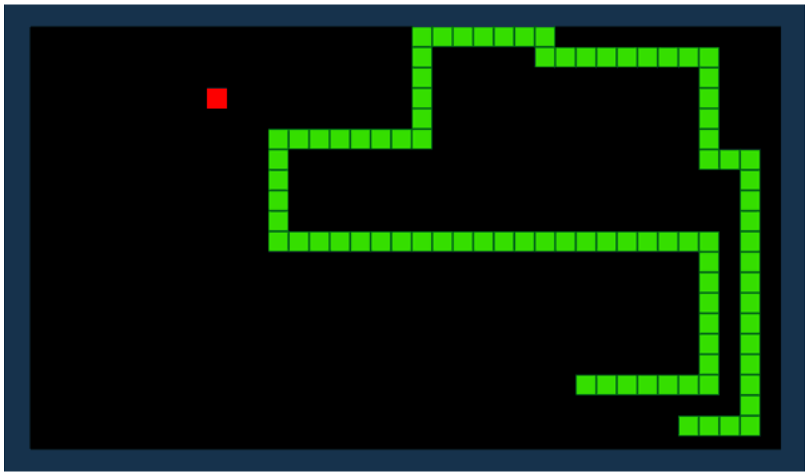

Final report - II2302

Gaze-based snake game

Examiner: Mark Smith

Presented by: Chloé Bernardoni
Lucile Cerda
Joshua Cohen-Dumani
Joe Saad

Contents

1 Introduction and context

1.1 Missing your old Nokia ?

In the frame of the sensor-based system course and with a nostalgia of the famous snake game on old Nokia phones, a new version of the game has been developed. The idea is to steer the snake using the player's gaze using special glasses on their face. It's very difficult for users to control their gaze and eye movements can sometimes seem stochastic. Detecting and characterizing this motion can be subject to many human factors such as blinking. Being able to control the snake with the eyes would offer an enhanced gaming experience, but would also lead to more inclusive games for those who may have problems with hand-based games.

1.2 Description of the system

A first idea was to use a color sensor to measure the orientation of the eye. However that method would have been too eye-dependent and thus the algorithm would have to be modified to adapt for each user. Furthermore, the up and down movement would have been more difficult to detect since the white part of the eye above and below the iris is hidden by eyelids. The eyelashes, the lighting conditions and the partially closed eyes would have been a big hinder to the color-based detection.

Another simpler way would have been to use camera but, as privacy concerns are becoming a crucial part of the customers choices, the main goal of this device is to be able to track the movement of the eyes without gathering personal characteristics and data.

For that, an alternative has been used: it consists of 4 transmitters and 4 receivers used to send and receive IR from the 4 different directions. This IR system is used to detect a distance since, depending on the reflection distance, an emitted IR ray may or may not be received by the receiver due to attenuation. A visualisation of that is showed in Figure 1.

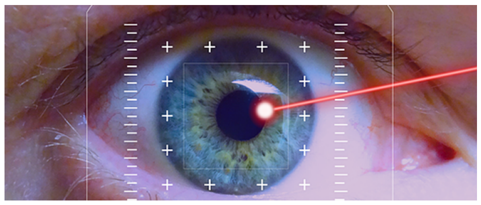

Figure 1: IR incident light from one direction.

The IR-sensor matrix will be put in front of one eye while the other one will be able to normally see without disturbance. The uncovered eye will also serve as a witness to the real motion of the covered eye (as both eyes usually move in the same direction). It will be therefore possible to compare the expected behaviour to the output of the system. The sensor matrix will consist of up to 4 IR receivers (4 is the minimum requirement for predicting a 4-direction motion).

Finally, nothing better describes a sensor-based system performance than the measurements done on an actual implementation. For that, a good accuracy will be the main target (since having the snake going left when a player is moving their eye to the right is definitely worse than the sweaty fingers on the touch screen!). Experiments will be conducted to measure the accuracy of the sensor in that application.

1.3 Specification

- The device should be able to classify the staring direction with high accuracy or else admit a conservative alternative (looking straightforward).
- Measurements should not be affected by natural predictable human errors such as blinking.

- The measurements acquisition period and the averaging time shall be reasonably small (less than 0.2 s) to give a fast-response behavior coherent with the gaming application, and large enough to ignore blinking and unstable gazes. That will lead to choose a loop frequency between 5 and 10Hz.

