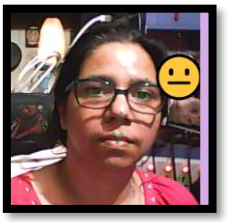
Project Mimic Me

Display Feature Points:

The feature points (key points) identifying specific areas of face like chin, jaw, lips, eye brows etc are identified by the affective API. They are available to us via: **face.featurePoints**. To display features, I declared a **canvas stroke** of green colour. A green **colour circle of 2 radius** was drawn on the video image using the **arc method**. There are three keypoints for each eye brow, four each for both eyes and nose. The two lips are covered by nine keypoints. There one each for ears edge, and three cover the chin area.

Dominant Emoji:



The variable **face.emojis.dominantEmoji** from Affectiva API represents the dominant emotion (and the corresponding Emoji). I added content in the given **drawEmoji f**unction to display the emoji as a canvas object at a **location near** one of the **face.featurepoints**, so that emoji is near the face of the person.

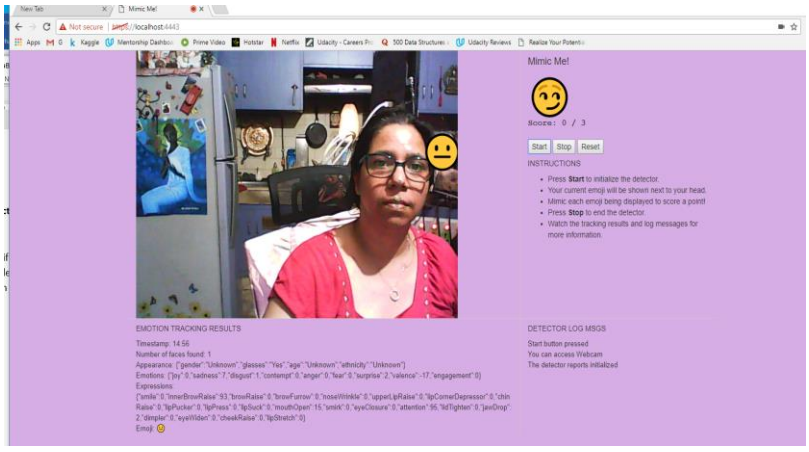
Show Random emoji to mimic:

For generating a random emoji, I wrote a function **randomEmojiCode**, it selects one the 13 emojis detected by affective API. The function first generates a random number between 0 and 12, the number is used as index to refer the **Unicode** of the respective emoji. The random emoji stays for **5000 milliseconds** so that the player has sufficient time to mimic it.

Match with current player expression:

To implement this, I wrote another function **compareWithFace**, it takes two arguments, one the Unicode of the **target emoji** to be mimicked, and other the face object as detected by the Affectiva API. The function returns True if the **dominantEmoji** of the face (emojis) is same as the target emoji, otherwise it returns false. To make the comparison I need to use the **toUniCode** function provided by Udacity wrapper for the project.

Reset and shows a new emoji



After every 5000 milliseconds the function **randomEmojiCode** generates a new target emoji. The process is repeated for 20 target emojis, after which the game decides if you are a winner or need improvement. It displays the result, and then reloads a new game. I could have kept it increasing, but that seemed boring.