Maya DeBellis and Brooke Weil
mdebel01 and bweil01
Comp 86
Assignment 6

# Project Proposal

## Goal of Project:

To create a online drawing canvas where user can create images with their eye movements.

## Project Description:

Using the WebGazer.js API eyecam tracking to create an application that allows for users to draw with their eye movements. Users will calibrate the eye tracker by moving their eyes to look at the cursor and clicking. Once the user thinks they have sufficiently calibrated the tracker, they will be able to click on colors and draw with their eyes. All of the drawing will be done through eye movements, but the start and stop drawing commands as well as the color change commands will be done through clicking buttons on the webpage.

## Target Users:

People who want to draw and create new images in a creative way. Not the most accurate way to draw an image, but it's a fun, lighthearted, and challenging drawing task to try.

## Technology Requirements and Risks:

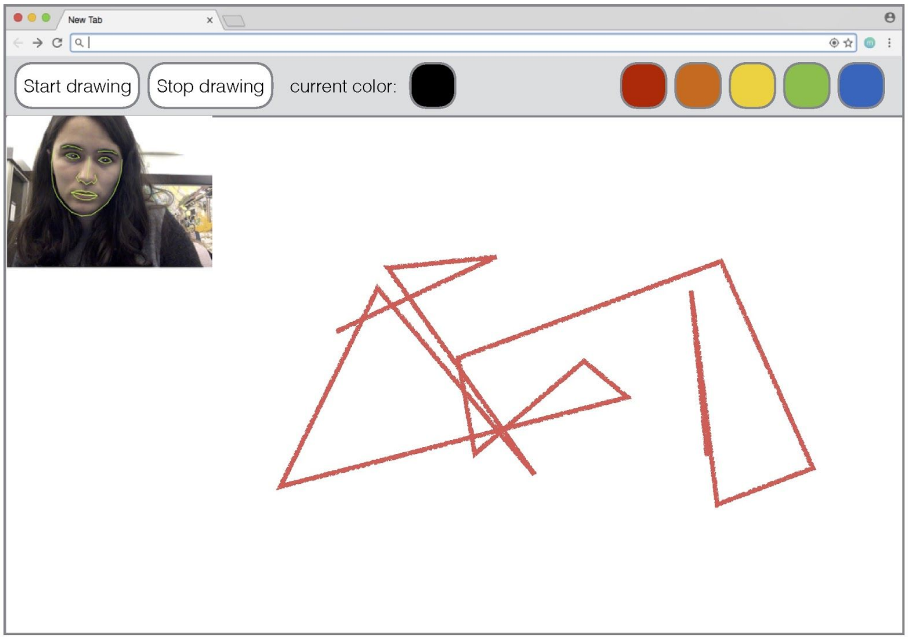
We will use the WebGazer.js API to implement eye tracking. This API will allow us to find the predicted screen coordinates targeted by the user's eyes. WebGazer's website provides some example programs we can reference while working on our own program. We plan to implement a web app, so we will code in HTML and Javascript.

A risk of the project will be that we are unfamiliar with the WebGazer API, but we can look at the example program as a model, so that should help mitigate that risk. Additionally, the eye tracking is not incredibly accurate, so we will probably choose to take some measures to smooth out the lines drawn by the user if the raw coordinates are choppy. Another risk of this plan is that neither one of us has built anything like this as a web application, so we may need to learn more HTML/Javascript along the way.

Maya DeBellis and Brooke Weil
mdebel01 and bweil01
Comp 86
Assignment 6

## Project Sketch:

Here is an approximation of how we'd like our web app interface to look when using our application:



## Technology Feasibility Test

We downloaded an example of a program using the WebGazer.js software. At first it did not run on our local machine and we had to set up a local HTTP server. Once we set this up, we were able to run the example properly and can see that the technology is capable of being programmed and tested on our machines. Please see JavaScript files for more information on the example game we downloaded.

## Design Documentation

Please see `README.md`