2 Software and hardware description

2.1 System block diagram

Two parts are discussed this section: first collecting and processing data from the sensor, and then implementing the system in a snake game using a graphical user interface. The following figure shows a block diagram of the system, mainly describing the data flow between consecutive elements.

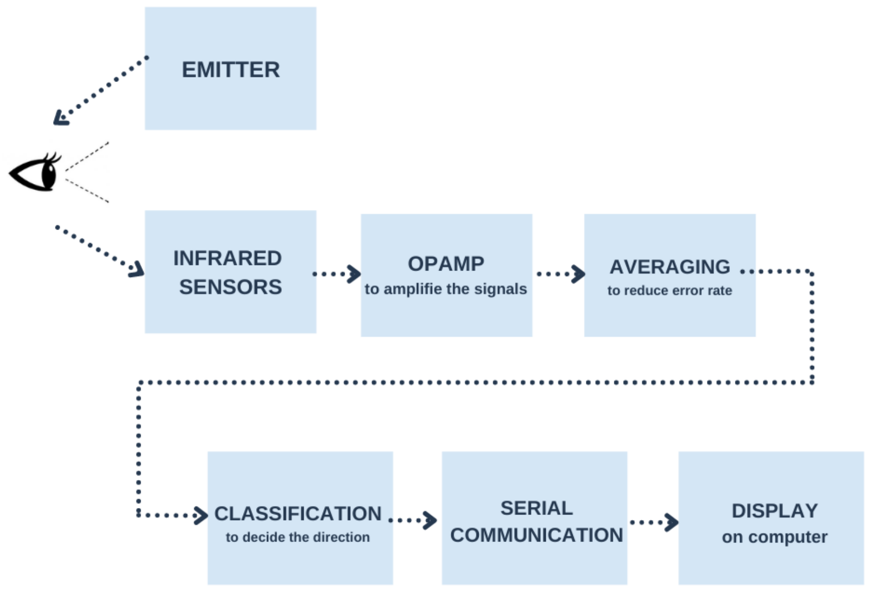

Figure 2: Block Diagram

2.1.1 The sensing circuit

IR rays are emitted towards the eye, some are directly reflected towards the sensor (by the white zone or the eyelids) whereas the ones entering the pupil travel a bigger distance, making them attenuated before reaching back the sensors. As infrared sensors are phototransistors, the gate opening varies as a function of the collected IR beam. However, quantum physics theory (tunneling effect) states that the variation is an exponential function of the collected beam, which means that the output can be considered nearly digital. In short, the receiver behaves as a distance comparator with respect to a threshold distance. The latter is fixed by varying the resistance that is put in series with the phototransistor, but also by adjusting the intensity of the emitted beam.

2.1.2 Data processing

Operational amplifiers used in comparator mode detect the variation of voltage between the 2 phototransistor states and outputs a digital signal value accordingly.

Averaging and classifying are the main part of the processing step as it will be explained further in the software flowchart.

2.1.3 Communicating data

Data is then sent to the GUI using a serial communication and is used to move the snake on the screen. More specifically, the direction is send to an algorithm running on Python through UART communication via a computer.

A simplified schematic is represented in figure below:

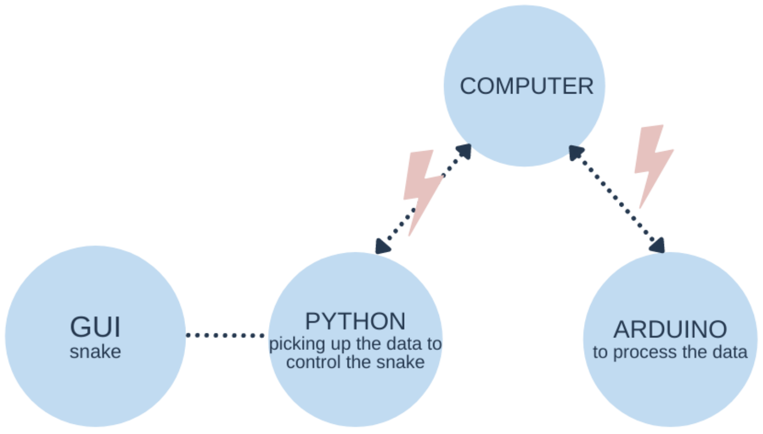

Figure 3: Flowchart for the processing algorithm

2.2 Hardware architecture

2.2.1 Sensor matrix

4 emitters and 4 receivers have been placed in 4 different directions in such way that the emitting/receiving circuits can be oriented independently. Since the IR beam given by an emitter is directional, 4 emitters are needed. Otherwise, using many receivers with one diverged emitter results in the loss of directions.

