General Instructions: Put the entire app directory into one zip file and submit as an attachment under Content → Homework 8 for this course on the Blackboard Learn system by the above deadline. Note that you may submit multiple times, but only the most recent entry submitted before the above deadline will be graded.

For this homework you will build an app that recognizes two gestures: smile and frown. This could be used to rate food items at some point, but for now, we will implement it as a standalone app. See screenshots below. Specifically,

1. Create a single-view app with the main view controller embedded in a navigation controller. The main view nav bar should have the title “Review” and the prompt “Food App”. Centered at the bottom of the view should be a button titled “Clear”. Above the button should be an image view, and above the image view should be a label.

2. Create a custom gesture recognizer called “SmileGestureRecognizer”. This gesture recognizer should recognize a single-touch interaction that draws a smile shape. Specifically, after the first touch, the movement should be down and to the right, but at some point change to up and to the right. Once the movement changes to up-and-to-the-right, it must continue in this way. The final point must be at least 100 pixels away from the initial point. Violation of any of the above constraints at any time during the gesture should result in a failed gesture. The initial point, final point, and points in between should all be drawn on the view using the red drawBox approach presented in class.

3. Create a custom gesture recognizer called “FrownGestureRecognizer”. This gesture recognizer should recognize a single-touch interaction that draws a frown shape. Specifically, after the first touch, the movement should be up and to the right, but at some point change to down and to the right. Once the movement changes to down-and-to-the-right, it must continue in this way. The final point must be at least 100 pixels away from the initial point. Violation of any of the above constraints at any time during the gesture should result in a failed gesture. The initial point, final point, and points in between should all be drawn on the view using the red drawBox approach presented in class.

4. When the smile gesture is detected, the label should appear and say “Tastes Good!”, and the image should appear as a smiley face. When the frown gesture is detected, the label should
appear and say “Tastes Bad!”", and the image should appear as a yucky face. The label and image should not be visible when the app starts.

5. Tapping the Clear button should remove any red boxes from the view and hide the label and image.

6. To ensure that both gestures can be detected, you will need to use the gestureRecognizer:shouldRecognizeSimultaneouslyWith method of the UIGestureRecognizerDelegate (see slides 18-19 of Gestures lecture). To ensure that the button taps are not redirected to the gestures, you will need to use the gestureRecognizer:shouldReceive method of the UIGestureRecognizerDelegate (see slide 20 of Gestures lecture).

7. Be sure that auto layout constraints are set so that all elements in the view are appropriately displayed (i.e., no overlap, occlusion, or cutoff at edges) regardless of device orientation.

8. Your app should run in Xcode 11 using Swift 5 and iOS 13. Your app will be tested using Xcode’s iPhone 8 device simulator, so make sure your layout constraints work for that device.

Storyboard:
Simulator:

App start or
Or after Clear:

Smile detected:

Frown detected:

Failed detection:

Smile detected (landscape):

Frown detected (landscape):