

Sohan Project

**HackaHealth Managers:** Bastien Orset, Sixto Alcoba-Banqueri

**Team Members:** Annina Stuber, Christelle Schneuwly Diaz, Darja Nonaca, Luisa Sanchez, Martyna Miśkowiec, Marzia Del Zotto, Robert Felipe, Sriskaran Viros.

*****Team Sohan. All the members presented in the picture participated in this project during a hackathon in 2020 at Geneva organized by HackaHealth. Sohan and his parents are also in the picture as well as his two therapists.*

**Project definition**

Sohan is an 8 years old child that cannot speak. Thus, he needs to communicate to other people using devices such as a computer interface with eye tracking and pressing buttons. This communication was facilitated so far thanks to pictograms placed on the board in front of him and a computer with Grid3 software (<https://thinksmartbox.com/product/grid-3/>). This software provides a large vocabulary and numerous types of control (button, eye tracker,...)

However, most of the controllers proposed in Grid3 interfaces require a high motor effort. Because of this, his capacity of interaction with other people is largely impaired. In this project, the main goal is to design a controller that allows him to better communicate thanks to an efficient interface requiring less physical effort.

Sohan has cognitive impairments that do not allow any control of his movements in general. Moreover, he cannot use fingers for pointing or typing or pressing buttons (no hand gestures). He can move only his left hand on the table. His head, very often, falls down on the left side of the body. This makes it very difficult to use eye-tracking devices to detect the gaze for leading the computer interface.

**Brainstorming**

List of the possible solutions:

* **Eye-tracker** to detect the motion of his eyes;
* **Leap motion**: a infrared camera to detect the motion of hands;
* **Capacitive sensors** that can detect the contact (position, direction etc.) with some surfaces of his left hand;
* **Hand tracking camera** to detect the motion of his left hand via image processing and computer vision.

**Chosen solution**

We decided to combine 2 solutions using capacitive sensors and hand tracking camera to give both discrete (keyboards) and continuous control (mouse). We discarded the eye-tracker and the leap motion because they are not able to accurately detect his eyes and hands motions for a long time due to his unsteady posture on the chair (issues with the calibration of the eyes or hands position link to the angle of the infrared camera - occlusion). Additionally, the leap motion presents a problem of reflection if put under a transparent table (the signal is not detected at all).

Capacitive sensors and the hand-tracking camera present different advantages:

* Easily detect his hand motion even not accurate and precise on a specific point
* Not sensitive to his posture or body movement
* No calibration for the posture or the position of the body
* Easily control the computer interface without any physical effort

The combination of the two solutions could even improve the precision of motor detection and decrease the physical effort, providing more precise and clear commands to use for better communication.

Brief explanation on Solutions:

1. **Capacitive sensors:** four arrows are displaced on the table (commands), indicating the four directions (up, down, right, left) plus another bottom in the middle (ok --> enter). We used conductor material(copper, aluminum) for the arrows to detect changes in electrical conductance when the person passes the hand over commands. The capacitive sensors are connected to an Arduino Uno that can process the signal to send to the computer interface allowing the navigation in Grid3 software.
2. **Hand-tracking camera:** the camera of the computer can detect the hand motion by sensors placed on the thumb and eventually on the index. The hand movement can be translated in the cursor position. Two methods can be used in the interface (HSV filter with a green patch on the hand, hand pose estimation).

# Interface

# A python-based interface was designed for this project. This interface is combining both solutions and allows interaction with the Grid3 software. The interface was allowing the choice of the controller type (Capacitive Sensor [sensor], Hand tracking [Mouse]) to navigate through the Grid3 interface. An arduino is automatically detected and connected to the arduino. If an arduino is connected, the interface is set up with the “Sensor” controller showing to the user 4 arrows (up, down, right, left) , 1 button of selection (circle in center) and another button on top left allowing the switch of controllers (sensor →mouse, mouse→sensor). These buttons are a visual feedback representing the capacitive sensors placed on the board in front of the user. The arrows and the button selection are colored in green when the user is touching the appropriate capacitive sensor. If the user pressed the checkBox “Mouse” or touched the capacitive sensor linked to the top left button, the hand tracking mode is activated turning on the webcam. Finally the button “Jouer'' opens a game designed during the hackathon (see Section Gamification).

# In the menu, the user can set up some parameters such as flipping the camera in x and y, flipping the cursor movement in x and y and finally choose what type of tracking: HSV filter (“Tracking Color”) or Hand post estimation (“Tracking\_DL”)

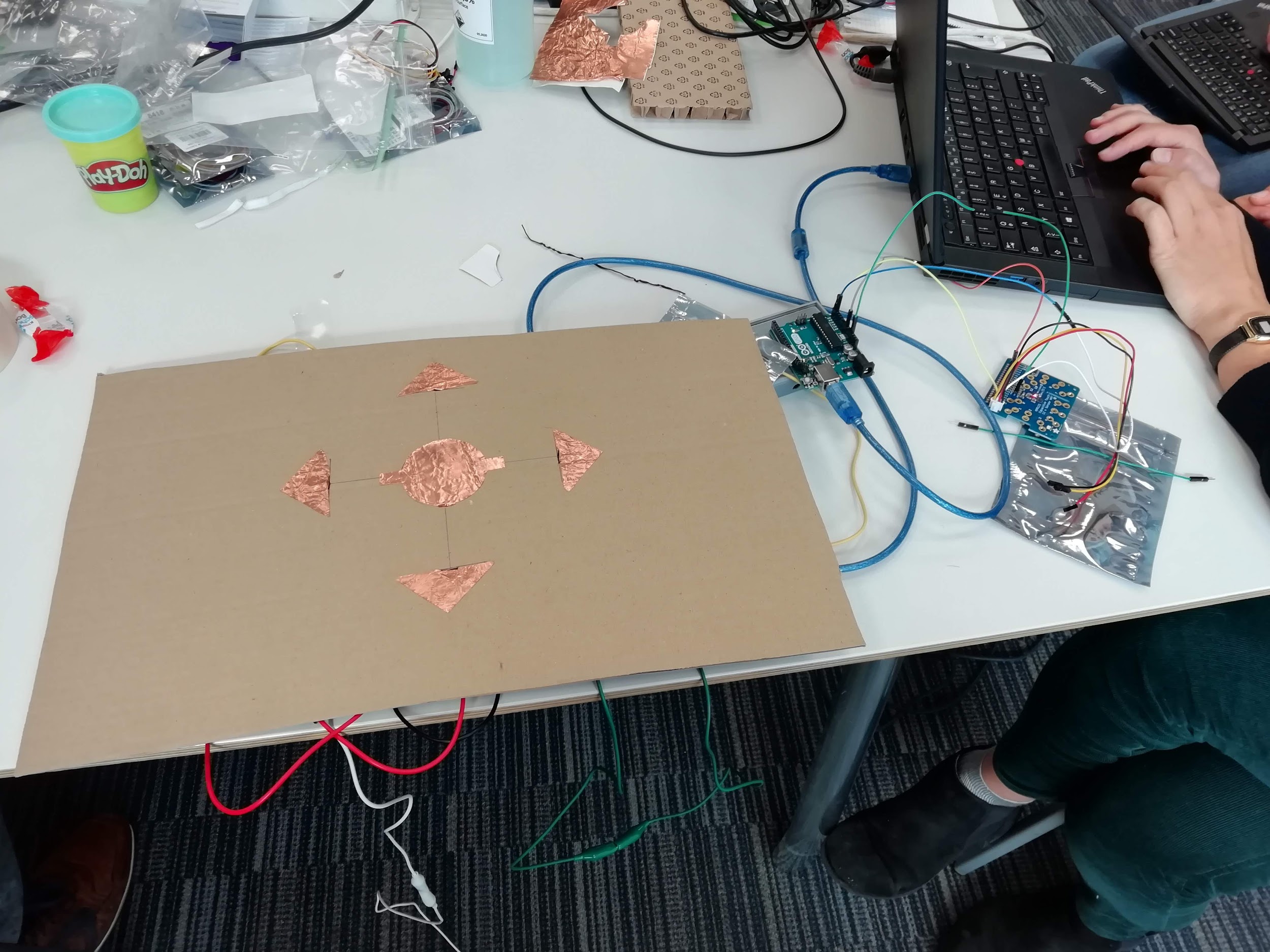
# Controller 1 - Capacitive Sensors

*Material:*

* *Arduino Uno*
* *Carte d'extension HAT tactile capacitive touch Adafruit* [*https://www.adafruit.com/product/2340*](https://www.adafruit.com/product/2340)
* *Crocodiles mini cables:* [*https://www.adafruit.com/product/1008*](https://www.adafruit.com/product/1008)
* *Aluminium for buttons*
* *Plexiglass Plaque*

Using capacitive sensors, Sohan could use some custom buttons to control the interface without any physical effort. Using the capacitive touch the user can interact with the interface without any physical force applied. The idea is to use the Arduino associated with a capacitive touch pad to sense all the contacts with an object.

