AIND - Mimic Game Report

Part 1 - Display feature points

```
for (var id in face.featurePoints) {
    var featurePoint = face.featurePoints[id];
    ctx.beginPath();
    ctx.arc(featurePoint.x, featurePoint.y, 2, 0, 2 * Math.PI);
    ctx.stroke();
    ctx.fill();
}
```

This is pretty straightforward - to draw the feature points I just iterate through the returned featurePoints and draw a circle at each position

Part 2 - Dominant emoji

```
ctx.font = '15px serif';
ctx.strokeStyle = 'yellow';
ctx.fillText(face.emojis.dominantEmoji, face.featurePoints[2].x, face.featurePoints[2].y);
```

First, I set the style for the emoji, then use fillText() function to display the unicode-encoded emoji at featurePoints[2] which lies near the chin and therefore makes it look less distracting.

Part 3 - Mimic game

Show random emoji to mimic

```
function nextTarget(){
  targetEmoji = emojis[Math.floor(Math.random() * (emojis.length - 1))];
  setTargetEmoji(targetEmoji);
  total += 1;
  setScore(correct, total);
  setTimeout(nextTarget, mimicTimeout);
}
```

This function selects a random emoji from the provided list and set it as current target, and also increase the total number of emoji challenges. It also update the scoreboard using the current number of correct vs total emojis and set a time out (10s) until next emoji challenge.

Match with current player expression, reset and shows a new emoji

```
function judge(dominantEmoji, targetEmoji){
  if (toUnicode(dominantEmoji)===targetEmoji){
    correct += 1;
    nextTarget();
  }
  setScore(correct, total);
}
```

This function is invoked when the onImageResultsSuccess event fires and it checks if player's facial expression matches with the emoji. If yes, total correct number are updated and next challenge is released immediately, alongside with updated scoreboard.

Screenshot