

# DSCE

## Dynamic Soft Control on the Edge

Team 6

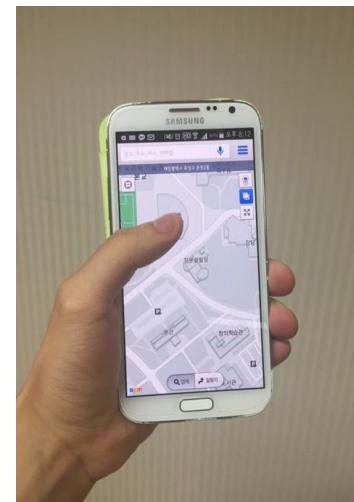
20110451 Park Yeonghoon

20120833 Jeong Gyeongjun

20130450 Lee Minho

# Background & Motivation

- Sometimes interacting with touchscreen is not very easy, requires user to hold and control the smartphone in uncomfortable way.
- For example, there is an occlusion problem when a user tries to move forward/backward a video.
- People sometimes have to control their phones with just one hand. There are many gestures which require two hands, such as zooming in and out.



# Introduction

- Goal
  - Get a slide/touch input on the side of mobile phone
  - Introduce some new gestures
  - Evaluate how user experience is improved
  - See if it solves occlusion problem.
- Approach
  - Attach touch sensors, integrate with modules
  - Get input via arduino, send data to Android device via TCP socket
  - React differently based on foreground apps

