction to the keyboard appearing over your text field is to scroll it visible.
If the first responder is not in a scroll view, then position it so the keyboard never covers it.
UITableViewController listens for this & scrolls table automatically if a row has a UITextField.

UITextField

- Other UITextField properties

```
var clearsOnBeginEditing: Bool
var adjustsFontSizeToFitWidth: Bool
var minimumFontSize: CGFloat        // always set this if you set adjustsFontSizeToFitWidth
var placeholder: String?            // drawn in gray when text field is empty
var background/disabledBackground: UIImage?
var defaultTextAttributes: [String:Any] // applies to entire text
```

- Other UITextField functionality

  UITextFields have a "left" and "right" overlays.

  You can control in detail the layout of the text field (border, left/right view, clear button).

Demo Code

Download the demo code from today's lecture.