Development Process

Group 8

Xingyu Li - 400032949
Adam Gordon - 400036676
Jacky Chan - 400015317
Bowen Yuan - 400005913
Jeffrey Sun - 400022051
Tongfei Wang - 001437618

Group 8

Eye-Clicker - Development Process

How we are developing our project

For now we have already found some useful openCV open sources projects that use cascade to recognize people's faces and eyes in an image. We are trying to configure the code to our needs and are continuing to search for other helpful open source projects to make pupil detection possible. Once that goal has been achieved, we will extract the eyes' and the pupils' position in the form of matrices so that we can calculate the relative position between them. Then we will be focusing on developing the relationship between this relative position and the position of cursor on the desktop of a laptop. We are looking to do this with AI. We will also allow the user to finetune the software as each user's computer, desk, and seating arrangement will be unique. Last but not least, we will write a software that interacts with the OS and tell it to move the cursor to the position we got. More functionalities will be developed from this point, like GUI, clicking, dragging, marking, etc.

Dividing work

We are developing two different ways to track the user's pupils and so the team is splitting into two groups. Group one is responsible for developing pupil-tracking algorithms for the webcam and finding open-source for AI algorithms that predict where the cursor should be placed on the screen based on pupil detection. Group two is responsible for doing research on how infrared sensors work and how to implement IR technology alongside with eye-tracking to help achieve better accuracy.

Each group has a group leader who keeps track of each member's progress. Group one leader is Bowen Yuan and his team members are Jeffery Sun and Tongfei Wang. Group two leader is Adam Gordon and his team members are Xingyu Li and Jacky Chan.

Outstanding Work:

1. Calibration
    a. Calibration should improve the Eye-Trackers accuracy
2. GUI/Menus
    a. Ability to activate/deactivate cursor control
    b. Ability to activate/deactivate visual indicator of where you are looking
    c. Ability to activate Calibration
3. Improved Accuracy Algorithm
    a. Develop an algorithm that helps improve user experience and accuracy with regard to cursor control
        i. A possible algorithm is the "Precision Circle Algorithm". This algorithm will have a specifically sized circle placed around where the user is looking. The size of the circle should be nearly equal to the accuracy of the Eye-Tracker so that when the user looks at a single spot, even if the Eye-Tracker "jitters" due to disturbances, it will jitter no greater than the radius of the circle (the accuracy). This circle will only move once the Eye-Tracker detects that the user it looking outside of the circle (a larger and likely intended action). The center of the circle will be where the mouse cursor is placed. A head movement in a certain direction will move the mouse cursor temporarily in that direction of the head movement.
        ii. Possibly smooth mouse movements. I.e. if the camera tracks at 10fps the mouse should not instantly move every 0.1seconds but constantly smoothly move.
4. Clicking Algorithm
    a. Needs to be ergonomic and not accidentally activate. Some possibilities are:
        i. Blinking
        ii. Staring at a spot for X amount of seconds
        iii. Voice control
        iv. An external button/remote
5. Eye-Tracking

a. Develop an algorithm that can detect the pupil. Possibilities are the following:
    i. We could use this with the software that we already have that tracks eye position to determine where the user is looking.
    ii. Other sources say that it may be better to use pupil tracking along with nose tracking because the nose is a constant spot on the face that doesn't move. This allows for a more stable/accurate tracking.
    iii. Utilize an AI algorithm on our self-created data set. This AI will interpret the pupil position and output the cursor position.
b. Improve fps (our openCV has a low frame rate currently), some possibilities to investigate
    i. Multithreading
    ii. Using a stronger computer
    iii. Lower resolution (will lower accuracy)
6. Organize our files on Github
7. Everyone in the group install openCV and successfully run the face and eye recognition software which we have already wrote. This is more challenging on some computers than others.