UDACITY

Project 1: Pitch Perfect

iOS Developer Nanodegree

Criteria	Meets Specifications
Basic Functionality	
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 `UINavigationController` 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	
The app uses IBAction methods to record audio and playback sounds.	The app connects each button on the Storyboard to the correct `IBAction` method.
Labels and buttons are shown or hidden as appropriate.	In the first scene, the `Recording` label and the `Stop` button are disabled and enabled appropriately: When no recording is taking place the `Stop` button is disabled. While recording is taking place the `Stop` button is enabled and the `Record` button disabled
AVAudioRecorder	
The first scene of the app uses AVAudioRecorder to record audio.	The app successfully uses `AVAudioRecorder` to record audio.
Delegates and Segues	
The app uses the audioRecorderDidFinishRecording() method to determine when the audio has finished recording.	The app uses the delegate pattern and implements the `audioRecorderDidFinishRecording()` method.

The app programmatically triggers a segue from the first scene to the second by using the performSegueWithIdentifier() method.	The app does not use a Storyboard segue hardcoded to the `Stop` button. A segue from the first scene to the second is programmatically triggered via `performSegueWithIdentifier()`.
UINavigationController	
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	
The second scene of the app contains the following audio effects: Snail (slow), Rabbit (fast), Chipmunk (high pitch), Darth Vader (low pitch), Echo and Reverb.	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	
Code is effectively abstracted.	Potentially repetitive blocks of code are effectively abstracted into reusable methods.
Code adheres to Swift naming and style conventions.	Code adheres to Swift naming and style conventions.
Code uses appropriate and effective comments.	Code is readable and easy to follow. Any code that may be hard to understand is commented effectively.

Once you have a functioning app, consider adding more features to make your app stand out! Here are some suggestions:

- **Use UIStackViews for the RecordSoundsViewController view too. You already use UIStackViews in the PlaySoundsViewController, try adding them to the RecordSoundsViewController.**
- **Add a UILabel to the audio playback view that shows the duration of the recorded audio. You'll need to read up on Apple's documentation for AVAudioPlayer to figure out how to do this.**