The device is shown in the following figure:

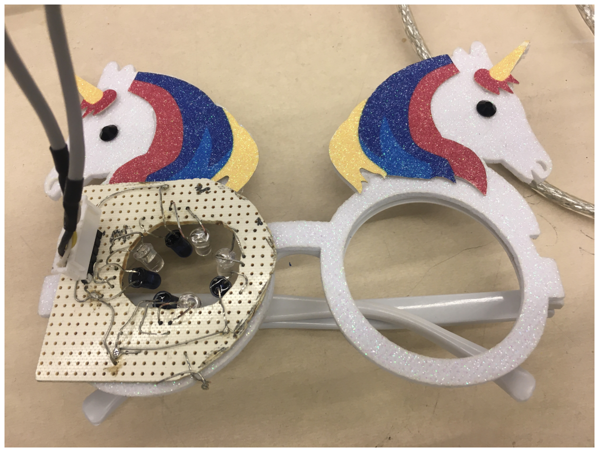

Figure 4: Emitters and receivers for 4 directions

2.2.2 Electric circuit

The amplifying circuit can be basically seen as 4 amplifying sub-circuits in which 4 OPAMPs are used in comparator mode. Potentiometers have also been added to control detected distances by varying the intensity of the IR beam and/or the opening of the phototransistor gates with different intensities. A full schematic is shown below:

