**Project: Mimic Me!**

Rob Straker – August 24, 2017

1. **Display Feature Points**

The first task was to display the feature points on top of the webcam image that are returned along with the metrics. To do this, I edited mimic.js, writing code for the drawFeaturePoints() function:

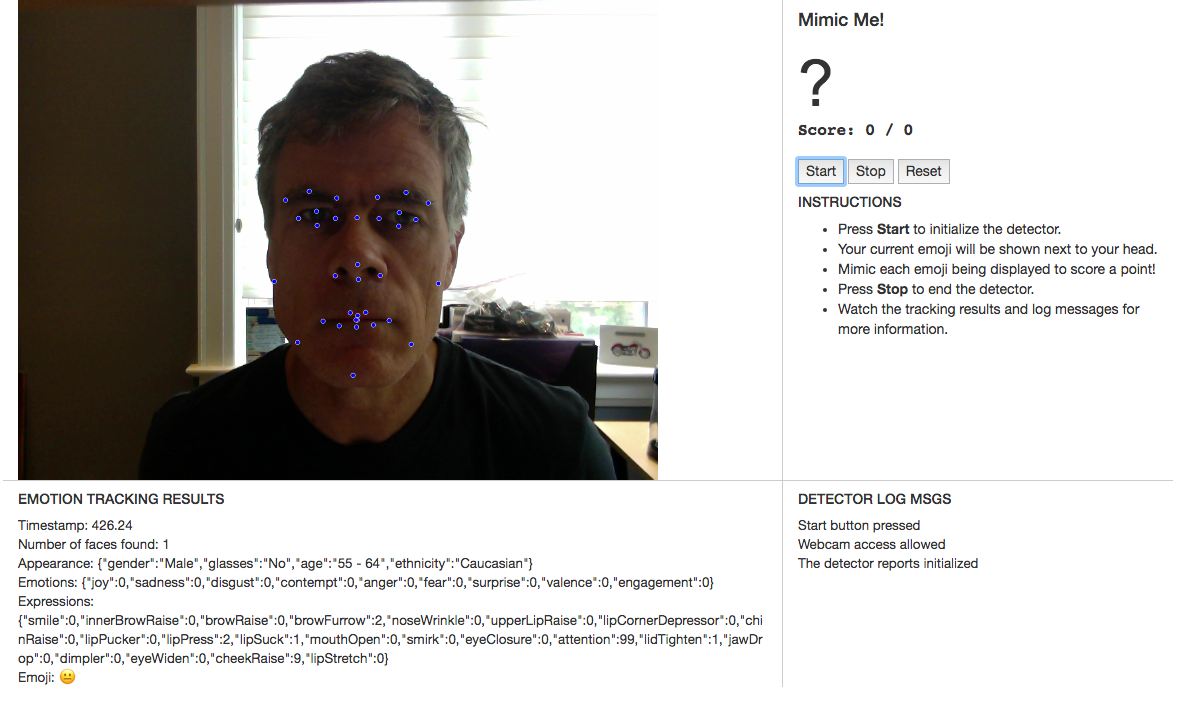
**function** **drawFeaturePoints**(canvas, img, face) {

...

}

Using the Mozilla Foundation website as a reference, I set the stroke and fill style, then looped over all featurePoints in the face to draw a feature point, where I drew circles using the ctx.arc command

Following is a screen shot of my output:



### Show Dominant Emoji

The second task was to draw the *dominant emoji* next to the face, which the Affectiva API reports as the emoji that best represents the current emotional state of a face. I edited mimic.js, developing code for the drawEmoji() function:

function drawEmoji(canvas, img, face) {

...

}

Using the Mozilla Foundation website as a reference, I selected a font and used ctx.fillText, accessing the dominant emoji as face.emojis.dominantEmoji. To ensure the emoji was located close to the face, I used one of the feature points as an anchor point.

Following is a screen print of my output:



### Implement Mimic Me!

The final task was to implement the game mechanics. I edited mimic.js, creating several additional functions, but did not modify the HTML or CSS files to change the look and feel of the game. New functions included:

*initGame* – initializes the game variables.

*selectEmoji* – select emoji from list of 13 recognized by Affectiva’s SDK.

*playGame* – determine whether there is a match to the target emoji.

In this game, the computer displays a random emoji, and the human player attempts to mimic that emoji as best as they can. As soon as they are able to mimic the face (or after a timeout of 10 seconds), the game moves on to the next random emoji. The game stops after 10 emojis are shown.

The score is based on the number of emojis the player is able to mimic within the timeout for each. Some feedback messages are displayed whenever the player is able to mimic an emoji successfully. Following is a screen shot of my output:

