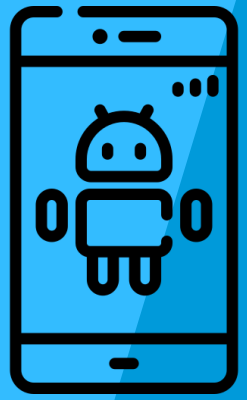


Manipulating videos and media remotely by using the Android’s motion sensors

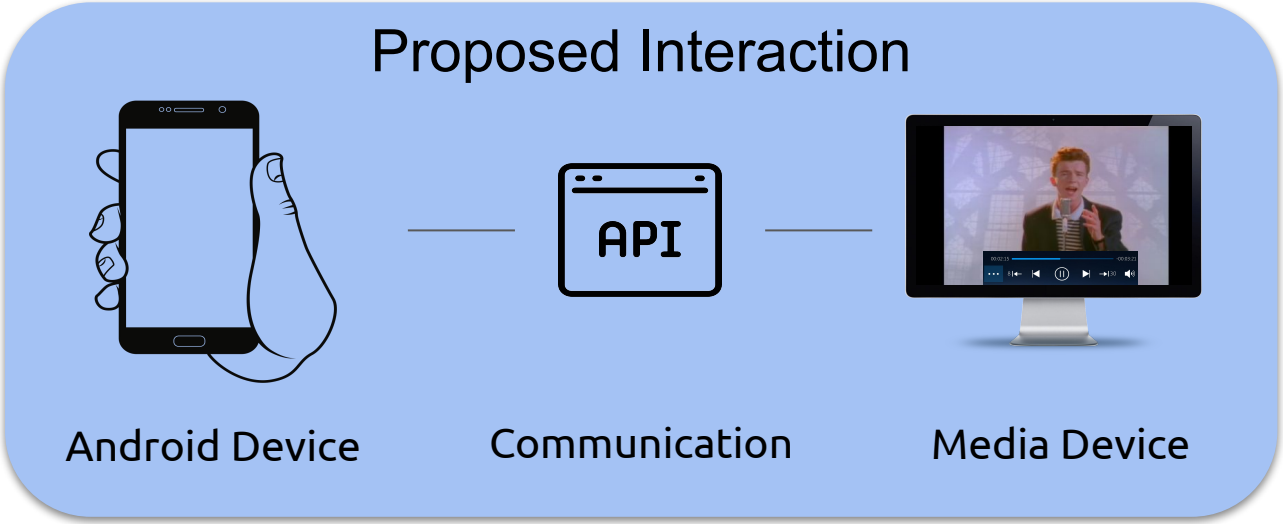
Archit YADAV
Sotirios TZAMARAS

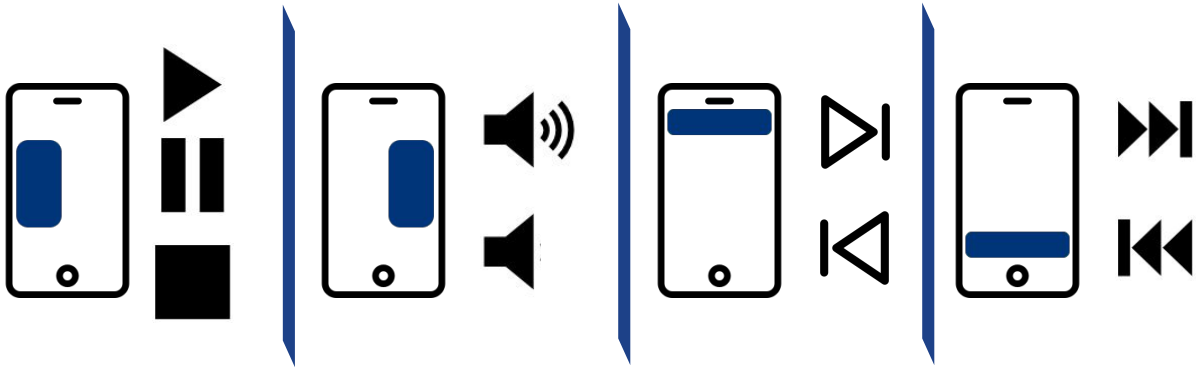
7 January 2022
Grenoble

Agenda

- Context / Problem Statement
- Prototype Design
  - Description
  - Demo
- Experimental Design
- Results
- Conclusion

Reminder


Proposed Interaction
API
Android Device
Communication
Media Device



Problem

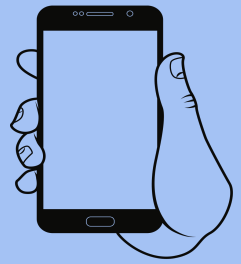
There are options for remote control of media devices (TV, computers)...
...but they have disadvantages.

