# ‘Eye contact’

## 🔴 **Challenge Description**:

We are doing a functional awareness service. Our project will internally respond to an external stimulus in the environment. This will alert you of external dangers, getting the information needed such as vehicles, people around you or some obstacles in the way of the internal hardware (cameras, connection between dispositives) or looking to the future, using the data collected by **Smart Cities** or **Neuralink**.

## 🟠 **Team Contributions:**

⚡Most powerful ability of each member⚡

Matteo the code:Alchemist

Edu the Idea Generator

Charlotte the soul writer

Melchior the saturday night fever one

| | | |
|---|---|---|
| ★ Brainstorming | - | Edu, Charlotte, Matteo, Melchior. |
| ★ Project documentation | - | Edu, Charlotte. |
| ★ Python prototype | - | Matteo. |
| ★ Video | - | Edu, Charlotte, Matteo, Melchior. |

## 🟡 **Development Timeline:**

Day 1 : Brainstorming session and getting the coolest and the most accurate idea. Each group integrator collects info and research for possible issues to solve, and expand the idea the most possible for future implementations.

- ☐ - Environment awareness.
- ☐ - Subtitles (sounds , or translator).
- ☐ - Focus on the sound desired (person who is speaking or the music i'm listening to).
- ☐ - The dispositive will shake where the sound / danger comes from (left or right).
- ☐ - 360° cameras to "predict" what's around.
- ☐ - Zoom.
- ☐ - Hologram.
- ☐ Wireless connection (if someone has the same app, recognizes and does something useful) …

*From all these ideas we must choose only 1 or 2.

Day 2 :

Once we set the main idea and started seeing the project functionality and how it is getting formed, now we have to face some real problems.

- Is the project possible to perform? How do we implement the code in the hardware?
- Which kind of people help? Is it easy to use for everyone?
- Does our feature solve a need? Does it create more problems or benefits?

Making the code, we had to implement a python library to make our sensor system functional.
We are working too on a video-presentation.
We added a creative logo.

Day 3 : Presentation Day

Last but most important step is being sure everything is correctly updated and uploaded to the respectives repos, and Devpost.

## 🟢 Implementation Details:

Tracking of shapes (people, cars, etc.).
Different interactions when a shape is recognised. Minimize the actual app's windows or increase transparency.

## 🔵 Technical Stack:

- Webcam .
- Python , VScode, Google Drive, Cupcut, Linux screen recorder .
- Yolov 6.0 , Opencv-python 4.5.3.56 , Torch 1.9.0 , Ultralytics 0.0.45

## 🟣 Lessons Learned:

Technological Challenges:

- We learned about the real-time performance and ensuring low-latency processing for real-time danger detection and alerting -> our tests on the laptop worked very well and the latency was quite fast.

- We learned how to implement a python library. -> We looked for a way to detect shapes with the camera and recognise them. Make the application superpositioning consistent.

- The battery life might be very short considering the high power consumption of smart glasses, especially in combination with camera use -> Optimize energy use with low-power devices and smart data handling.

The weather problem:

- Lighting and environmental conditions (low light, glare, or weather interference might be problematic) -> light and glare might be solved with machine learning and for bad weather there could be waterproof case for the glasses, and anti-fog pad.
  Anti fog-pad allows you to prevent fogging when you do physical effort.

## ⚫ Future Possibilities:

- Scale by adding more sensors like LIDAR and cameras for **broader coverage.**
- Use **cloud computing** for better data processing and storage.
- Integrate with smart city infrastructure for comprehensive **environmental awareness.**
- **Share data** across multiple regions for improved safety and traffic management.
- Use eco-friendly materials for hardware and packaging.
- **Integrate AI** for better object recognition and predictive alerts.
- **Personalize** alerts based on user preferences.
- Support autonomous systems with **real-time** data.

---