Washington State University
School of Electrical Engineering and Computer Science
Spring 2021

CptS 479 Mobile Application Development
Homework 8
Due: March 19, 2021 (11:59pm)[1]

General Instructions: Put the entire app directory into one zip file and submit as an attachment under Content → Assignments → Homework 8 for this course on Blackboard Learn 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.

This homework will be a standalone version of our HealthApp that tests the user's ability to trace a circle using a custom gesture. See screenshots below. Specifically,

1. Create a new Xcode project called HealthApp that has a single view embedded in a navigation controller with title "Circle Test" and prompt "Health App". Centered at the top of the view is a two-line label containing "Trace a circle in between the two circles below."
2. In the center of the view draw two concentric blue circles, one with radius 50 and one with radius 150. You can use the code snippet below to draw a circle. You can access the center point of the view using "self.view.center".

```
func drawCircle(center: CGPoint, radius: Float) {
      let shapeLayer = CAShapeLayer()
      let circlePath = UIBezierPath(arcCenter: center,
            radius: CGFloat(radius),
            startAngle: 0, endAngle: (2 * CGFloat.pi), clockwise: true)
      shapeLayer.path = circlePath.cgPath
      shapeLayer.fillColor = UIColor.clear.cgColor
      shapeLayer.strokeColor = UIColor.blue.cgColor
      self.view.layer.addSublayer(shapeLayer)
}
```

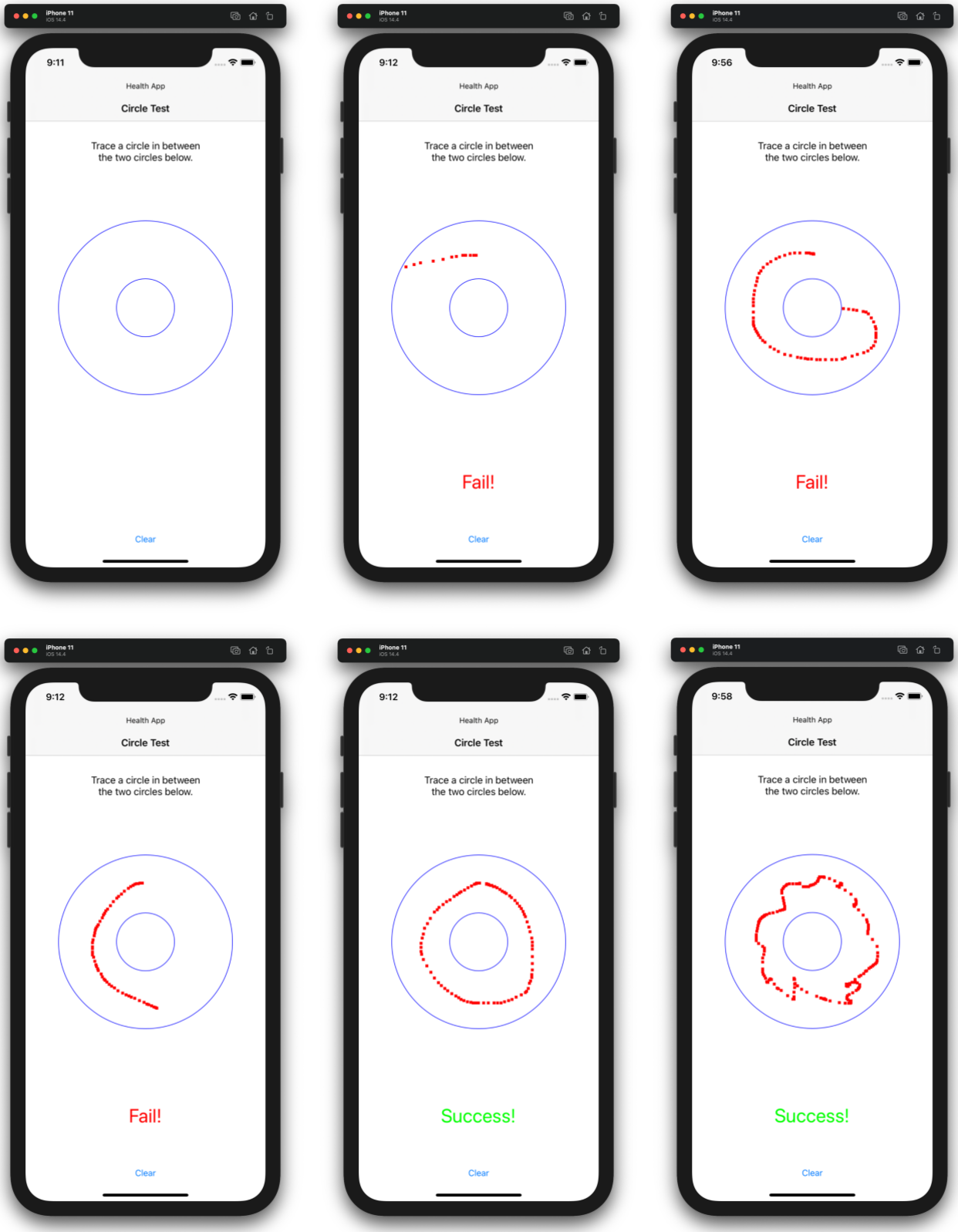
3. Centered at the bottom of the view, add a "Clear" button, which will be used to clear the red boxes that trace your gesture (see #6 below). Just above the button, center a label that will display "Success!" in green or "Fail!" in red after each attempt to trace the circle. This label should be hidden initially, hidden as soon as you begin a trace, and hidden if the Clear button is tapped.