1. Turn it on, unlock device, look for the buttons, press the buttons…
2. Pointing devices: Be precise, point at the buttons, click on them…

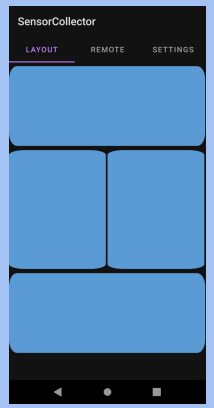
Worst case scenario

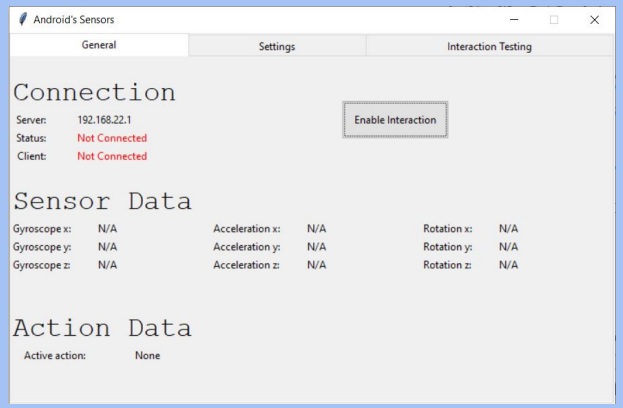
The user has to be close to the device to control it!

Proposed Interaction

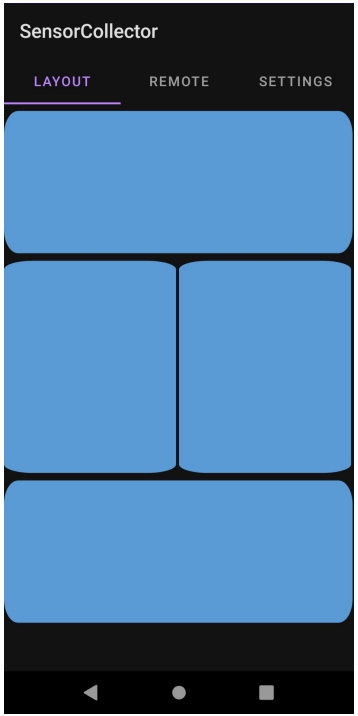


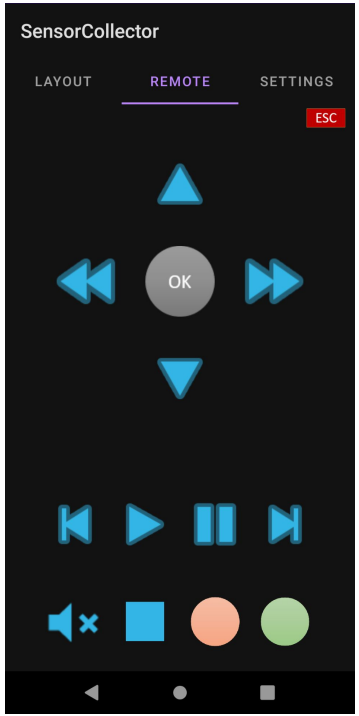
Android Device Communication Media Device

