

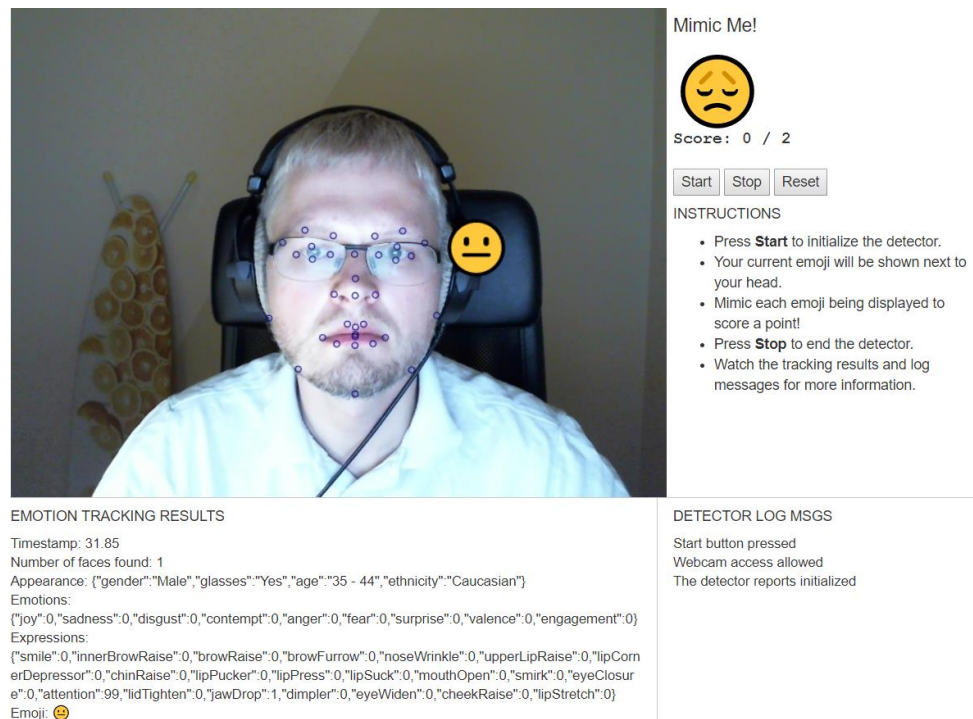
Project Mimic Me!

1. Display Feature Points

To display feature points:

- we have to extract x and y coordinates (featurePoint.x, featurePoint.y respectively),
- define their size and color. I use 'navy' color for stroke.
- I don't fill them because in this case the picture seems to be cluttered.

The result of this work is below:



Mimic Me!

Score: 0 / 2

Start Stop Reset

INSTRUCTIONS

- Press **Start** to initialize the detector.
- Your current emoji will be shown next to your head.
- Mimic each emoji being displayed to score a point!
- Press **Stop** to end the detector.
- Watch the tracking results and log messages for more information.

EMOTION TRACKING RESULTS

Timestamp: 31.85
Number of faces found: 1
Appearance: {"gender": "Male", "glasses": "Yes", "age": "35 - 44", "ethnicity": "Caucasian"}
Emotions: {"joy": 0, "sadness": 0, "disgust": 0, "contempt": 0, "anger": 0, "fear": 0, "surprise": 0, "valence": 0, "engagement": 0}
Expressions: {"smile": 0, "innerBrowRaise": 0, "browRaise": 0, "browFurrow": 0, "noseWrinkle": 0, "upperLipRaise": 0, "lipCornerDepressor": 0, "chinRaise": 0, "lipPucker": 0, "lipPress": 0, "lipSuck": 0, "mouthOpen": 0, "smirk": 0, "eyeClosure": 0, "attention": 99, "lidTighten": 0, "jawDrop": 1, "dimpler": 0, "eyeWiden": 0, "cheekRaise": 0, "lipStretch": 0}
Emoji: 😞

DETECTOR LOG MSGS

Start button pressed
Webcam access allowed
The detector reports initialized

2. Show Dominant Emoji

To show dominant emoji I realized the following steps:

- After some experiments I found optimal size for emoji – 50px (ctx.font = '50px Arial')
- To stick the emoji to my face I used particular feature points as an anchor.
 - I made a simple cycle to pass through all feature points and find points with max/min X and Y coordinates.
 - Then I take point in the right upper corner add small margin (to avoid face overlapping) and place to this coordinate the emoji
 - To find dominant emoji, I used *face.emojis.dominantEmoji* value
- You can find the result on the picture above

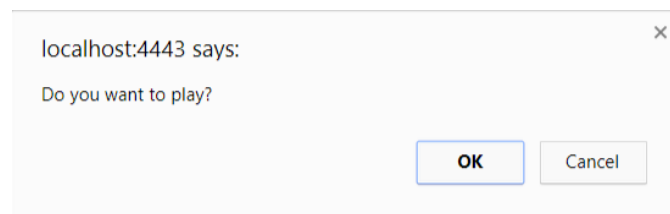
3. 3. Implement Mimic Me!

First of all, I defined main variables:

Variable name	Description
randomEmoji	Emoji we need to mimic (target)
correct	Total number of correct answers
total	Total number of played rounds
match	Boolean to check if we can mimic target emoji
maxTimeToGuess	Number of seconds to replicate an emotion
numbersOfRounds	Number of rounds to replicate an emotion
curDate	Date when round begins
stDate	Date after round began to check if we are within limit

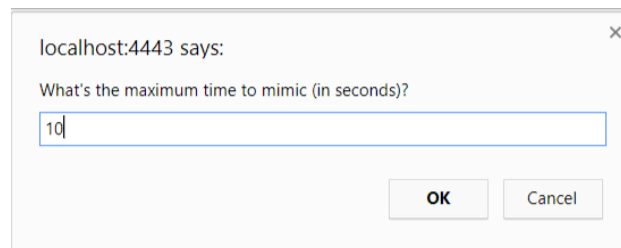
The logic of game is the following:

- When the system is initialized or we begin new game the function `newGame()` is activated. We have to confirm that we want to game:

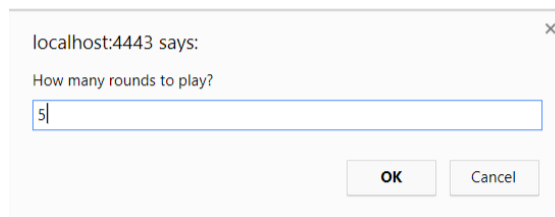


A dialog box titled 'localhost:4443 says:' with the text 'Do you want to play?'. It has two buttons: 'OK' and 'Cancel'.

- If so, we have to initialize key parameters for game (max time play for each round and number of rounds:



A dialog box titled 'localhost:4443 says:' with the text 'What's the maximum time to mimic (in seconds)?'. It has an input field containing '10' and two buttons: 'OK' and 'Cancel'.



A dialog box titled 'localhost:4443 says:' with the text 'How many rounds to play?'. It has an input field containing '5' and two buttons: 'OK' and 'Cancel'.

- After this, we begin to play. If we mimic target emotion (it is selected randomly from the list of possible emotions), the next round begins.
- If we cannot mimic target emotion, we get the warning and the next rounds begins:

