# AV Storybook

We're going to build a photo storytelling app. The app will prompt a user to import photos, and record audio to accompany those photos.

We'll be using AVFoundation to record and playback audio, UIImagePickerController to import photos, and a UIPageViewController to flip between pages.

## Goals

- The app opens to a blank view with five blank "pages"
- In each view there is:
    - an image view
    - a camera button
    - a microphone button
- Tapping the microphone button starts recording audio. Tapping it while your audio is already recording will stop recording.
- Tapping the camera button prompts the user to take a photo or pick one from the camera roll.
- Tapping the image view plays the audio.
- Swiping left and right moves between the pages. each one displays a different photo and plays a different audio clip.

## Tools + Tips

- Start with the single view controller template.

- Use a UITapGestureRecognizer to detect taps on the image view. Note: by default, `UIImageView`has it's `userInteractionEnabled` property set to `NO`, enable this to allow touch events to be detected.
- Use AVAudioPlayer for audio playback. Keep a strong reference to the player or it won't play and you can't stop it when you turn the page.
- Use viewWillDisappear to stop audio when you turn the page.

# Step overview

- 1. Create a model class to store the data for a given page. Think about what properties it will need.
- 2. Create the capture view controller. This will have the image view, the audio recording, and the playback button. It will fill in the model object using this data.
- 3. Add pagination to the app.

# Step 1 - Model

Create a model class. Add properties for everything you need to store on a per-page basis.

# Step 2 - Input

Create a `StoryPartViewController`. Each of your pages will be an instance of this class.

Add a `StoryPartViewController` to your storyboard, and start setting up the image view, and buttons. Connect outlets for everything on screen, and

actions for the buttons. You can set up the tap gesture recognizer in the storyboard, or in code.

Your part controller class should have one `AVAudioPlayer`, and one `AVAudioRecorder`. Think about where you're recording your data to, and how that will scale if you have five view controllers doing the same thing.

As your user picks a photo, and records audio, set properties on an instance of the model class you created in step 1. For now, we just create the instance in `viewDidLoad`, but in Step 3, we will create and store these instances outside of the StoryPartViewController, and pass them in as needed.

# Step 3 - Paginate

Make a new `UIPageViewController` subclass, this will store the model objects, and handle creating new `StoryPartViewControllers` as the user swipes through the story.

Add a `UIPageViewController` to your storyboard, and set it as the inital view controller. Change it's class to the subclass you just created.

In your storybook view controller class' `viewDidLoad` method set yourself as your own `dataSource`and `delegate`. This is an achitectural choice... we could easily have another object act as the `dataSource/ delegate` instead of subclassing `UIPageViewController`.

# Create an array of model objects

Add an array property to your UIPageViewController subclass, and add five empty model objects to it in `viewDidLoad`.

# Implement `UIPageViewControllerDataSource`

The page view controller will request the next and previous view controller if a user tries to page left or right. You can provide them by implementing the two required methods in `UIPageViewControllerDataSource`. For now, you can have them return `nil` to indicate there is no page to the left or the right of the initial one.

# Supply an initial view controller

While supplying the next and previous view controllers via the dataSource protocol handles paging to the left and right, you still need to set an initial story part view controller.

We don't just want to `alloc` and `init` a new `StoryPartViewController` here, as we also want all the storyboard setup we did earlier. We need to instantiate a view controller using our storybaord.

Once we have an instance of our story page controller, we need to pass it the model object that represents the first page of our storybook, and then present the controller using `setViewControllers:direction:animated:completion:`.

Now that we know how to instantiate a view controller from a storyboard, go back to the two `UIPageViewControllerDataSource` methods and have them return a `StoryPartViewController`instead of `nil`. Be sure to pass it the correct model object before returning it (you can look its index up in the array of model objects).

# Stretch goals

- Add a way to add pages dynamically, then start with just one page instead of five.
- Add a way to delete pages.
- Disable the play button unless a recording has already been made.
- Disable the record button while audio is playing.