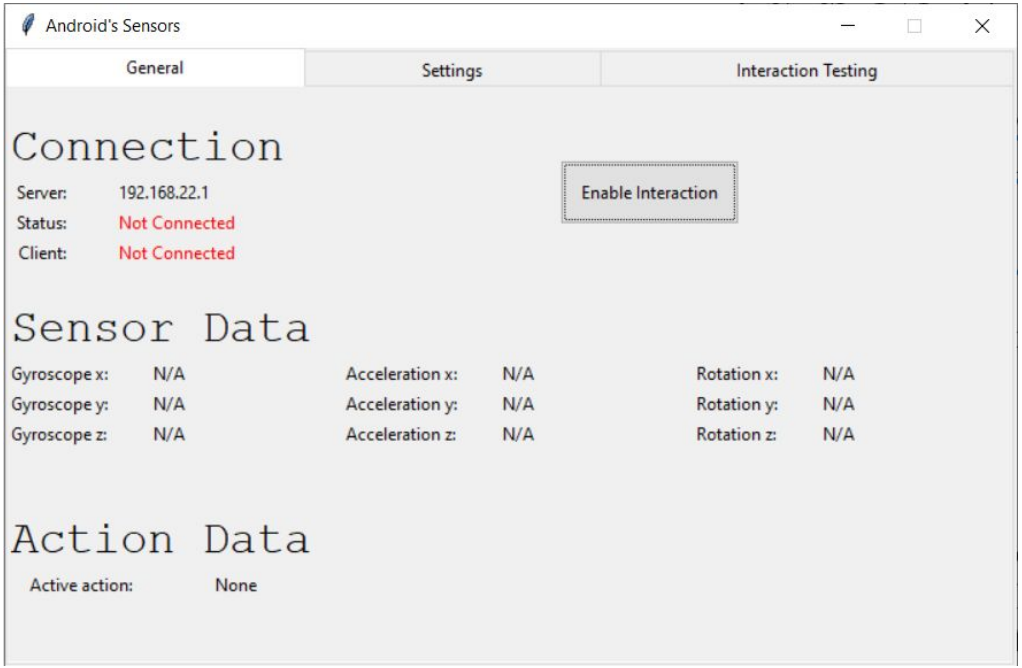




Prototype Design

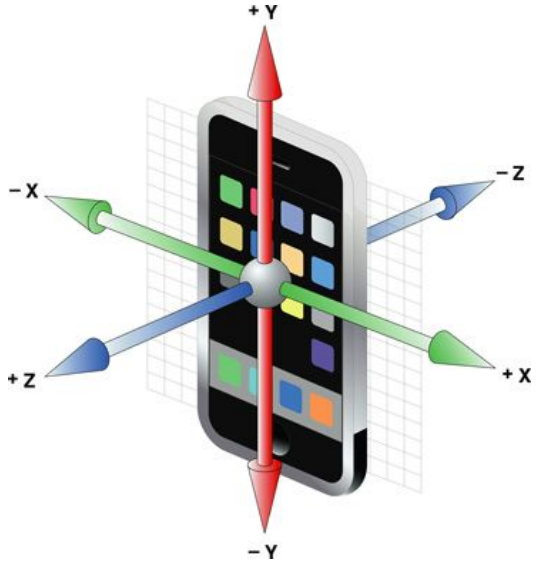




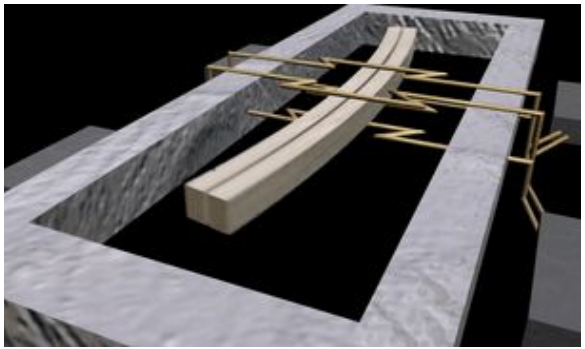


Prototype Design

- Accelerometer [Used] ✅
  - Piezoelectric-based MEMS sensor
  - Measures the acceleration based on **g**
  - Used to detect **orientation**
  - Android API gives access to **X, Y, Z**
  - **Simplest to use**



- Gyroscope [Not used] ❌
  - Angular rotation velocity or acceleration
  - Often Used along with Accelerometer



- Rotation Vector [Not used] ❌