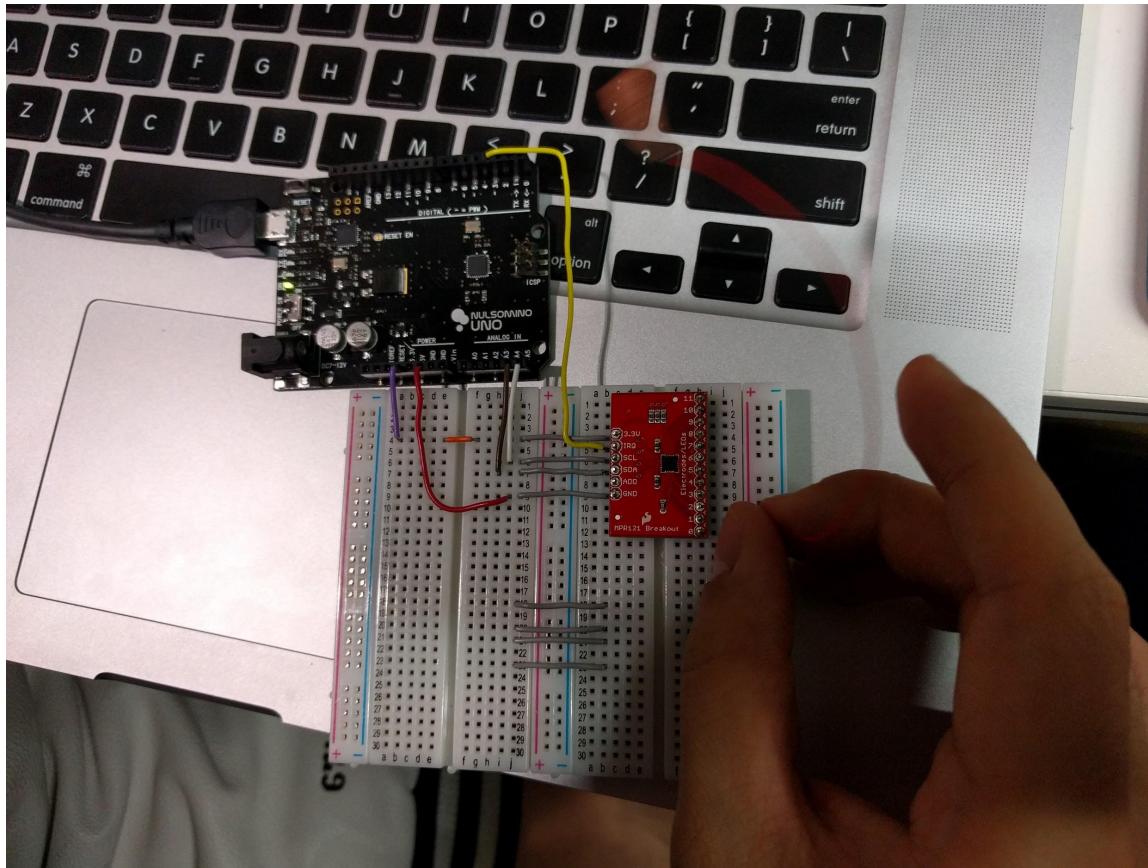
# Plan and Progress



	W8	W9	W10	W11	W12	W13	W14	W15	W16	else
Hardware Works										
Develop Algorithm										
Make demo app										
User Evaluation										
Make video / paper										

