EyeClicker

System Requirements Rev 1

SFWR ENG 4G06 / MECHTRON 4TB6
GROUP 8

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

Contents

1 Revision History

Date	Revision	Authors
Oct 28, 2019	0	All group members
Feb 25, 2020	1	All group members

2 Purpose

The purpose of the EyeClicker is to give our users an alternate method to use their computer. With our device, a user will be able to control their computer or laptop without having to use a mouse, and instead use their eyes. This is especially helpful to users who have limited mobility with their arms, those who are injured, or simply someone who wants to rest. Given an alternate method of using a computer, our device can help more people use a computer comfortably and be connected to the digital world we live in. Eye tracking has become fairly popular over recent times, and our EyeClicker will utilize this concept to provide a full user experience that mimics the behavior of a mouse. The EyeClicker will identify where the user is looking at to position the cursor to that spot on the screen. It will also track a unique set of simple eye movements/actions as well as certain voice commands to deliver the appropriate response. Whether the user wishes to click, drag or enter text the EyeClicker will provide all those functionalities simply though tracking the eye and voice control.

3 Scope

The project will be based around tracking user's eye movements and making the cursor react, including but not limited to moving the cursor, left-clicking and click and dragging. This will be achieved through image processing, more specifically, human eye recognition and a well-developed algorithm to control mouse actions.

In-scope functionality items for the EyeClicker including the following:

- Calculate the position the user is looking on the computer display
- Calibration system to increase the accuracy of the EyeClicker
- Allows user to perform a left-click as well as dragging with certain special actions that can be performed with voice commands
- Be able to let user disable the EyeClicker system to prevent misoperation
- Provide a functional GUI to enable/disable the EyeClicker and enter the calibration system

The following items are out of scope: Allows user to control the cursor with voice input

4 Context Diagram With Boundaries

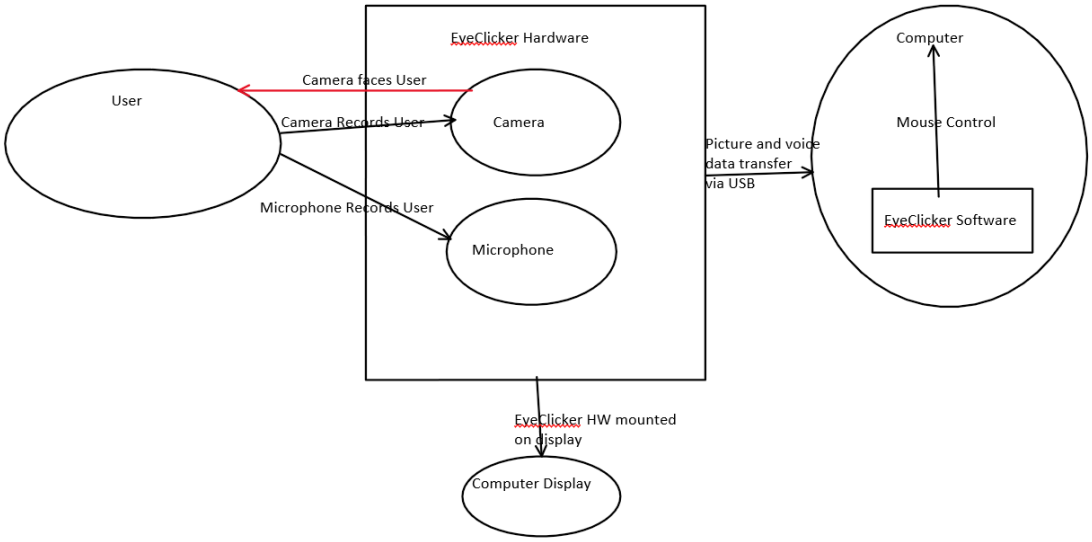

Figure 1: Context Diagram

The EyeClicker hardware (the camera and the microphone) is to be provided by the user, and can be either external or internal computer components.