  - Combines data from Accelerometer, Gyro and others
  - Based on Quaternion coordinates

https://www.youtube.com/watch?v=zwe6LEYF0j8

Demo

Experimental
Plan

The Experiments

The experiments are divided into 2 categories:

1. **Speed** Tests
2. **Interactive** Tests

Each category is tested twice:

1. **Layout** tab
2. **Remote** tab

In total there were 4 set of tests.

Experimental
Plan

The Experiments

Speed Tests

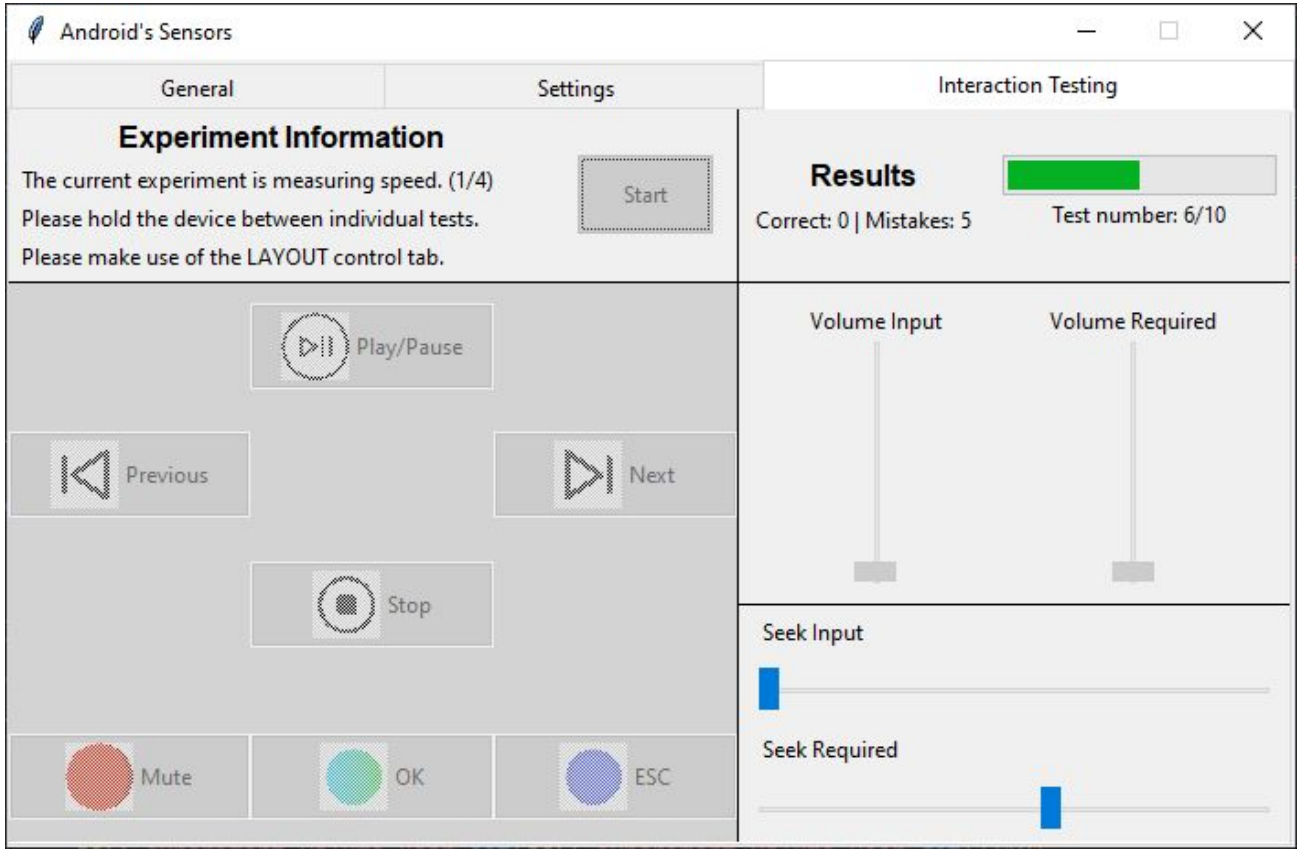
Compare the **amount of time** the user needs to accomplish an interaction.