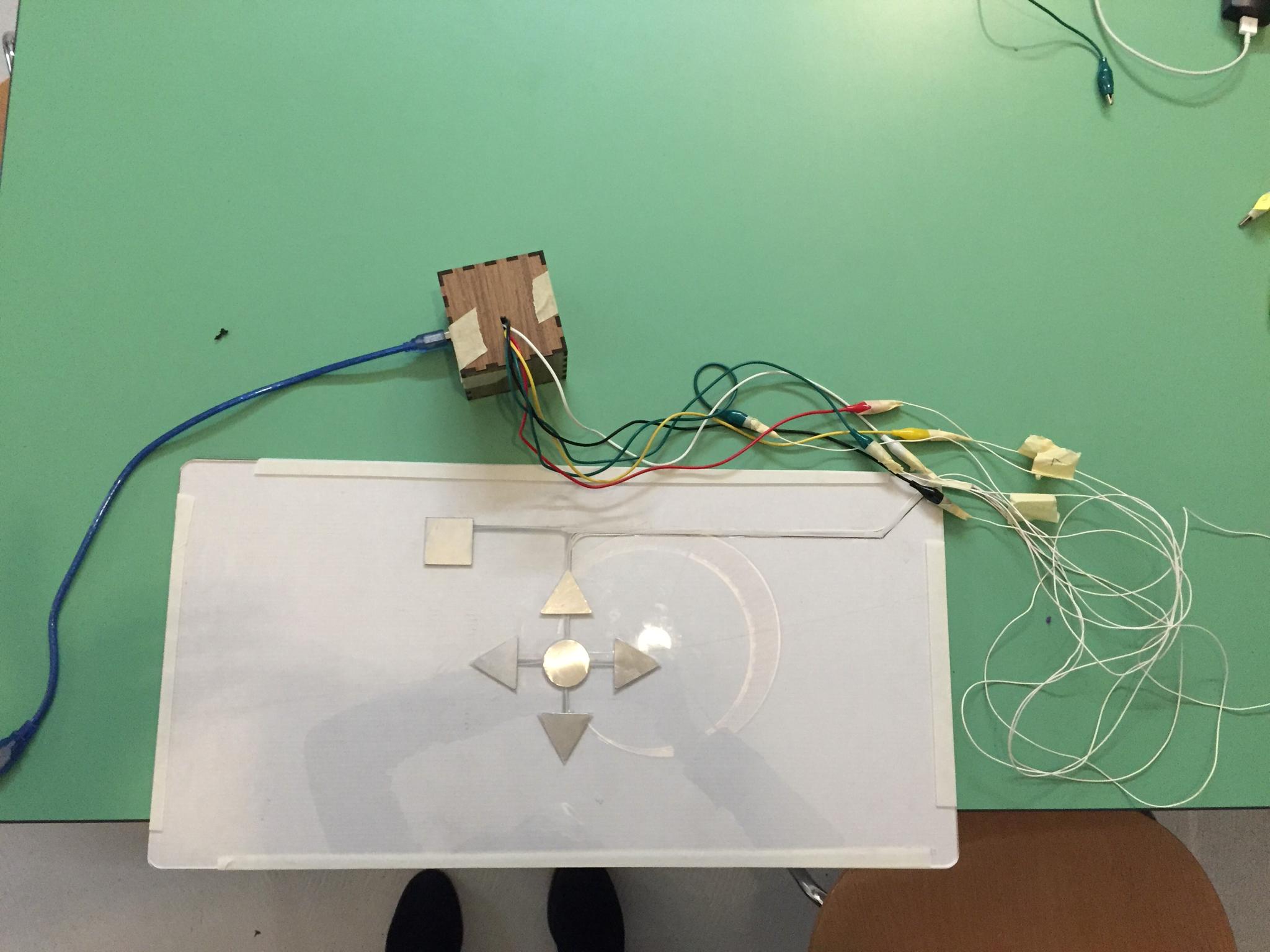
A first version of a support having directions arrows and confirmation button has been designed as first prototype, the prototype is shown below:



*Prototype done during the hackathon.*

The team glued thin copper film for the different buttons needed, and those films are linked to the capacitive pad using wires.The Arduino links the sensitive pads to the directional arrows buttons. The Arduino’s code can be found on ther github in the folder “Touch Sensor”. The step has been to associate this whole combination with the interface that Sohan is using to interact.

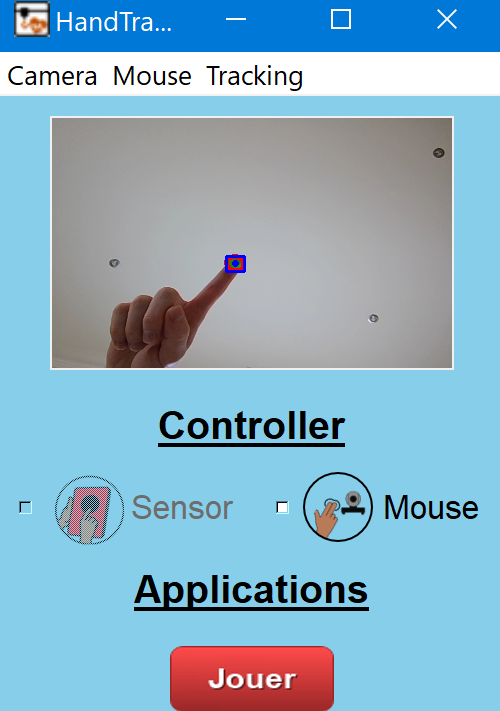
A second prototype was done after the hackathon to improve the design. This time, the team used plexiglass and aluminium as materials. A part of the cables was inserted in the transparent board. The arduino was also hidden in a wood box.



*Second prototype continued after the hackathon.*

**Limitation:** The present solution is intuitive and simple to be used for the user. A potential issue is that to switch off a button, the user might involuntarily touch another button (e.g. the button selection if going from left to right). This issue was solved by putting a glove to the user with only the index finger free to touch the buttons.

# Controller 2 - Hand Tracking with Image Processing (HSV Filter)



As the best way he has to communicate is through pictograms he points with the thumb on his left hand. One possible solution is to use Computer Vision in order to track Sohan’s hand. To do this, we detect a green patch placed on his thumbs with a webcam and use the position of this patch to control the mouse on Sohan's computer.

Main hand segmentation code with OpenCV in Python:

1. Detects a specific shade of green in the webcam image by thresholding HSV values (lower and upper bounds)
2. Perform morphological operations to extract the contour of the detected green shape.
3. Fix a bounding box and its center using this contour
4. Extract the x and y position coordinates of the green shape
5. Maps it to the screen to move the mouse.

Below, on the different windows we can see the masks generated as pixels labelled with 1. One is the opened version of the shape, another one is the closed shape and the difference of the two gives the contour of the original shape.

**Limitation:** HSV Filter is fast and accurate but can be compromised depending on the luminosity of the room. The detection could be improved by tuning the lower and upper bound of the green color using sliders that could be integrated on the interface). The position of the camera is also important and will impact the way the mouse is moving.

# Controller 3 - Hand Pose estimation with Deep Learning

Using the mediapipe library, the hand tracking solution was improved using a hand pose estimation and selecting one of the joints as the one controlling the mouse position. This solution presents the advantages of not requiring any patch to stick on the hand. This solution seems also to be more robust as the luminosity doesn't seem to affect the detection contrary to HSV Filter solution.

**Limitation:** The last solution would need to be improved, the detection of the hands seems to depend on the position of the hand in the camera with less accuracy when the hands is near the frame of the camera.

# Gamification

In the final part of the project, we designed and implemented a game to train Sohan to both types of control. Using Unity, we could design and implement a game that is easy to use and adapted for physical impairment. This game was based on an attractive design and graphics to catch Sohan’s attention and motivate him.

**Contact**

For any questions,please contact HackaHealth on our website <https://www.hackahealth.ch/>