5 Monitored and controlled variables (with units)

Monitored Variables

Name	Type	Range	Units	Comment(s)
image_frame	Binary	N/A	N/A	Video input captured by web camera
coord_landmark_left	Two dimensional array of floating numbers	[0,1]	Percent	Coordinates of points around user's left eye(in percentage)
coord_landmark_right	Two dimensional array of floating numbers	[0,1]	Percent	Coordinates of points around user's right eye(in percentage)
coord_pupil	Array of floating numbers	[0, 1]	Percent	Coordinates of points indicating left and right pupil positions(in percentage)
get_cursor_xy	Boolean	[0, 1]	Percent	Signal to tell the module to output the current position of the cursor
voice_input	Binary	N/A	N/A	Voice commands from the user
coord_pupil_training	Two dimensional array of floating numbers	[0, 1]	Percent	Array of coordinates of points indicating left and right pupil positions(in percentage)
coord_pupillooking_training	Two dimensional array of floating numbers	[0, 1]	Percent	Array of coordinates of points indicating where the user's eyes are looking at on the screen(in percentage)
GUI_start	Boolean	[0,1]	N/A	Signal for initializing the Eyetracking Module
GUI_exit	Boolean	[0,1]	N/A	Signal for exiting the GUI Module
GUI_calibration	Boolean	[0,1]	N/A	Signal for initializing the Calibration Module

Name	Type	Range	Units	Comment(s)
predicted_cursor_position	Array of floating numbers	[0, 1]	N/A	The coordinate where the cursor will move to
cursor_xy	Array of floating numbers	[0,1]	N/A	Current coordinate of the cursor(in percentage)
click_left	Boolean	[0,1]	N/A	Perform a double left click action if Click_left = 1
click_right	Boolean	[0,1]	N/A	Perform a right click action if Click_right = 1
init_calibration	Boolean	[0, 1]	N/A	Initially 0, 1 when user clicks Calibrate, set to 0 when calibration is initiated
calibration_in_proccess	Boolean	[0, 1]	N/A	Set to 1 when calibration is in process, else 0
predicted_cursor_position	Array of floating numbers	[0, 1]	N/A	Coordinates of points indicating where the user's eyes are looking at on the screen(in percentage)
Init_eyetracking	Boolean	[0,1]	N/A	Initialize the Eyetracking Module
Init_calibration	Boolean	[0,1]	N/A	Initialize the Calibration Module

6 Constants

Constants

Constant	Unit	Value
Camera Frequency*	Pictures / Second	10
calibrationCoordinates[5]**	(x%,y%) ***	(0,0), (1,0), (0,1), (1,1), (0.5,0.5) where 1 = 100%

* Camera Frequency is the frequency the camera will take pictures and feed it to the computer.

** Five coordinates used during the calibration procedure

*** x% is percentage is the horizontal percentage of the screen you are looking at where the left edge is 0% and the right edge is 100%. y% is percentage is the vertical percentage of the screen you are looking at where the top edge is 0% and the bottom edge is 100%

7 Behaviour Overview

The user will connect the device to the computer via USB and mount the device onto the screen/monitor (facing the user). The user will then open the EyeClicker software to calibrate the system to improve accuracy. The calibration will be done by having the user look a predetermined point on the screen and confirm they are doing so. This will be repeated for 16 predetermined points. They can now enable the EyeClicker tracking through the GUI or voice control. The user can left click while the EyeClicker is enabled by saying "Hey EyeClicker, left click". Then the EyeClicker will send a left mouse down event.

