



Project Overview

You will make an iPhone app that will allow users to record their voice and will then modulate the recorded audio to sound like a Chipmunk or Darth Vader. This app will also let the user speed up or slow down the rate of playback, and experience fun echo and reverb effects.

Why this project?

The first step in becoming an iOS Developer is to create your first app! Pitch Perfect provides a substantial, while not overly complicated, introduction to the iOS ecosystem.

What will I learn?

You will learn how to:

- Create a new project from existing iOS app templates
- Add buttons, images, labels and create user interfaces for the app using the Storyboard
- Leverage the Model View Controller (MVC) design principles; this includes writing custom model code, and using UIViewController, and UIView
- Create Outlet and Action connections between the views and view controllers. Benefit from the power of Delegates by using them in code
- Navigate between two scenes in the app using UINavigationController
- Play audio files using AVFoundation's AVAudioEngine class
- Run the iOS simulator and download the app on a physical device
- Search documentation, resolve bugs, and read other people's code

Why is this project meaningful to my career?

MVC is the cornerstone of all iOS app development; understanding these principles early sets you on the path to success. Learning how to find answers to development questions by searching documentation emulates the practice of expert software engineers.

You will also make a relevant project that you will feel proud to share with your friends, family, and peers.

PROJECT SPECIFICATION

Pitch Perfect

Basic Functionality

CRITERIA	MEETS SPECIFICATIONS
The app contains two scenes of content: one for recording an audio file, and one for playing the audio with different effects.	The app contains two pages of content (one each for recording and playing audio), and uses <code>UINavigationController</code> to navigate between these two scenes.
All UI elements (buttons and text) are appropriately formatted for iPhone and iPad Portrait and Landscape layouts.	UI elements are appropriately positioned on the screen for iPhone and iPad portrait and landscape layouts.

Actions and Outlets

CRITERIA	MEETS SPECIFICATIONS
The app uses <code>IBAction</code> methods to record audio and playback sounds.	The app connects each button on the Storyboard to the correct <code>IBAction</code> method.
Labels and buttons are shown or hidden as appropriate.	In the first scene, the <code>Record</code> and <code>Stop</code> buttons are enabled and disabled appropriately. When no recording is taking place, the <code>Record</code> button is enabled and the <code>Stop</code> button is disabled. When recording is taking place, the <code>Record</code> button is disabled and the <code>Stop</code> button is enabled.

AVAudioRecorder

CRITERIA	MEETS SPECIFICATIONS
The first scene of the app uses <code>AVAudioRecorder</code> to record audio.	The app successfully uses <code>AVAudioRecorder</code> to record audio.

Delegates and Segues

CRITERIA	MEETS SPECIFICATIONS
The app uses the <code>audioRecorderDidFinishRecording()</code> method to determine when the audio has finished recording.	The app uses the delegate pattern and implements the <code>audioRecorderDidFinishRecording()</code> method.
The app programmatically triggers a segue from the first scene to the second.	The app does not use a Storyboard segue hardcoded to the <code>Stop</code> button. A segue from the first scene to the second is programmatically triggered via <code>performSegueWithIdentifier()</code> .

UINavigationController

CRITERIA	MEETS SPECIFICATIONS
The app allows users to re-record audio after a recording is complete.	The app allows the user to re-record by navigating back to the first scene from the second.

Sound Effects

CRITERIA	MEETS SPECIFICATIONS
All sound effects are present and play properly.	The second scene of the app contains the following buttons for audio effects: Snail (slow), Rabbit (fast), Chipmunk (high pitch), Darth Vader (low pitch), Echo and Reverb. All six buttons work properly to play the associated sounds.

Code Quality

CRITERIA	MEETS SPECIFICATIONS
Code is effectively abstracted.	Potentially repetitive blocks of code are effectively abstracted into reusable methods.
Code follows appropriate style.	Code adheres to Swift naming and style conventions .
Code is readable by others.	Code is readable and easy to follow. Any code that may be hard to understand is commented effectively.