Interactive Tests

Compare speed in a more **natural way**. An example is controlling the television. The user needs to **pick up the control device**.

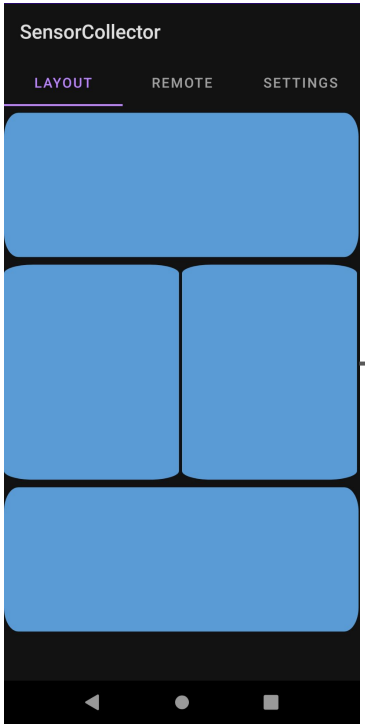
Interaction

Testing

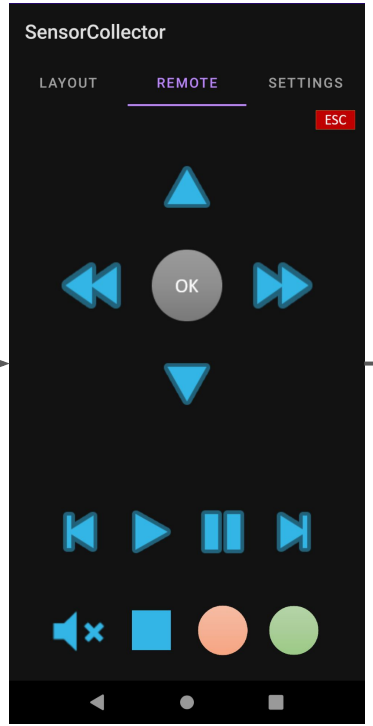


Speed Mode

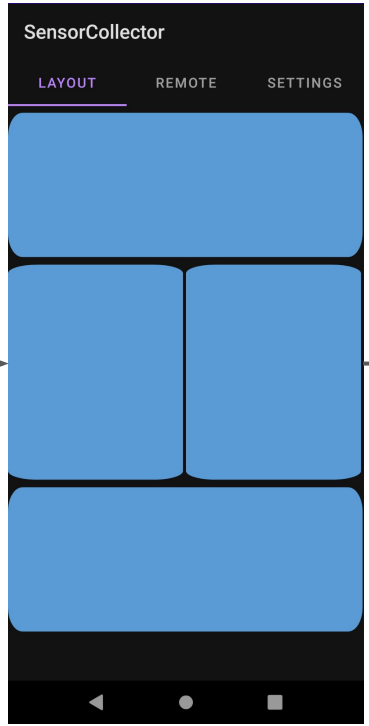
Interactive Mode



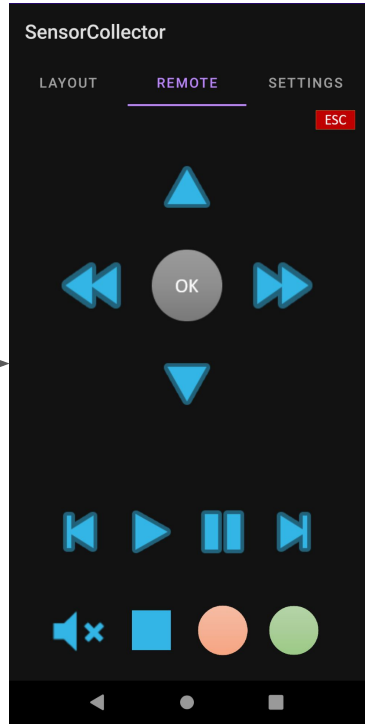
5 sec break



5 sec break



5 sec break



10 tests

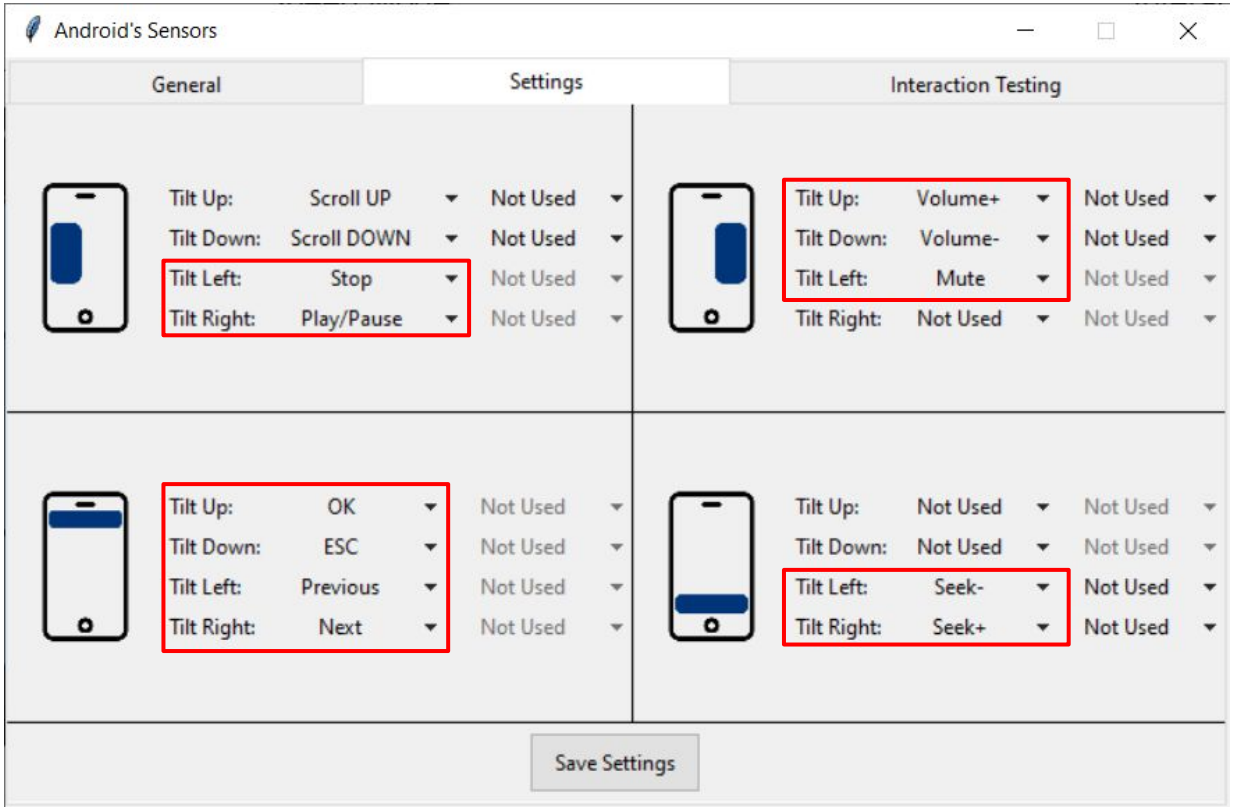
10 tests

10 tests
Rest between each test

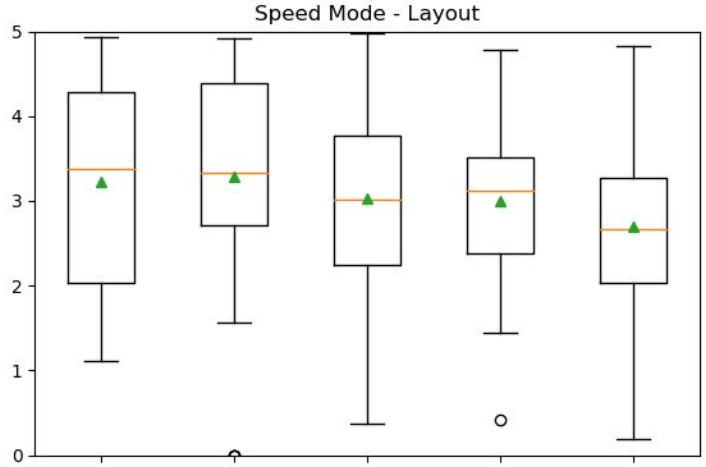
10 tests
Rest between each test
Screen lock

