Project Overview

With this first version of MemeMe, students will create an app that enables a user to take a picture, and add text at the top and bottom to form a meme. The user will be able to share the photo on Facebook and Twitter and also by SMS or email.

Why this project?

MemeMe covers many of the major UI components found in essentially every app. It combines them with the camera and social media activity view to create a fun and personal user experience.

What will I learn?

- Access the Camera and Photo Album using the UIImagePickerController
- Understand how Swift optionals, closures, collections, classes, structs and protocols are used in iOS apps
- Create Actions and Outlets with extremely high proficiency
- Use a UIActivityViewController to share media with Facebook, Twitter, SMS, and Email
- Describe the delegate pattern and give a full account of its importance and uses in iOS classes

Why is this project meaningful to my career?

- Becoming comfortable using a variety of UIControls and stock view controllers is an essential step in becoming an iOS developer.
- Understanding UIKit allows a developer to make educated assumptions about the architecture of apps that they use.
- Mastering UKit fundamentals frees a developer up to learn model-oriented skills like networking and persistence.

NEXT

App Specification: MemeMe 1.0, Meme Editor

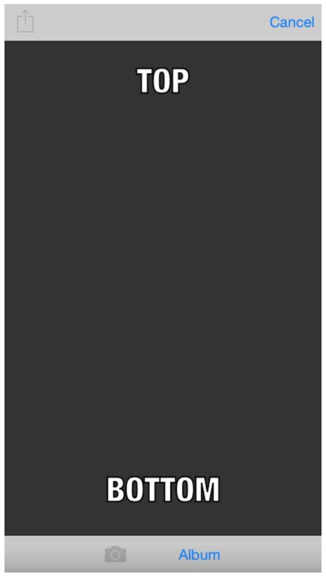
iOS Developer Nanodegree

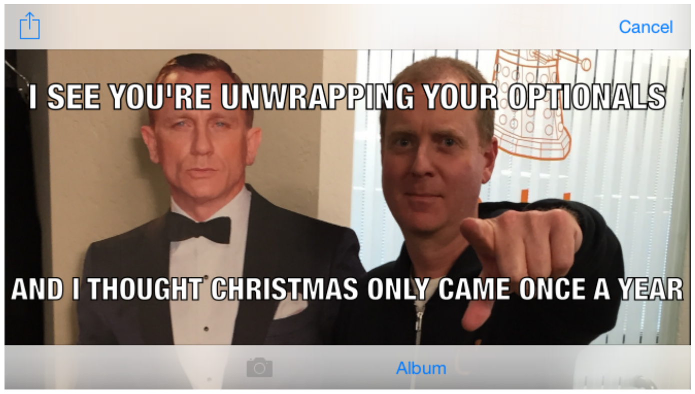
[Note that this is an informal app description. It will give you an idea how the app should work, but when you submit your app it will be rated based on the Rubric.]

MemeMe 1.0 is a meme-generating app that enables a user to attach a caption to a picture from their phone. After adding text to an image chosen from the Photo Album or Camera, the user can share it with friends.

Meme Editor View

The **Meme Editor View** consists of an image view overlaid by two text fields, one near the top and one near the bottom of the image. This view has a bottom toolbar with two buttons: one for the camera and one for the photo album. The top navigation bar has a share button on the left displaying Apple's stock share icon and a "Cancel" button on the right.




Cancel
I SEE YOU'RE UNWRAPPING YOUR OPTIONALS
AND I THOUGHT CHRISTMAS ONLY CAME ONCE A YEAR
Album

User Flow

When the user first launches the app the **Meme Editor View** will appear.

In the **Meme Editor View**, when the user clicks on the "Album" button, an Image Picker is presented, making it possible to choose an image from the Photo Album. If there is a camera available on the device, pressing the camera button launches the camera, and a newly snapped photo can be chosen for the meme. If a camera is not available on the device, the camera button is disabled.

After an image is chosen, the image picker is dismissed, allowing text to be entered into the top and bottom text fields of the editor. When a user clicks inside one of the text fields, the default text disappears and the keyboard slides up. When the user finishes entering text and presses return, the keyboard is dismissed and the new meme is displayed.

When the user presses the "Cancel" button, the **Meme Editor View** returns to its launch state, displaying no image and default text.

When the user presses the share button, Apple's stock Activity View appears, displaying several options for sharing the meme. After an option is chosen, the Activity View is dismissed and the **Meme Editor View** is visible again.

UDACITY

PROJECT SPECIFICATION

MemeMe 1.0: Meme Editor

General

CRITERIA	MEETS SPECIFICATIONS
Code uses control flow effectively.	Control statements (`for` , `while` , `if` , `else` , `switch`) are used appropriately.
Code is effectively abstracted.	Potentially repetitive blocks of code are effectively abstracted into reusable methods.
Code inputs and outputs are appropriate.	Arguments and return values are procedure-appropriate.
Code is readable by others.	Code is easy to understand. Any code that may be hard to follow is commented effectively.
App content is properly displayed in both portrait and landscape mode.	User interface elements, images, and generated memes are properly displayed and function as expected in portrait as well as landscape orientations.
Code follows appropriate style.	Code adheres to Swift naming and style conventions.

Meme Editor

CRITERIA	MEETS SPECIFICATIONS
Buttons on the MemeEditor toolbar are correct.	The Meme Editor view has a toolbar with two buttons: one that allows user to open the Photo Album to select an image, one that launches the camera.
The Camera button is enabled correctly.	The app displays the camera when the Camera button is pressed on a phone.
The Camera button is disabled correctly.	The Camera button is disabled when app is run on devices without a camera, such as the simulator.

CRITERIA	MEETS SPECIFICATIONS
Images are displayed properly.	The chosen image from the camera or the photo album is displayed and scaled properly with `AspectFit` to fit the device screen.
The Album button is displayed correctly.	The app displays the image picker when the Album button is pressed.
The necessary text fields are provided.	Text fields are provided for top and bottom text.
Font in text fields is easy to read.	The font and style used to display the meme text is easy to read: bold, all caps, white with a black outline, and shrink to fit.
The text fields are properly shown while typing.	The text field that the user is currently editing remains fully visible with the keyboard on screen. For the bottom text field this is achieved by moving the entire view up to keep the text field on screen, then back after the keyboard is dismissed.
There is a social share button.	The app has a social share button that uses the “Action” icon built into iOS.
The share button works as intended.	The share button launches the Activity View.
When the share action is complete the meme is saved.	The meme is saved before the Activity View Controller is dismissed.
A `Meme model` groups the properties of a single meme.	The `Meme model` is a `struct` that includes: • two `strings` representing the top and bottom text • the original image • a memed image combining image and text

Suggestions to Make Your Project Stand Out!

- Have your app use the `Impact` font by default.
- Let the user choose between different fonts for the meme text.
- Allow the user to crop the image.
- Use your creativity to improve the UI so your app displays exceptional graphic design.