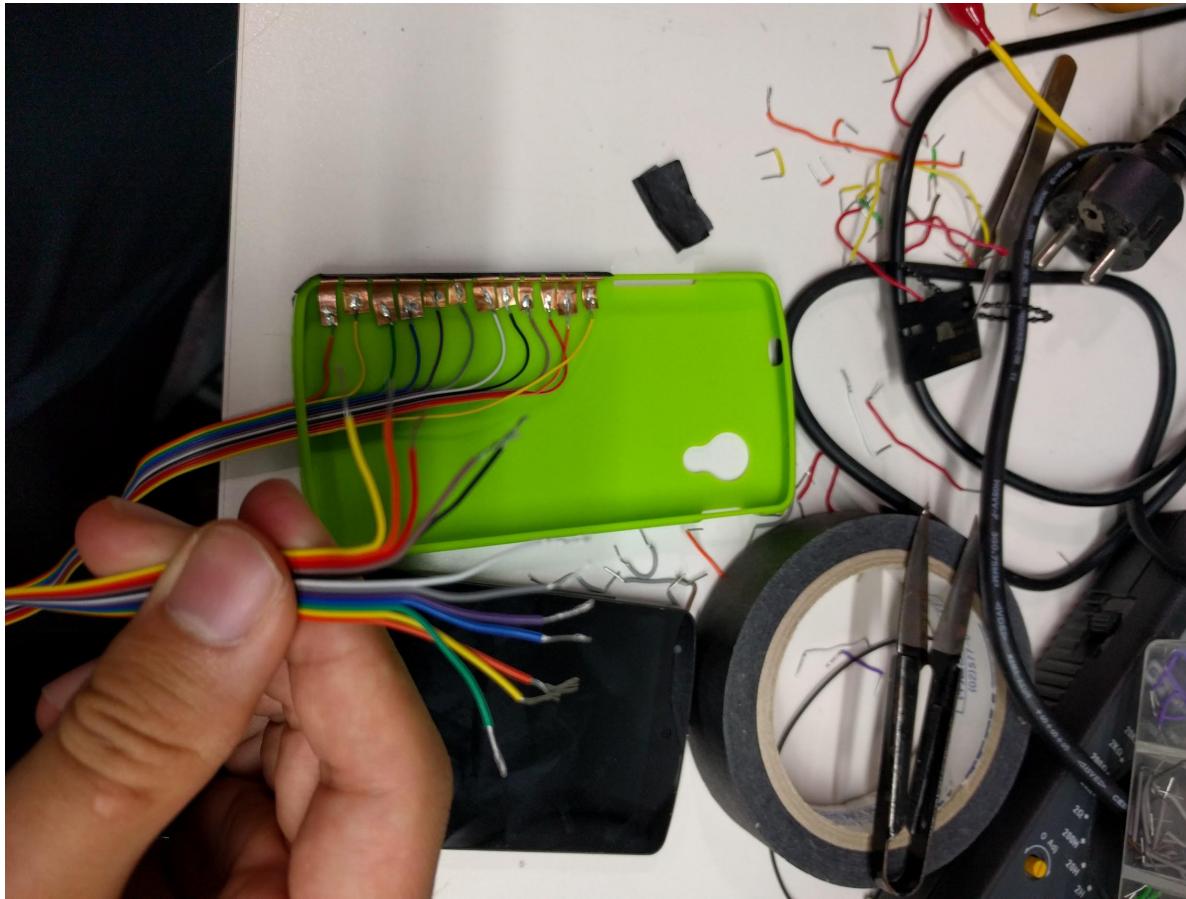
█ : Currently done

# Hardware Works



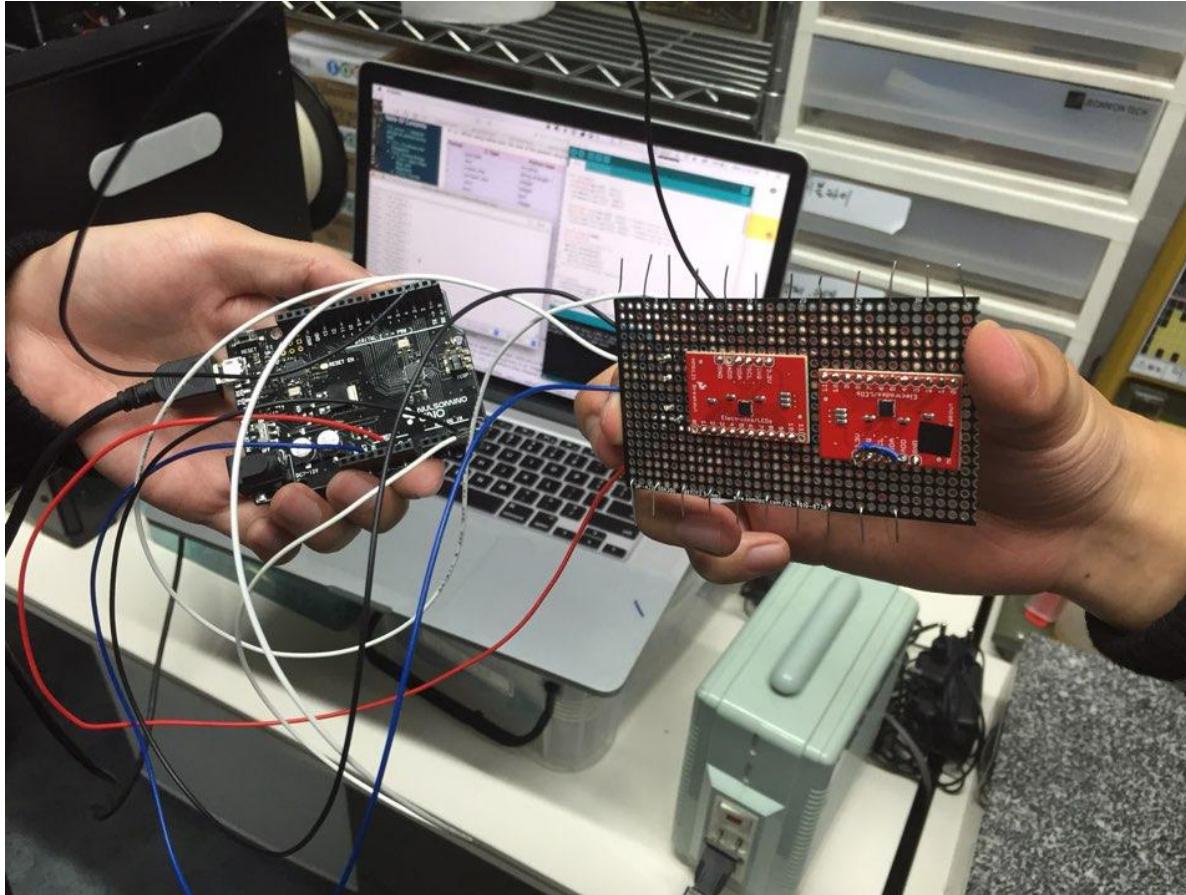
Touch Sensor Module  
Arduino

# Hardware Works



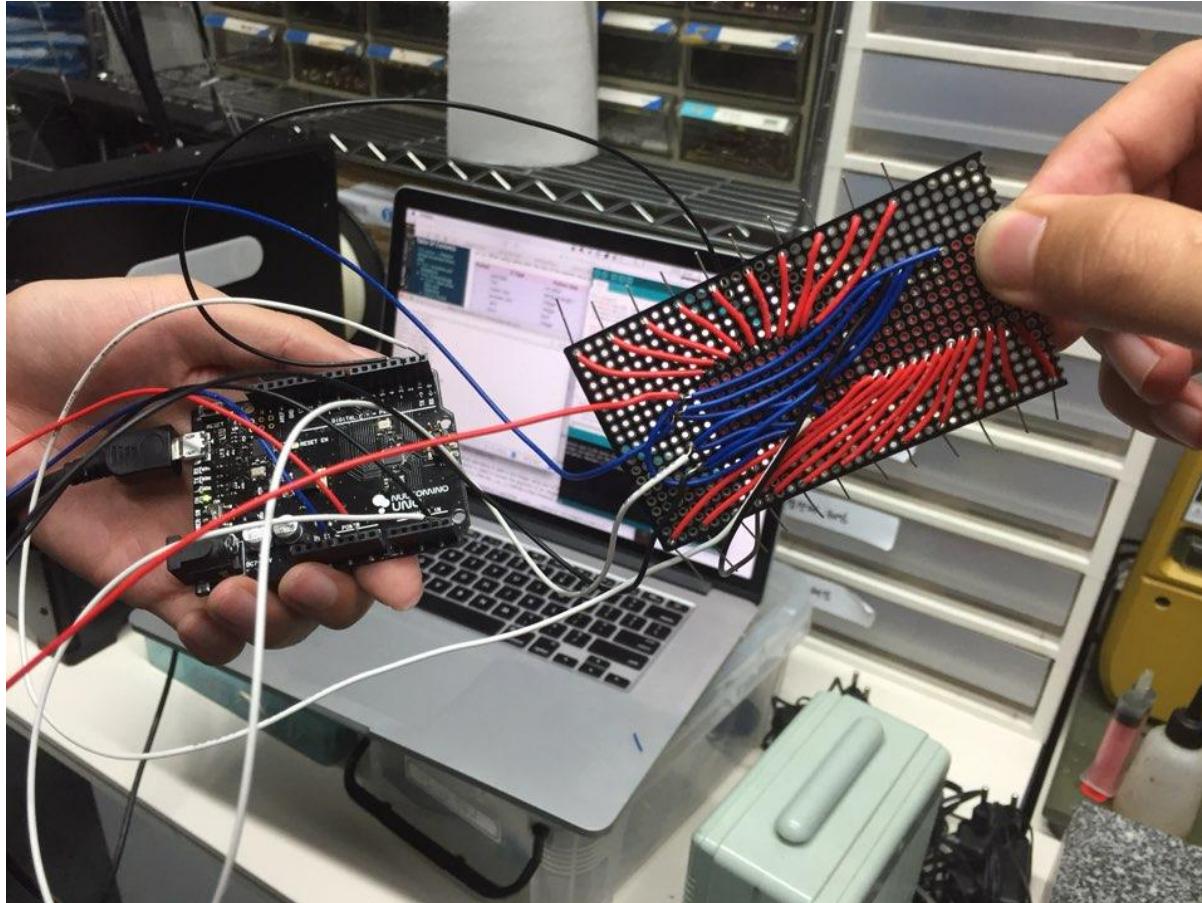
Sensors attached to phone case

# Hardware Works



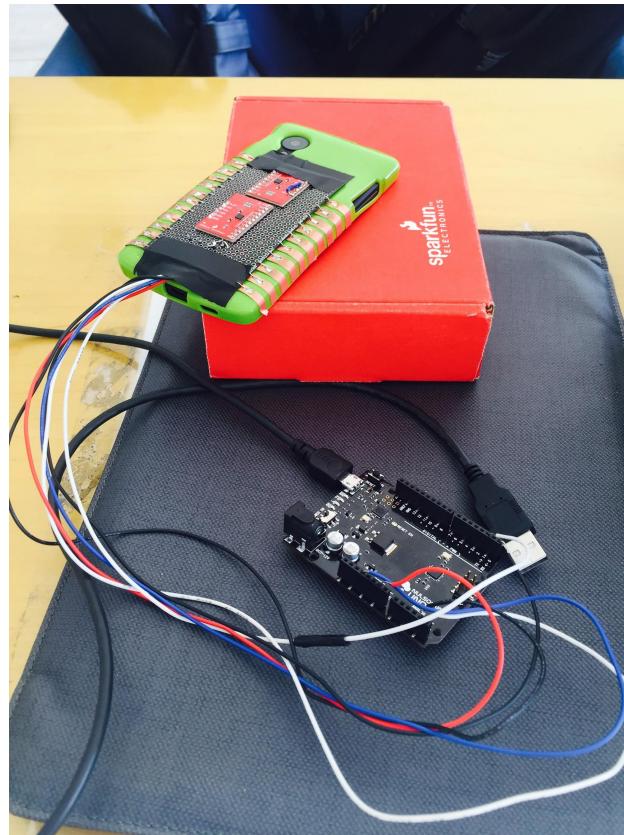
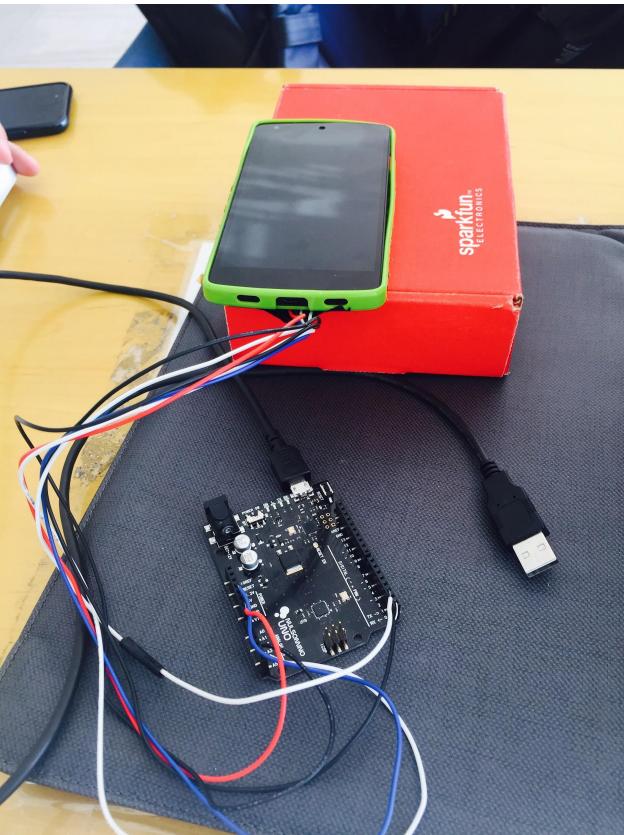
22 sensor with modules(front)