8 Diagrams Showing Functional Decomposition

9 Behaviour Description

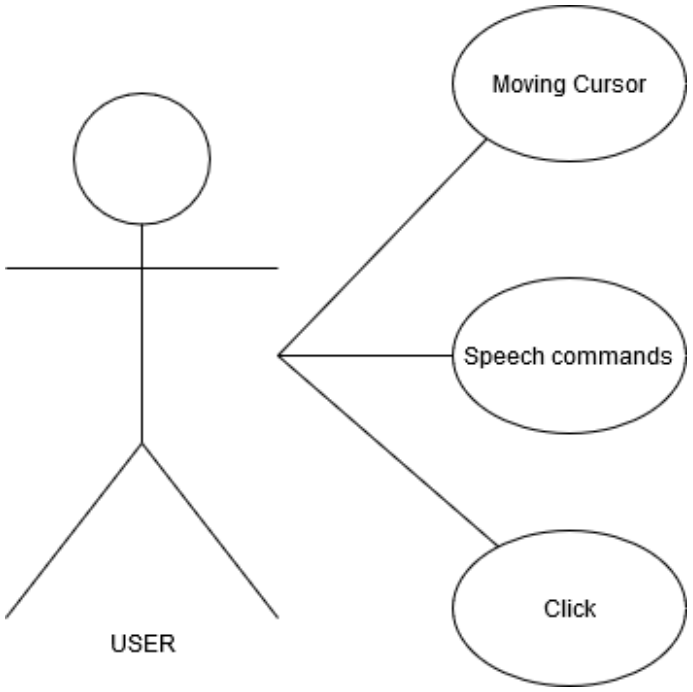

Figure 2: Use Case Diagram

10 Rationale

No additional rationale.

11 Performance Requirements

- The deviation of detecting where the EyeClicker thinks user is looking compared to actual position the user it looking at must be smaller than 5cm while within effective working distance of the EyeClicker (60cm from the screen).
- The latency of detecting whenever users' eyes are moving must be shorter than 150 milliseconds.
- The latency of receiving, translating, and executing user's voice command must be shorter than 3 seconds.
- The accuracy of speech recognition must be greater than 65 percent.

12 Normal Operation

During the calibration stage, the user should be within the effective working distance of the EyeClicker (60cm of the screen). Sitting too far away from the camera causes inaccuracy or worsens the ability of EyeClicker to locate users' eyes and to control the cursor accurately.

During the working stage after calibration, the user now can navigate the cursor using

their eye movements. Left and right clicking are the basic functions the user can operate. These functions operate at the optimal level when the user is sitting at a stable position and as close to the position that the system was previously calibrated in.

Speech recognition is one of the solutions to help the user have more capability to do a task. When using speech recognition, the user should speak at a steady pace to obtain optimal performance. The speech recognition will also be used to activate the left and right click mouse actions through predetermined commands that the user will need to learn.

13 Handling Undesired Event

- Unable to click the desired point on the screen. When the user is having trouble clicking the desired item, one should redo the calibration to improve the accuracy. To access the calibration test, click the "Calibration" button in the GUI.
- EyeClicker cannot locate the user's eyes after turning his/her head away from the screen. When the user looks away from the screen for too long, EyeClicker might lose track of the locations of the user's eyes. To solve this problem, the user should move closer to the camera/screen (around 60cm away) and stare at the middle of the screen for a short amount of time.

14 List of Requirements (Likely to Change)

1. The system shall allow the user to left click by focusing on a spot for 1.5 secs and then blinking (instead of utilizing voice control).
    - Rationale: Allow the user to have more than one method to left click so that the user can still left click when they are in situations where voice control is not feasible (a quiet library).
2. The system shall notify the user if he is sitting too far away from the camera/screen.
    - Rationale: Sitting too far away from the camera causes inaccuracy or worsens the ability of EyeClicker to locate users' eyes.
3. The system shall be able to execute additional speech commands to be more user-friendly, specifically dragging and copying and pasting.
    - Rationale: There are some limits for eyes to operate certain functionalities such as copy and paste.

15 List of Requirements (Unlikely to Change)

1. The system shall have enough accuracy to reasonably track where the users' eyes on the screen.
    - Rationale: Needed to allow for accurate cursor control for the user.
2. The system shall precisely calculate how long has users focused on one spot.
    - Rationale: The system should know how long has users focused on one spot to enable the clicking functionality.

3. The system shall have 16 predetermined points for calibration.
    - Rationale: The system should have at least 16 predetermined calibration points to improve the accuracy of detecting where the user is looking at on the screen.
4. The system shall translate users' voice correctly into system commands.
    - Rationale: The system should allow users to say commands while avoiding unintended actions.
5. The system shall allow the user to perform mouse click actions through voice commands.
    - Rationale: To meet the goal of allowing users to perform left and right click through the system.
6. The UI can initialize all the voice, calibration, and eye-tracking modules effortlessly.
    - Rationale: Provide a UI to give user feedback for the system's operation. This UI also needs to be able to enable the other modules for the system to be stable.