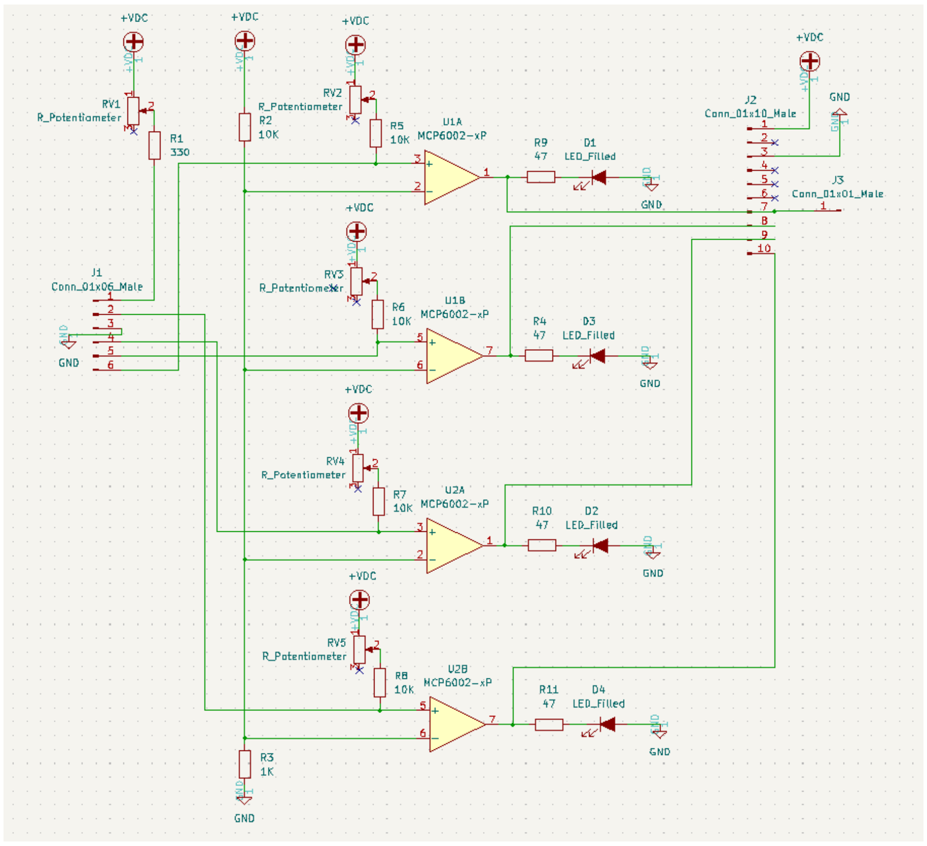

Figure 5: Amplifying circuit schematic

Direct wires connect the left connection pins (on the schematic) to the eyeglasses (sensor matrix). Emitters are supplied with the voltage between pin 1 and pin 3, and the latter is controlled by the potentiometer RV1. The 4 phototransistors are plugged to one of the pins 2,4,5,6 and the ground pin 3. Current, therefore distance, is controlled by the potentiometers RV2, RV3, RV4 and RV5.

2.2.3 Printed circuit board

The previous circuit has been printed on a PCB and the 3D model and layout are given in the following figures:

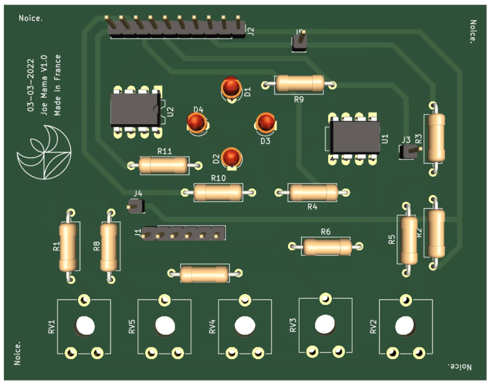

Figure 6: Circuit 3D model

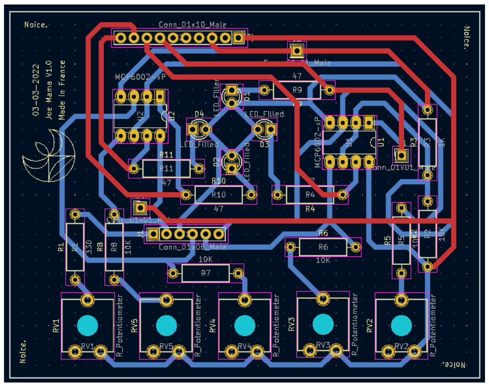

Figure 7: PCB layout

2.3 Software flowchart

Here is given the flowchart of the processing algorithm, it can be summarized by the 4 following steps:

- The output values of the OPAMPs are collected on an ARDUINO UNO and then normalized to 0 or 1 based on a threshold.
- An average on 500 (to cover blinking while keeping high sensitivity to fast eye-motion) values is computed.
- The average result is then normalized to 0 or 1 (binary state) for each receiver.
- A direction is computed with these normalized output: a direction is chosen if and only if only one output is 1 while others are 0, else the snake moves straightforward.