# Hardware Works



22 sensor with modules(back)

# Hardware Works



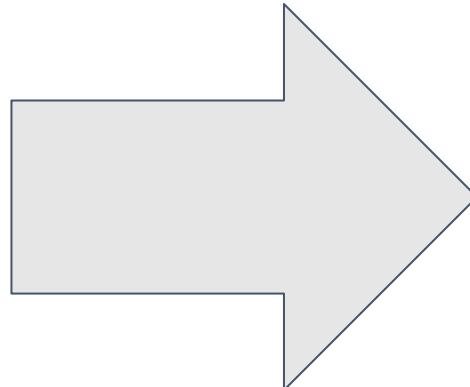
Final Prototype

# Gesture Detection Algorithm

- Input data : Arduino serial port(touch/untouch data)
- Raw touch data -> Slide / Click
- Used pySerial and text processing
- ~~Complex use like 'grab and click' or 'click with slide'~~

---

```
!untouch 8 from sensor 1
!untouch 6 from sensor 1
!untouch 7 from sensor 1
!touch 1 from sensor 2
!untouch 1 from sensor 2
!touch 2 from sensor 2
!untouch 2 from sensor 2
!touch 3 from sensor 2
!touch 4 from sensor 2
!untouch 3 from sensor 2
!touch 5 from sensor 2
!untouch 4 from sensor 2
!untouch 5 from sensor 2
!touch 2 from sensor 2
!untouch 2 from sensor 2
```