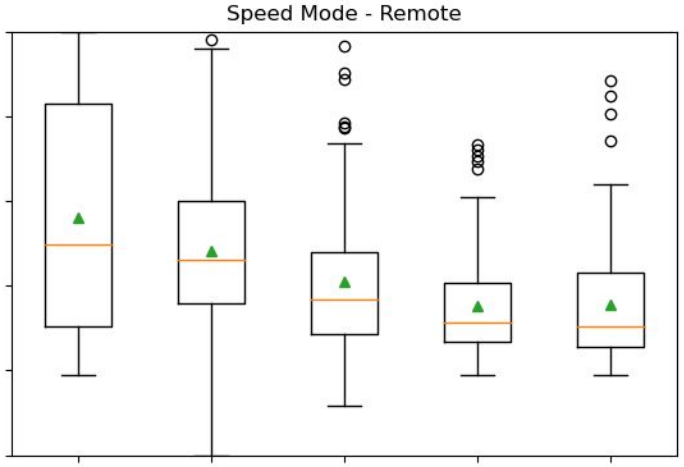
Settings

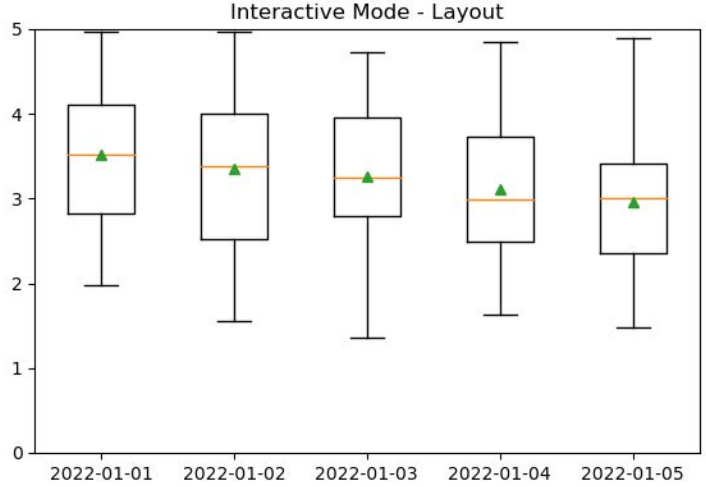
Used for experiments

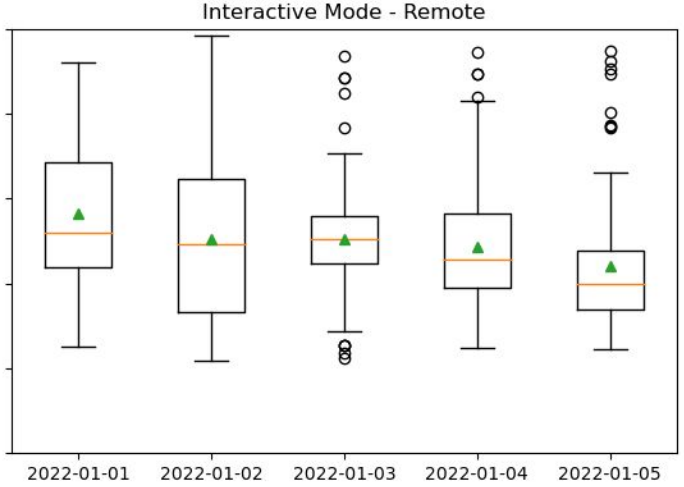


Results


Speed Mode - Layout
0
1
2
3
4
5

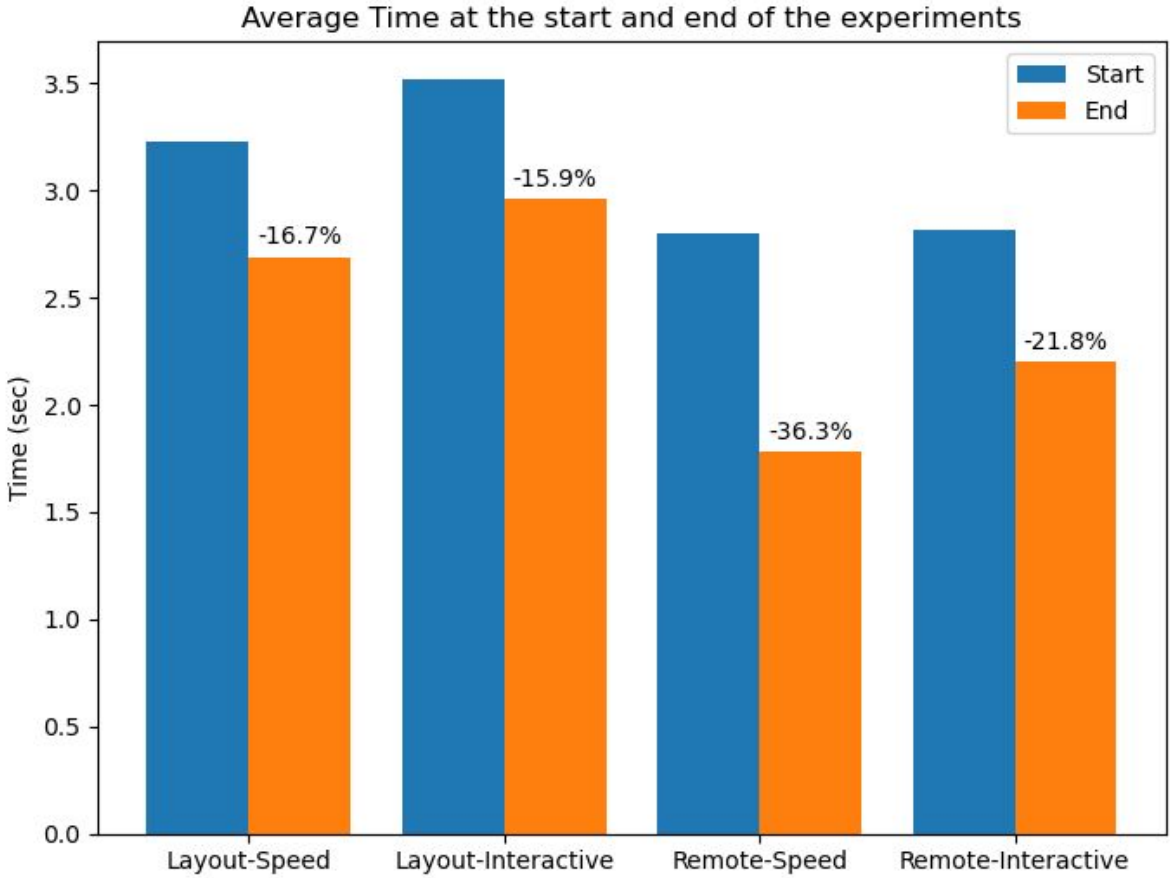

Speed Mode - Remote


Interactive Mode - Layout
0
1
2
3
4
5
2022-01-01
2022-01-02
2022-01-03
2022-01-04
2022-01-05