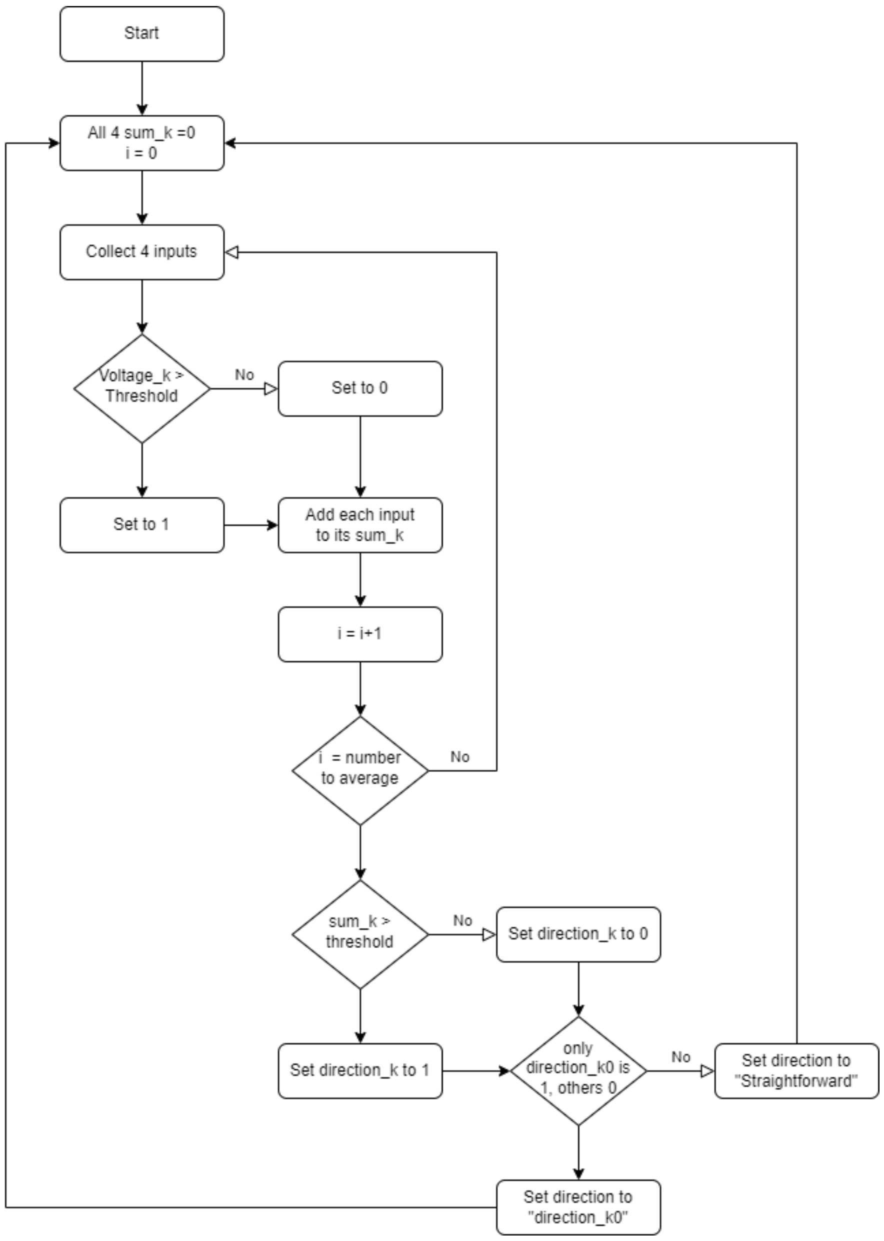

Figure 8: Flowchart for the processing algorithm

3 Data collected

3.1 Radial threshold

The first tested parameter is the open angle that is defined as the deviation from the center of the eye to a given direction. A radial threshold decide from which open angle a change in direction is detected. The further (larger open angle) the experimenter look towards a given direction, the more accurate is the response of the system. Smaller open angle means lower distances between directions (smaller circles) and thus more errors. The radial threshold experimentally found is 18°, meaning that an open angle of less than 18°results in mostly undetected gazing directions.

Measurements have been collected for open angles of 14°and 45°. For each angle the outputs of 100 measurements are plotted for each of the 4 possible directions. Results are given by the following graphs:

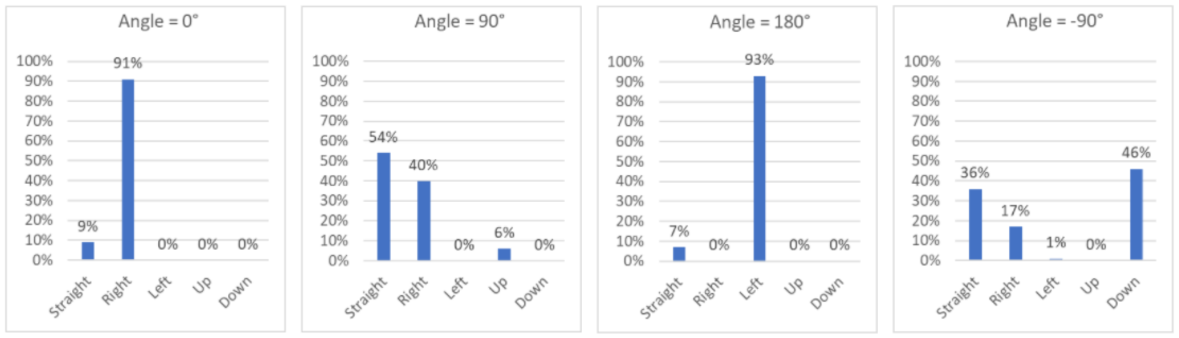

Figure 9: Accuracy for 100 measurements with an open angle of 14°

Results show that good accuracy (more than 90%) is obtained for the left and right directions even for an open angle lower than the threshold. However, the up (resp. down) directions are wrong in 94% (resp. 54%) of the time.