[1] The university has asked us to not have assignments due on March 17-18, because March 17 is a school holiday. So, HW8 is due on March 19. But note that HW9 will still be due on March 24.

4. Implement a custom gesture class called CircleGestureRecognizer. This gesture should have three properties: minRadius, maxRadius, and center. The general behavior of the gesture is to recognize a complete circle in which the distance D of each point from the center satisfies (minRadius ≤ D ≤ maxRadius). As soon as this distance constraint is violated, the gesture should fail. If after the user completes the trace, the circle is not complete, then the gesture should fail. The five methods of your CircleGestureRecognizer class should function as follows:
    a. touchesBegan: Check that the first point in touches satisfies the distance requirement. If not, set the state to .failed, else retain the point and set the state to .began.
    b. touchesMoved: Check that the first point in touches satisfies the distance requirement. If not, set the state to .failed, else retain the point and set the state to .changed.
    c. touchesEnded: Check that the first point in touches satisfies the distance requirement. If not, set the state to .failed, else check that the circle is complete by checking that there is at least one point in each of the four quadrants of the circle. If not, then set the state to .failed, else set the state to .ended. For example, a point is in the first quadrant if (point.x > center.x) and (point.y < center.y).
    d. touchesCancelled: Set the state to .cancelled.
    e. reset: Clear your array of retained points.
5. In the view controller, create an instance of the CircleGestureRecognizer, set the center to the view's center, set minRadius to 50, and set maxRadius to 150. Add the gesture to the view.
6. In the action that is called by the circle gesture, first extract the point and the state of the sender. If the state is .began, then draw a small red box[2] at the point and ensure the label is hidden. If the state is .changed, then draw a small red box at the point. If the state is .ended, then draw a small red box at the point and display "Success!" in the label. If the state is .cancelled, then display "Fail!" in the label.
7. Test your app using the iPhone 11 simulator, which is the same simulator we will use to grade your app.
8. You may assume the device will always be in portrait mode for this app.

[2] See the lecture notes for code to draw and clear small red boxes in the view.

Simulator:


iPhone 11
iOS 14.4
9:11
Health App
Circle Test
Trace a circle in between the two circles below.
Clear
9:12
Health App
Circle Test
Trace a circle in between the two circles below.
Fail!
Clear
9:56
Health App
Circle Test
Trace a circle in between the two circles below.
Fail!
Clear
9:12
Health App
Circle Test
Trace a circle in between the two circles below.
Fail!
Clear
9:12
Health App
Circle Test
Trace a circle in between the two circles below.
Success!
Clear
9:58
Health App
Circle Test
Trace a circle in between the two circles below.
Success!
Clear