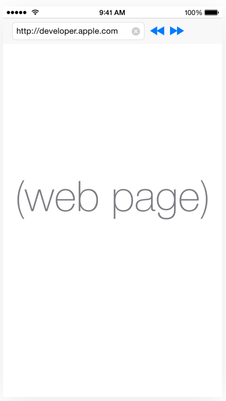
Project and Lesson Overview
Level 3: Frameworks and APIs

Construct numerous small apps that each illustrate different frameworks and iOS features. Teach students how to integrate maps, web content, sound, gestures and graphics.

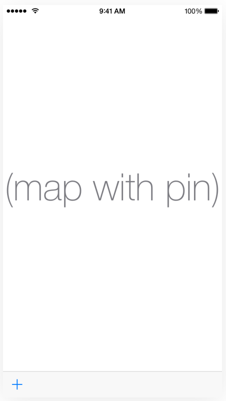
EasyBrowser (Five Lessons)



Build a simple web browser using WebKit-and UIWebView. Learning outcomes include:

- Defining URLs, requests, NSURL, NSURLRequest, and UIWebView
- Recognizing the UIApplication singleton
- Interpreting Interface Builder Auto Layout technology and constraints
- Combining constraints to ensure user interface usability on different screens
- Analyzing text-based input and different keyboard types
- Relating delegates, delegation and protocols to app implementation

Found (Five Lessons)



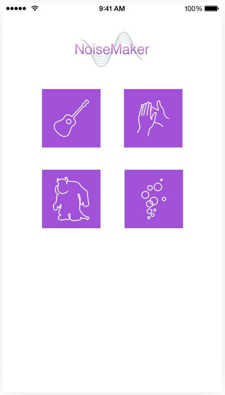
Build a map app using Map Kit and Core Location. Learning outcomes include:

- Describing how frameworks provide additional app functionality
- Defining URLs and the NSURL class
- Combining additional frameworks in an Xcode project configuration
- Relating delegates, delegation and protocols to app implementation
- Relating Swift structures and classes
- Discovering the fundamental features of the MKMapView API

NoiseMaker (Ten Lessons)

Play four different sounds using the AV Foundation framework. Learning outcomes include:

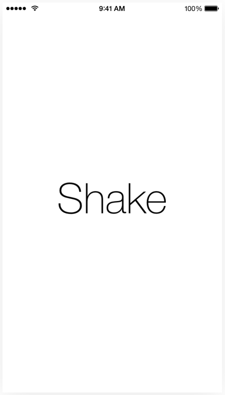
- Integrating multimedia assets into an Xcode project
- Discover how an AVAudioPlayer object can play sounds
- Recognizing asynchronous method calls
- Assess repetitive code and efficient object instantiation
- Apply initializers and initialization requirements
- Practice using arrays, for-in loops, and the map function



Gesturizer (Nine Lessons)

Recognize taps, double taps, pinches, rotations, swipes and shakes. Learning outcomes include:

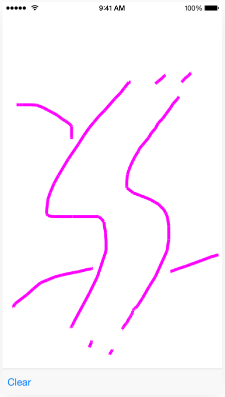
- Discovering how iOS encapsulates standard gesture recognition
- Combining view attributes and closures to exhibit visual effects
- Recognizing the Swift closure expression syntax
- Describing what closures are, and how they are invoked by other methods
- Exploring the features of the UIGestureRecognizer API
- Explaining the purpose of enumerations



FingerPainter (Six Lessons)

Create a drawing using the touch-based user interface of iOS, and Core Graphics. Learning outcomes include:

- Describing object-oriented inheritance and subclassing
- Discovering how to respond to touch events
- Analyzing and developing a drawing strategy using touch locations
- Discovering Core Graphics contexts and procedural drawing idioms
- Distinguishing the intents of UIView and UIImageView objects
- Practicing establishing connections between controllers and views


9:41 AM
100%
Clear