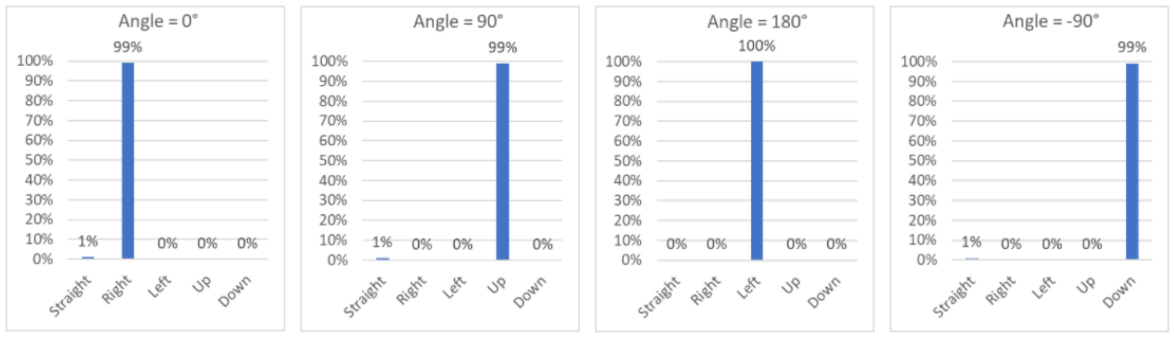

Figure 10: Accuracy for 100 measurements with an open angle of 45°

An accuracy of at least 99% is obtained for the 4 directions when setting the open angle to 45°.

3.2 Rotation angle limits

The different outputs have been observed when varying the rotation angle (= the angle on a trigonometric circle with a given open angle) with empty and overlapping zones appearing on the diagram. Each direction has a range of around 100°(the exact values are shown on the following graph).

For instance, if the gaze is oriented between -30°and 30°, the snake will go to the right. Empty zones are sections in which no direction is toggled (not covered by sensors) while overlapping zones are those for which 2 sensors return a 1. In both cases, a straightforward direction is taken. White zones are empty zones while mixed colors define an overlapping zone

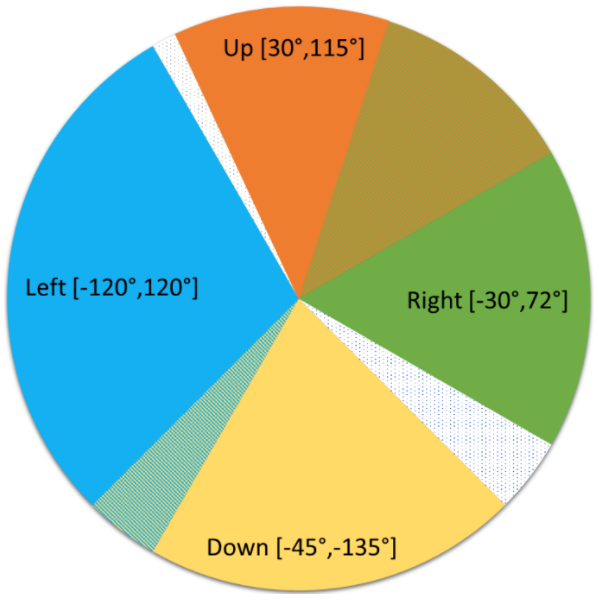

Figure 11: Rotation angle limit diagram

3.3 Empty and overlapping zones

As seen in the previous figure, not all the area of the circle is covered by the sensor, and some sections are covered twice. What about the accuracy of measurements in these zones?

Experiment shows that accuracy is still as high as 99% for the two even though the behavior of the outputs differs. For each case, the given outputs (those set to 1) are given in addition to the resulting classification.