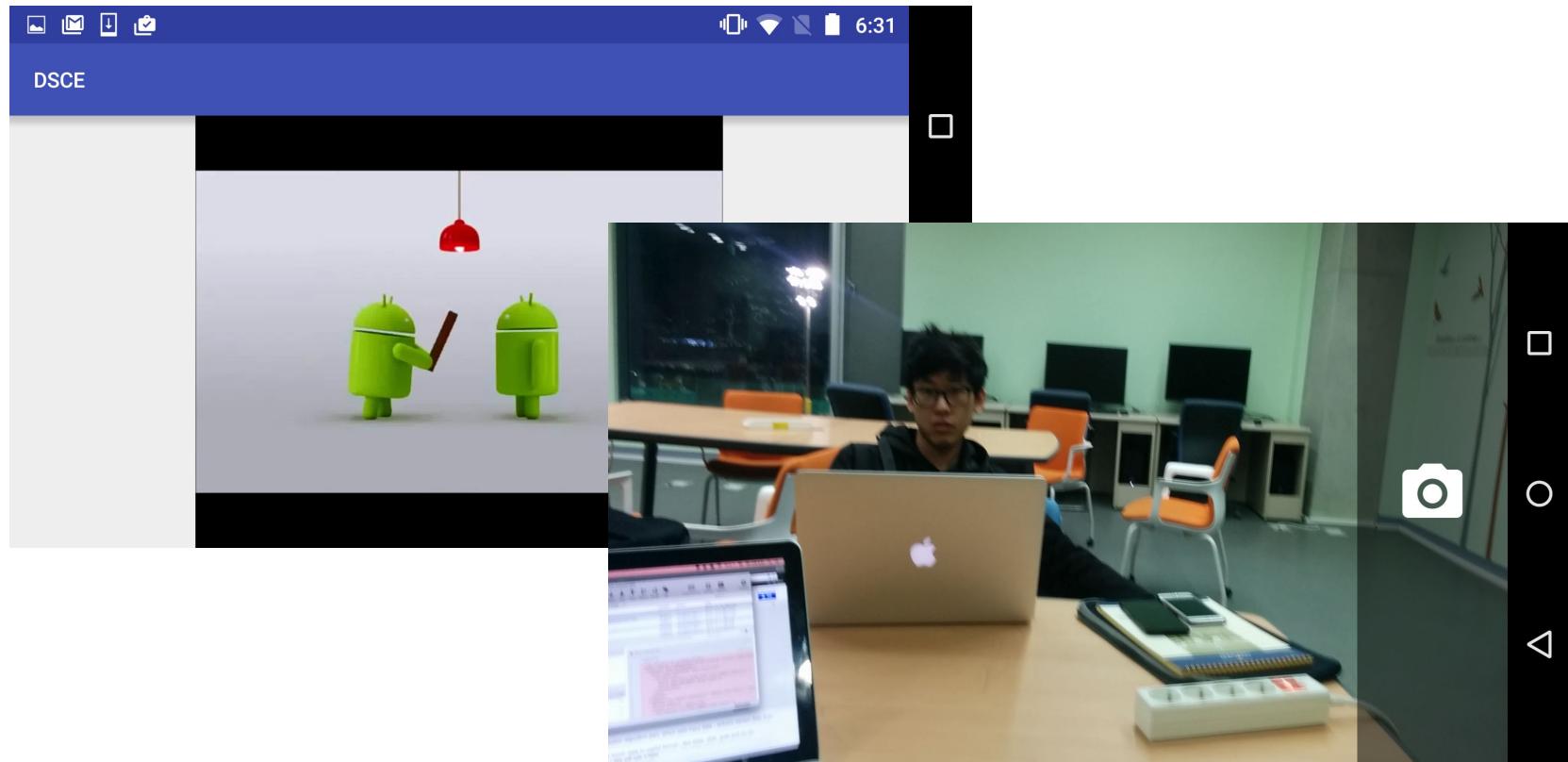
raw data

```
left side slide up
left side slide up
left side slide up
right side slide down
right side slide down
right side slide down
right side slide down
click left 6.500000
left 5 grabbed
left 6 grabbed
left 1 grabbed
left 2 grabbed
```

