Nonwimp2 documentation

Details about how you arrived at the final design, how you implemented the main features of your program, and any modifications you needed to make along the way:

We arrived at the final design after fiddling around with WebGazer, we decided that a game like the one we implemented was a good choice. The main features of the program involve tracking the user's eyes through a webcam and moving the cursor that way. Since the program itself is not the most accurate, we had to implement a calibration that the user must go through before starting the program so that they can use the program correctly. We also had to implement an offset so that the gaze method of webgazer would work correctly.

A short description of the good aspects of your UI design:

We have few widgets because most of the work done is through webgazer. We made the UI simply so that it does not clutter the screen. The focus of the user should be on the game itself, not the UI so we think this was a good design choice.

Interesting / unusual technical features or issues:

As mentioned before, we had to create a small offset for the gaze when we obtained the coordinates for the gaze, but apart from that everything worked smoothly. We also tried to implement the calibration in the landing page, and then take the player to the main game with the calibration saved, but we couldn't get it to work properly

Some interesting aspects of your UI design:

Our UI design didn't need much, however we had to spend some time figuring out how to edit the string in the level and scoreboard so that we can edit it like a integer, although it was a string.

Instructions for how to run your program:

This is the link for the program: https://connord627.github.io/nonwimp/

First, there will be a landing page, and clicking the button will take you to the main game. Here, some instructions will pop up, instructing the user how to calibrate the webgazer. Once that is done, the user can begin playing the game. The goal of the game is to have the lowest score possible. Every time a user looks at a square, they gain a point. Also, every 10 seconds, the user passes onto the next level and the difficulty increases.