Interactive Mode - Remote
2022-01-01
2022-01-02
2022-01-03
2022-01-04
2022-01-05

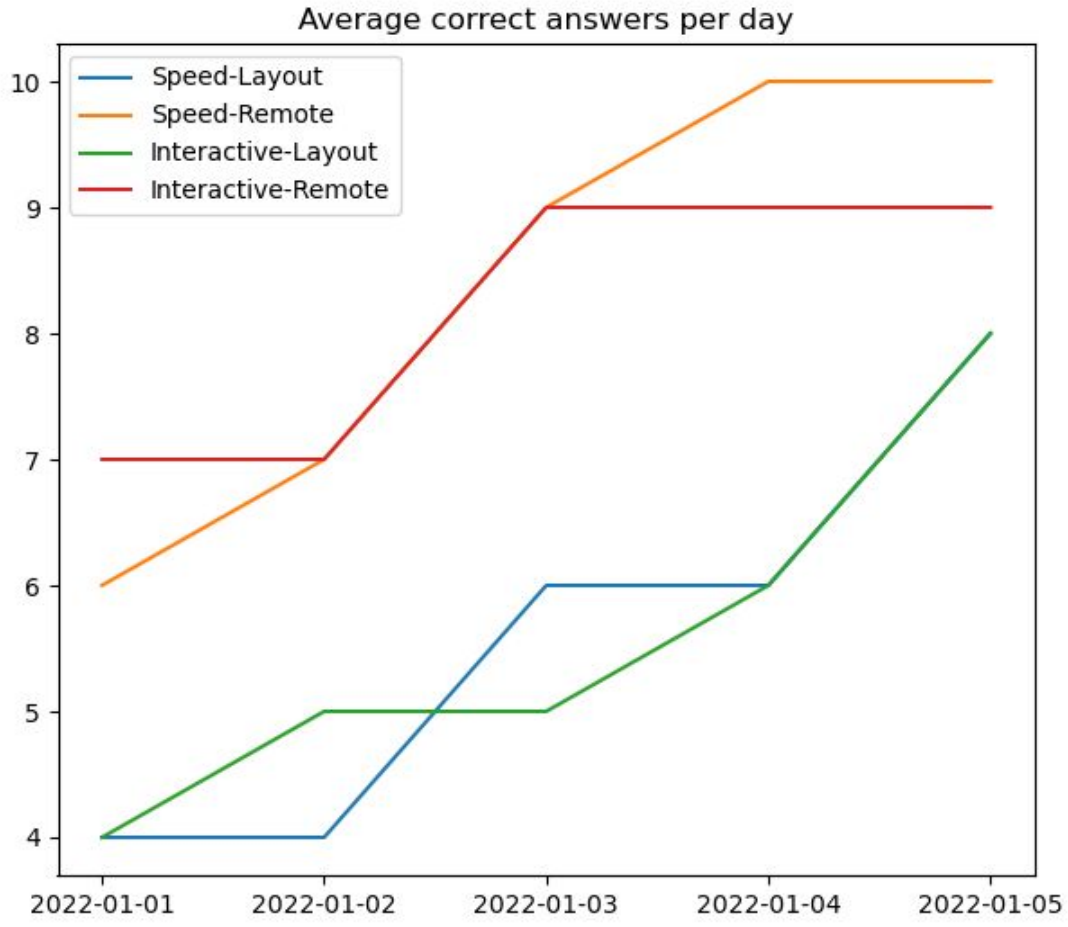
Results

Plots and figures



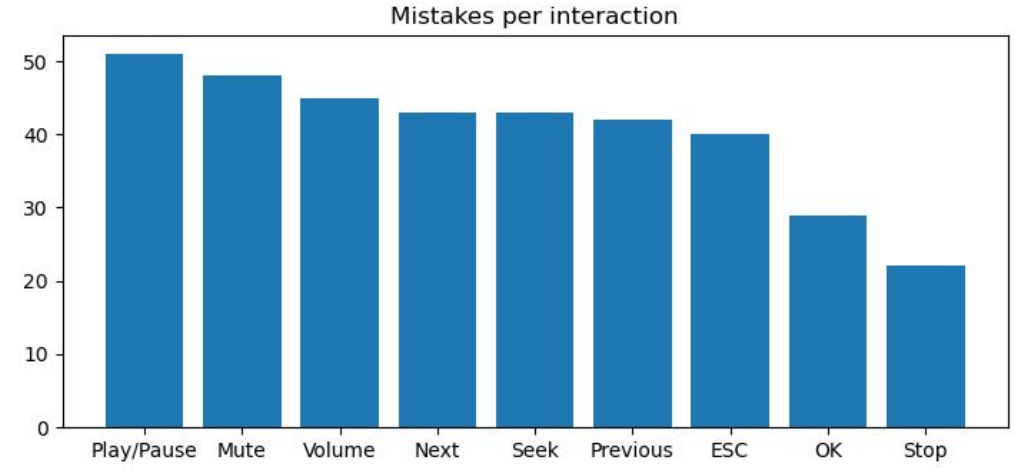
Results

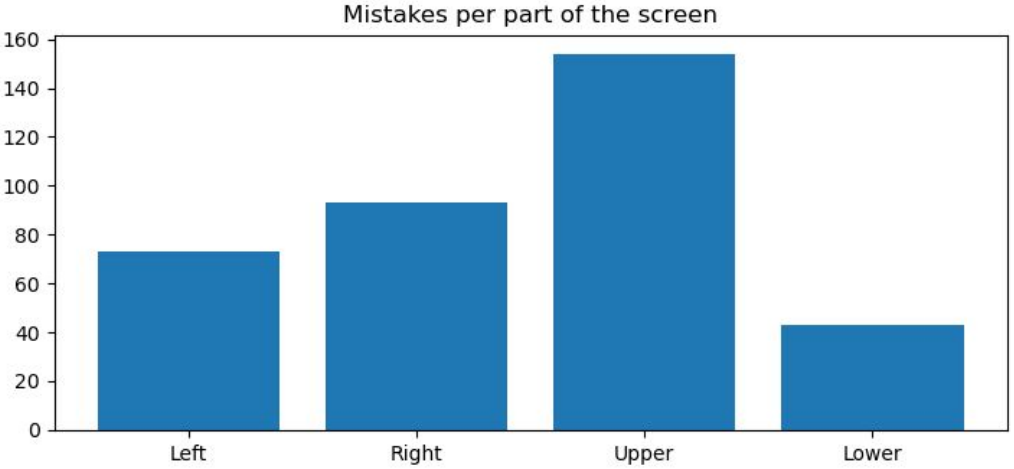
Plots and figures



Results

Plots and figures

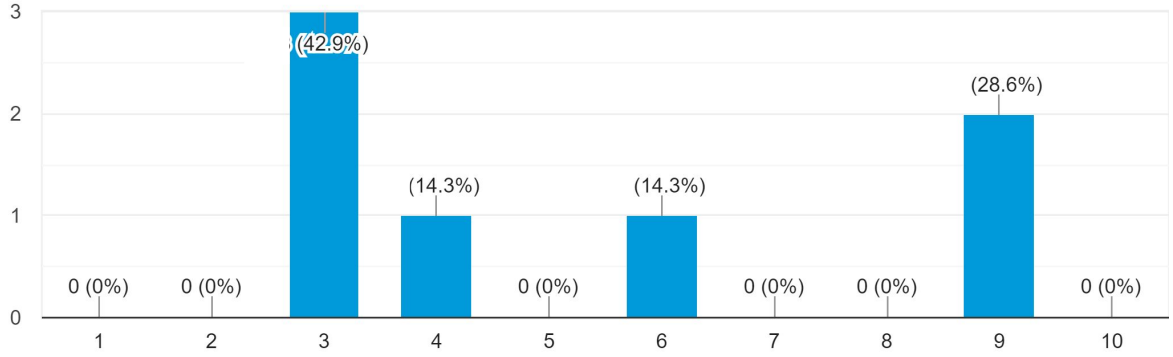




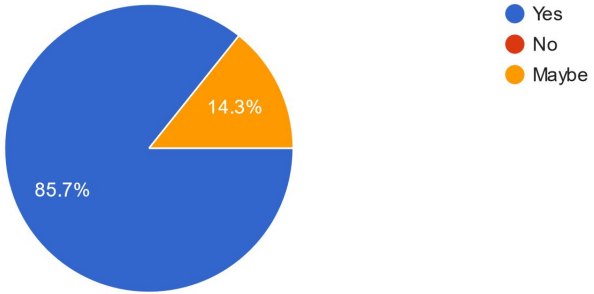
Results

Survey

How would you rate the difficulty to learn our new interaction?



If the Android app was programmed to have been functioning in the background while in locked screen mode (screen turned-off), would you still be able to locate the 4 rectangles?

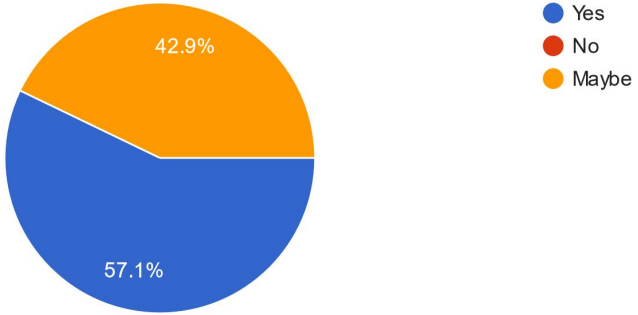


Survey Link

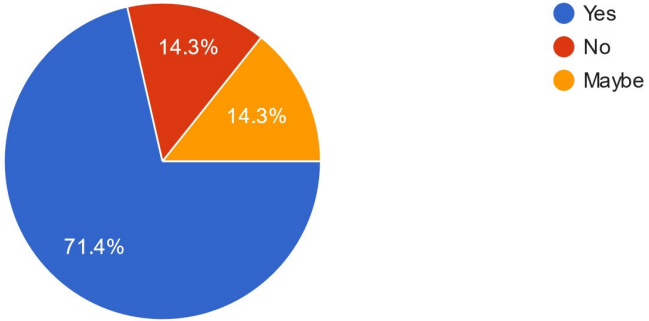
Results

Survey

Do you think that if you were allowed to change the settings to your liking, it would be easier to remember?



Do you think you could replace the standard remote-like applications to this one?



Results

Survey

In which scenarios can this approach be used?

- Video and music control
- Presentations
- Smart home (e.g smart lighting)
- **There is potential for more!**

Discussion

What can be improved

- ➔ Better sensor manipulation (for example by using gyroscope or the more advanced rotation vector sensor)
- ➔ Improve time of identifying the interaction
- ➔ Add support for different ways of continuous interaction (for example steps vs dynamic)

- ✓ Easily extensible to support specific APIs for control (for example VLC)

Discussion

Overall we think the results seem pretty promising!

With just a few proper modifications the behavior can be improved greatly!

Thank you!

https://github.com/sotostzam/sensor-media-control