Kenneth Slaby and Natalie Gilbert
kslaby01 ngilbe01
Kenneth.Slaby@tufts.edu
Natalie.Gilbert@tufts.edu

COMP 86 -- ASSIGNMENT 6

Project Proposal:

A simple game that combines an emotion controlled interface with simple eye tracking as its input methods. The player uses their laptop webcam to play the game, which using the WebGazer API keeps track of their eye movements, and using Affectiva's emotion reading API determines if they have matched one of the given emotions on screen. The more emotions that are matched, the more points the user wins!

Initial Design:

A simple HTTP site, which upon giving permissions to access the webcam starts a simple game using the two aforementioned API's. At runtime, the use of webcam takes its input, and gives it to both the API's at once, allowing it to process the information simultaneously. We will be using the emotions included in the Affective Demo, thus several emotions might be classified to one particular base; ex. Happy = Blissful; Anger = Frustration.

High Level Framework:

Canvas:

- Score (int)
- Incoming Emotions:
    - Cubes:
        - JPEGS of Emotions as textures
        - Array of Classifications of Emotions per Cube
        - Position (x, y, z)
    - Destruction Feedback
        - JPEG of an explosion (played upon death of a Cube)

Legend

- JPEGS of Emotions
- Labels with possible equivalent emotions

Start/Stop Widgets

- JButtons with callbacks to control timer for drawing

Technological Requirements:

We will use 3JS to draw our Canvas and our Cubes, which we haven’t had much practice with thus far.

Additionally, we will be using the Affectiva emotion API, as well as the WebGazer eye tracking API. Both of which are technologies we have not used before, but believe to have sufficient knowledge from provided examples to understand how to combine them to fit the needs of our project.

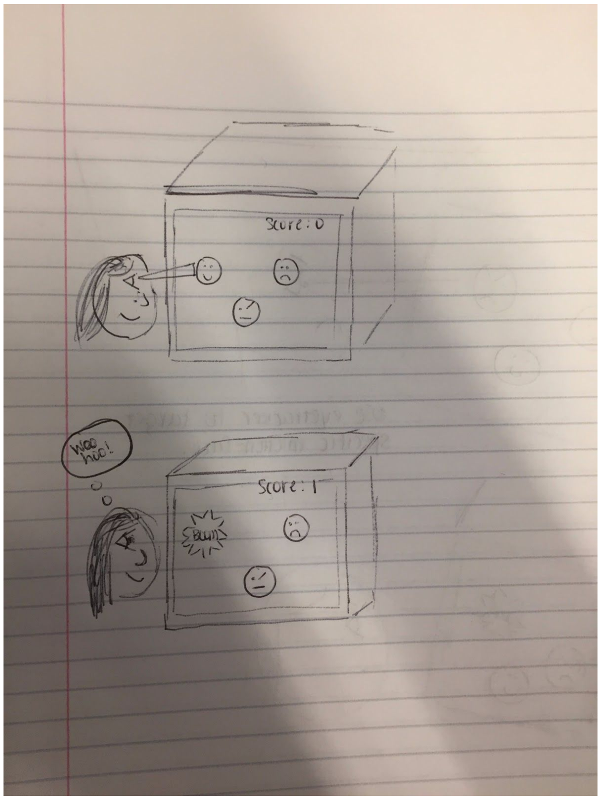
Thus, we will create some initial tests of both Affective and WebGazer to determine how each individually can be used for our project. Once we confirm that each can be used on their own, we will then determine how to combine them together. As for 3JS, we believe that we will get sufficient practice for those methods doing Assignment 7.

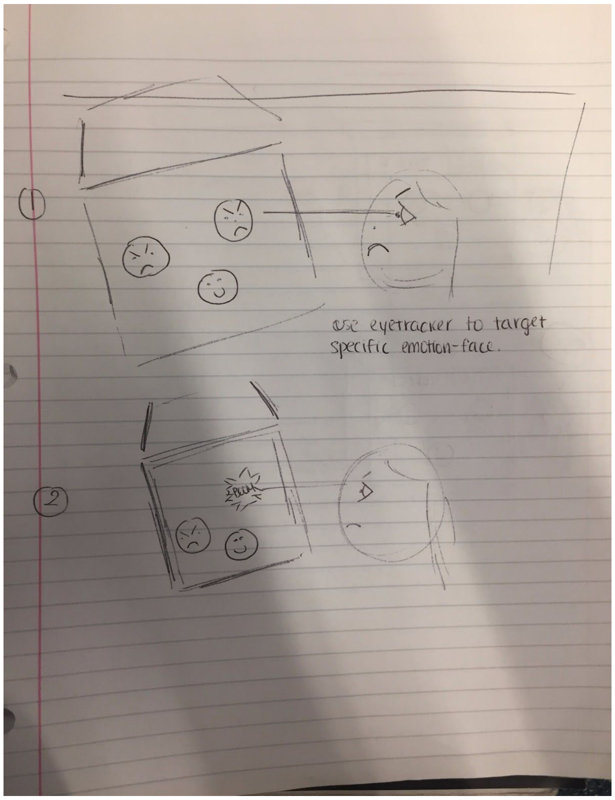
Feasibility Test

We will create a simple test program (which should be included with the Assignment submission), which hopefully proves that both of these API’s can be activated at the same time. Under the circumstances that this is not the case, we will make our program solely use emotions to destroy the incoming game components. However after our initial testing it appears that it is working as intended.

Usage instructions:
In the directory: **python -m SimpleHTTPServer**
In the Browser: open **http://localhost:8000/affectiva.htm**
On the Page: Press the ‘start’ Button at the bottom of the screen, then both Affectiva and WebGaze will be active and tracking at once.

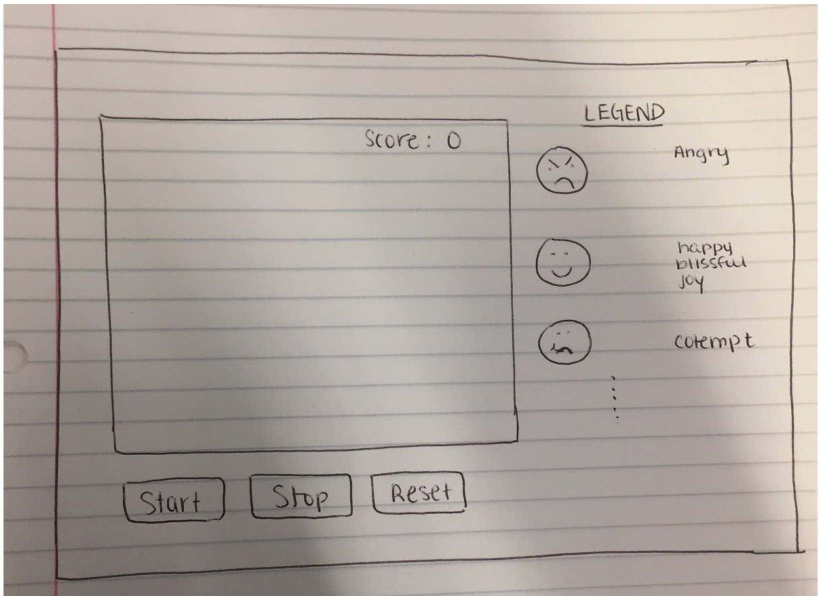
Usage Examples





use eyetracker to target specific emotion-face.

Wireframe


Score: 0
LEGEND
Angry
happy
blissful
joy
cotempt
Start
Stop
Reset