processed

# Sample Applications for Evaluation

- Video Player, Camera application



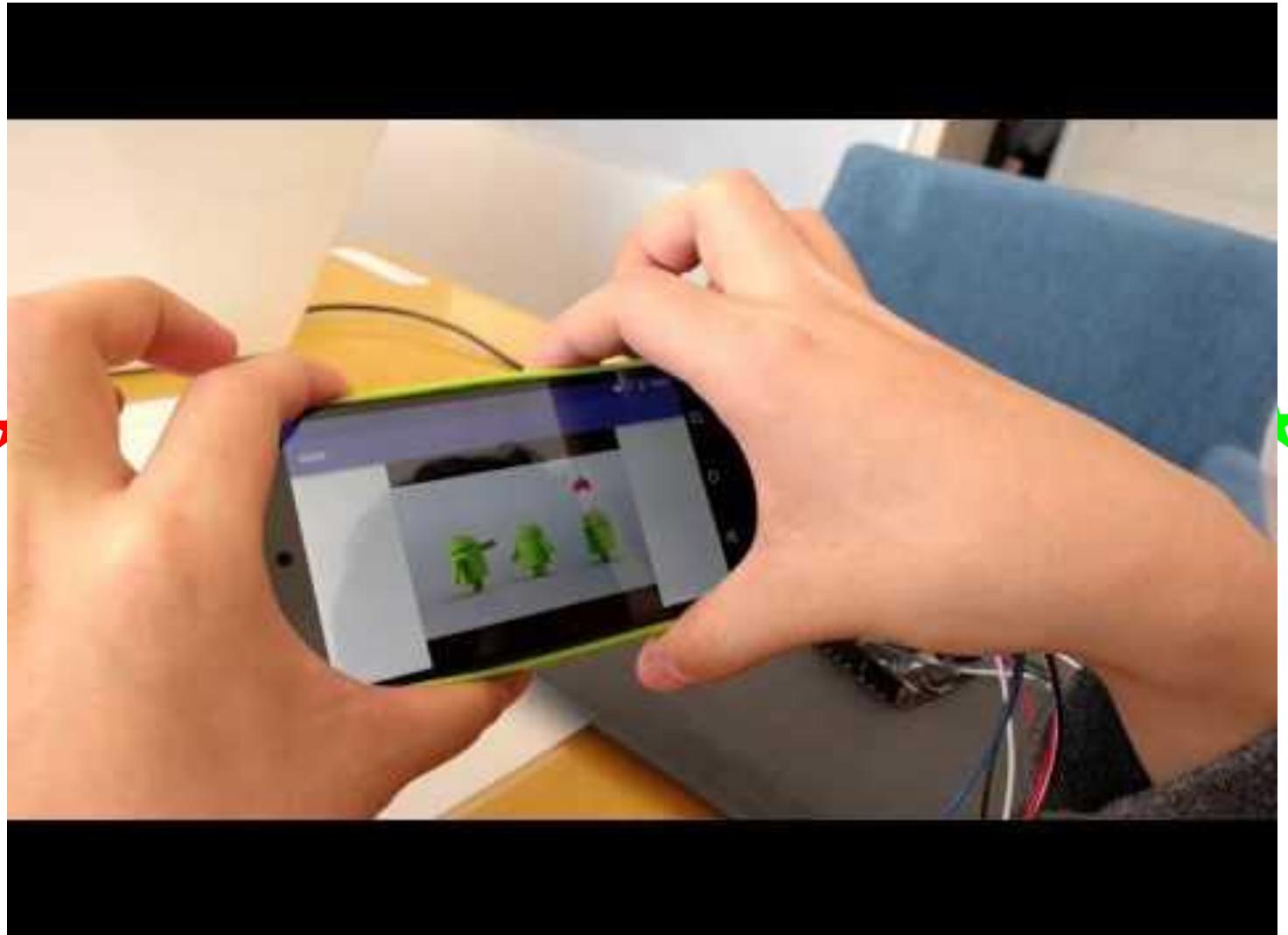
# What we intended to do & What we have done

- **Ideal work**
  - Users can use our module with any grabs they want
  - Our app doesn't change, and translate gestures to different way based on the currently running app.
- **Real work**
  - Users could use our module only with restricted grab.
  - Provided video player, camera demo application.