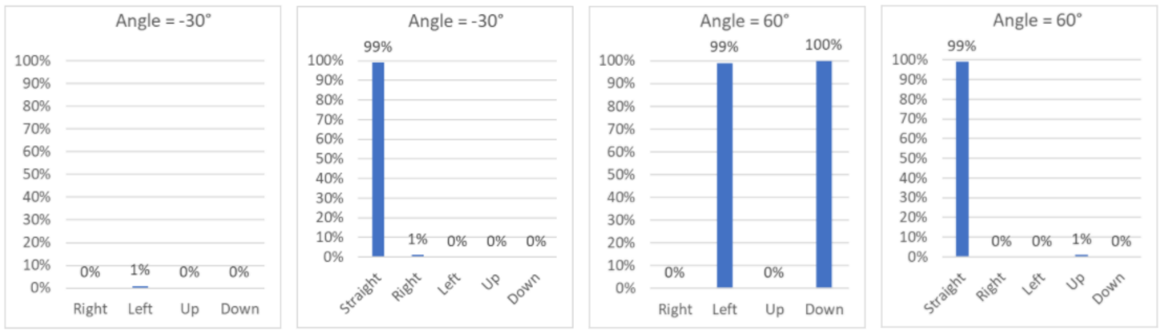

Figure 12: Accuracy for 100 measurements with an open angle of 45°

3.4 The effect of averaging

The algorithm for processing data averages over 500 samples before choosing an output direction. The idea comes from the natural human blinking factor. In fact, when eyes are closed, the device outputs a straightforward direction as none of the sensors can detect the pupil.

However, blinking takes approximately less than 0.1 second which means that it is possible to average over many samples and cover the blinking phase with other values if the sampling period is long enough and the threshold low enough to accept a reasonable number of positive decisions, but also high enough to avoid false uncertain selections. The choice is to take 500 samples with a threshold of 100 for a selection.

To justify the role of averaging, an experiment on decisions without averaging has shown that the accuracy drops to around 94% and a ”straightforward” decision is made when blinking. Results are given as follows:

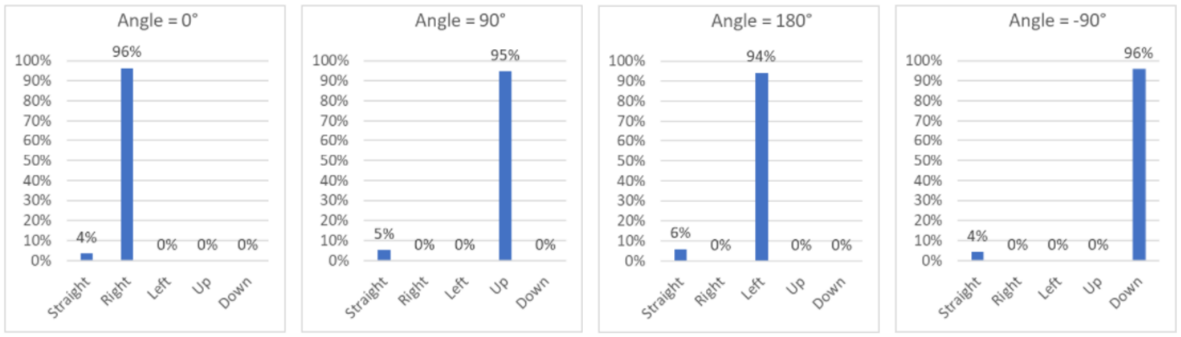

Figure 13: Accuracy for 1000 measurements without averaging with an open angle of 45°

3.5 Center gazing

When looking towards the center of the circle, no direction shall be taken and the result shall return ”straightforward”. The experimental result confirms that as they show that, most of the time, the left and right sensors detect the pupil which results in a compensation of directions. The down sensor also contributes in around 70% of the time while the up sensor can rarely detect the pupil due to the eyelids and eyelashes that cover the sensor when staring in the center.

