Camera

By: Katie Kuenster, Jack Larson, and Max Walsh

History of iPhone Camera

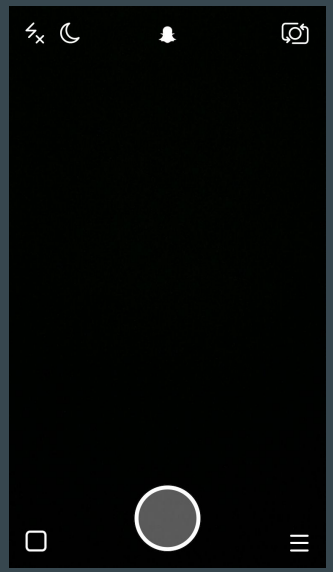
- iPhones have always had a rear camera
- iPhone 4 was the first phone to have a front facing camera
- Two ways to handle the camera in your app
  - 1. UIImagePickerController (iOS 2.0 onwards)
  - 2. AVFoundation (iOS 4.0 onwards)
- Numerous apps utilize the camera

Contemporary Examples

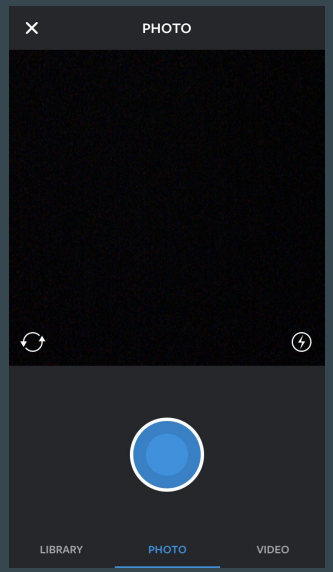
- UIImagePickerController class used in many popular apps
  - GroupMe, LinkedIn, Pinterest, WhatsApp, many photo editing apps
- Most have small customizations

Alternate Implementations

- AVFoundation Framework
  - Primarily Objective-C
  - Allows you to fully customize the UI and its functionality
  - All features are manually implemented
- Examples
  - Snapchat, Instagram



Snapchat

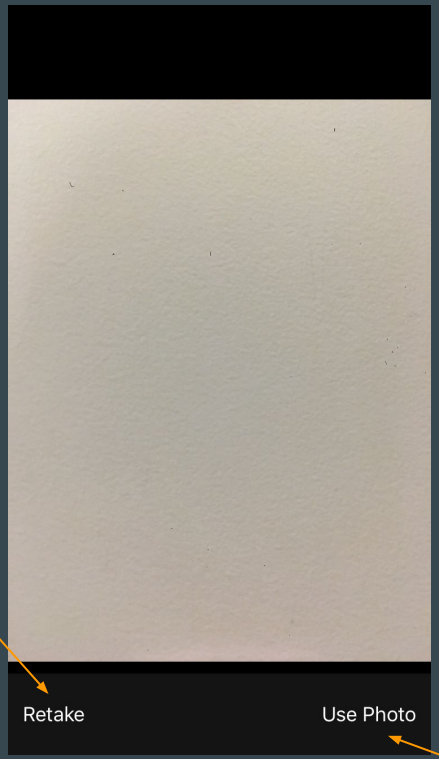


Instagram

Library Overview

- **UIImagePickerController**
  - Older, easier framework
  - Standard Apple interface
  - Utilizes both front and rear camera
  - Requires little code to run
  - Can check whether device has a camera
  - Tap to focus functionality
  - Image persists in camera roll
  - Equal performance but less flexible than AVFoundation
- **MobileCoreServices**
  - Allows use of kUTTypeImage and kUTTypeMovie data types

User Interface



flash

switch camera views

tap to focus

choose media type

exits

tap to capture image

return to taking picture

exits and sets UIImagePickerOriginalImage

UIImagePickerController Details

- Manages the user interface for pictures and movies
- Allows user to customize what types of media are allowed (e.g. picture, video)
- Automatically adjusts if the device only has one camera
- To implement default controls:
  - 1. Verify that the device is capable of picking content
  - 2. Check which media types are available
  - 3. Adjust UI according to the media types you want to make available
  - 4. Display the User Interface
  - 5. When media is selected, dismiss the image picker

Camera Roll - Persistence

- Saving to the camera roll
  - In imagePickerController function set image = UIImagePickerControllerOriginalImage
  - UIImageWriteToSavedPhotosAlbum(image, self, "image:didFinishSavingWithError: contextInfo:", nil)
- Accessing the camera roll
  - Check for source:
    - UIImagePickerControllerSourceType.SavedPhotosAlbum
  - Set the imagePickerController's sourcetype:
    - imagePicker.sourceType = UIImagePickerControllerSourceType.PhotoLibrary
  - Display camera roll:
    - self.presentViewController(imagePicker, animated: true, completion: nil)

Editing images - Overlays

- Give image: colored tint:
  - Set rendering mode for imageView.image
  - imageView.tintColor = UIColor.redColor()
- Edit opacity of image:
  - Edit the alpha property of an imageView
  - imageView.alpha = 0.5
- Lay UIImageViews on top of one another (in ViewController)
  - Edit opacity of images to view all imageView