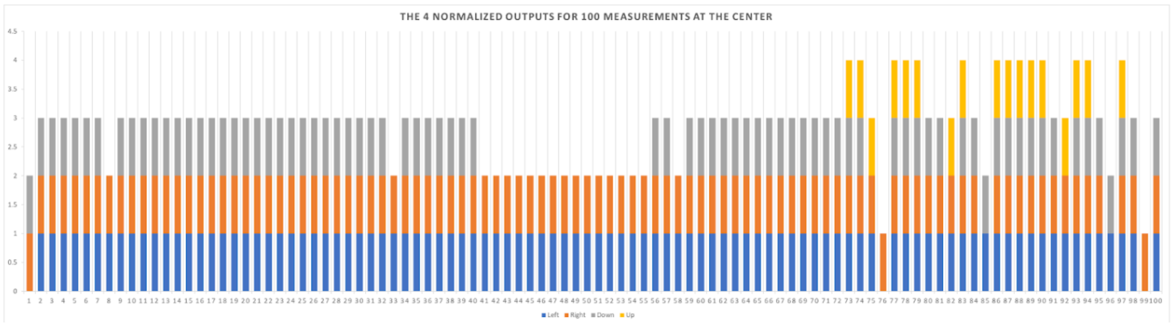

Figure 14: Accuracy for 100 measurements with an open angle of 45°

The accuracy is hence of 98% which is satisfying for a direction that is not theoretically monitored by a proper sensor, but therefore deduced by the output of the other 4.

4 Task distribution

The work is an achievement that mainly came as a fruit of teamwork and mutual effort. Each participant has contributed based on their expertise and the contributions have been the following:

- Joe: Hardware design for the sensor and the amplifying circuit on PCB, experiment framework setup and statistical analysis of the results.
- Chloé: initial circuit on the arduino board, software design (Snake game and coding) in cooperation with Lucile and Joshua, technical test and calibration.

- Lucile: software design with Joshua and Chloe, calibration of the IR sensors, test of our product and collection of data.
- Joshua: Software design (snake game gui, communication with Arduino board, input/output implementation), technical tests.

5 Conclusion

5.1 Performance analysis

Taking everything into account, one could say that our device works well, but is limited in terms of its usability. It is very accurate and reactive, but in its current state needs to be calibrated to each person. Our data shows that the sensor clearly works, but many UX improvements could be made to smooth usage.

5.2 Cost estimation

Item	Quantity	Unity price	Total price
Arduino UNO Board	1	250 SEK	250 SEK
Infrared Emitter	4	1 SEK	4 SEK
Infrared Receiver	4	1 SEK	4 SEK
Glasses	1	20 SEK	20 SEK
Total			278 SEK

Table 1: Cost summary

5.3 Recommendations for a future cycle

Our project has generally fulfilled the specifications set out at the beginning of the development process. However, many improvements are possible which could be touched upon in a future development cycle:

- Bring unit cost down. the current cost is too high compared to the usability of the end product. A possible solution for this is to use a cheap MPU which could bring the price down significantly. However it is generally to be expected that a prototype's cost basis is relatively high compared to production costs once a product is established.
- Improve UX. While functionality has been accomplished for the intended user (Chloé), the user experience was not very intuitive. This could be improved by making the circuit more compact, perhaps adding a hull for the sensors in order to hold them in place, try to make the whole thing portable (using a battery for example). Explore ways to keep sensor usable while maintaining vision in the sensed eye (smaller sensors, different placement, more powerful IR sensors, etc.).
- Improve inter-operability. While the sensor worked very well for Chloé, it needed to be calibrated for her eyes and face shape. It worked decently well on some other people but the results have massive variance depending on the user.
- Expand use cases. Our gaze sensor can be adapted to anything with 4-5 DOFs, which means that a wide array of use cases exist. Snake was a more "fun" application but one could imagine many more (augmented reality), which would give new specifications to the sensor (take less space, faster frequency, etc.).

5.4 Closing thoughts

Taking this into account, we might argue that someone is unlikely to pay 280+ SEK for our prototype. However, given that almost 90% of our costs are due to the Arduino pack we had to buy, using cheap MPUs and more compact circuits might bring this cost down significantly, which could potentially lower the price point to where our device could be a fun buy for many people. Of course the

UX needs to be significantly improved if we would like to commercialize at any time. Inter-operability would also need to be addressed.

All in all, Sensor Based systems was a very interesting class which combined a lot of prior knowledge and was very hands-on. It allowed me to learn a lot about sensor use cases and development processes. It is one thing knowing that sensors are all around us and another to learn how they work and exactly where they shine/are limited. The classes on RFID and identification problems were particularly interesting, and sparked many questions regarding privacy (and combination with other technologies such as blockchain, a personal center of interest). A key takeaway I also took from this course was to better understand the performance differences between analog has over digital. It was very enjoyable to do a DIY electronics project like this, where we had so much freedom and were given the tools to execute.