# Self Diagnosis on Prototype

- Attached too many sensors
  - Lots of sensors in unused area
  - Unexpected noises
  - Users had to grab phone in uncomfortable way
  - Eventually, we decided to ignore signals from unused area.
- Gesture detection algorithm problem
  - Not sufficient noise handling (Heuristic)

## Using Prototype & User Evaluation



# User Evaluation Method

- **Usability Test**

- Users used camera and video player application.
- Users are asked to do some tasks with and without DSCE.

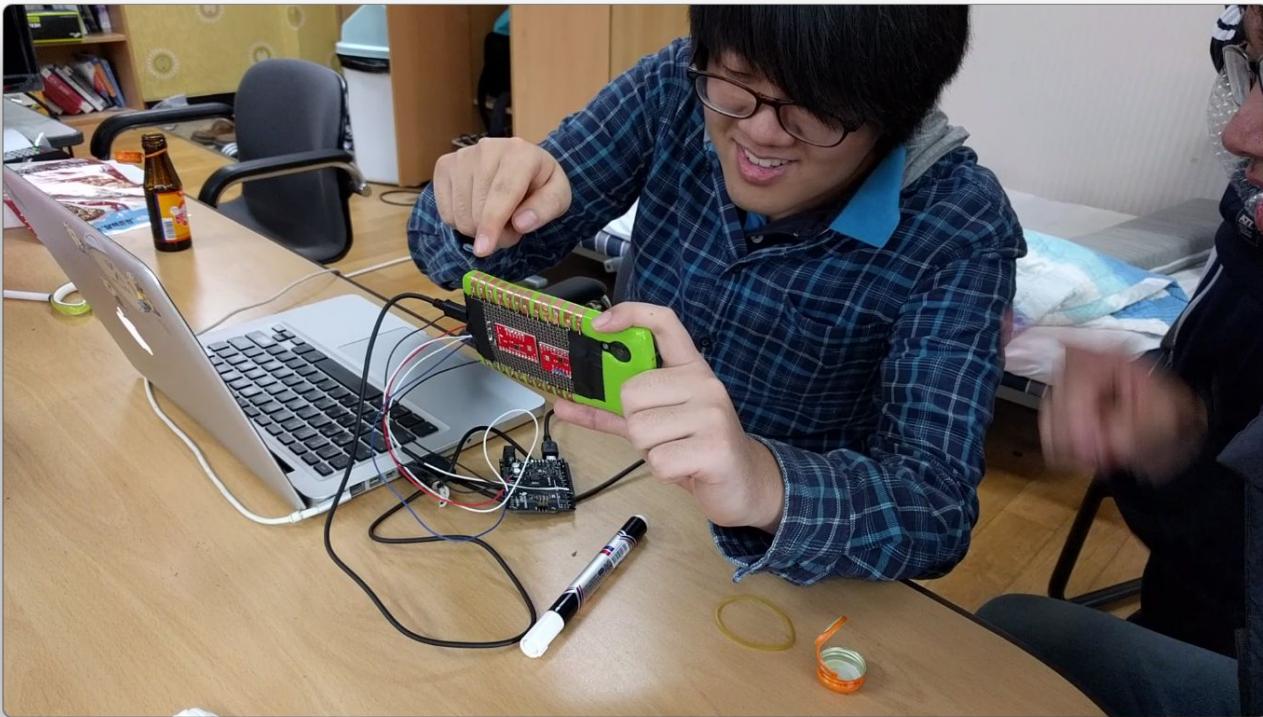
A. Video Player with DSCE	A-1. Control a video to go forward/backward to view specified screen by slide gesture.
	A-2. Pause / Resume a video by click gesture
A'. Video Player without DSCE	A'-2. Control a video to go forward/backward to view specified screen by dragging soft control on the screen.
	A'-2. Pause / Resume a video by button click
B. Camera with DSCE	B-1. Take a photo (shutter) by click gesture
	B-2. Zoom in / zoom out by slide gesture
B'. Camera without DSCE	B'-1. Take a photo (shutter) by button click
	B'-2. Zoom in / zoom out by pinch zooming

# User Evaluation Method

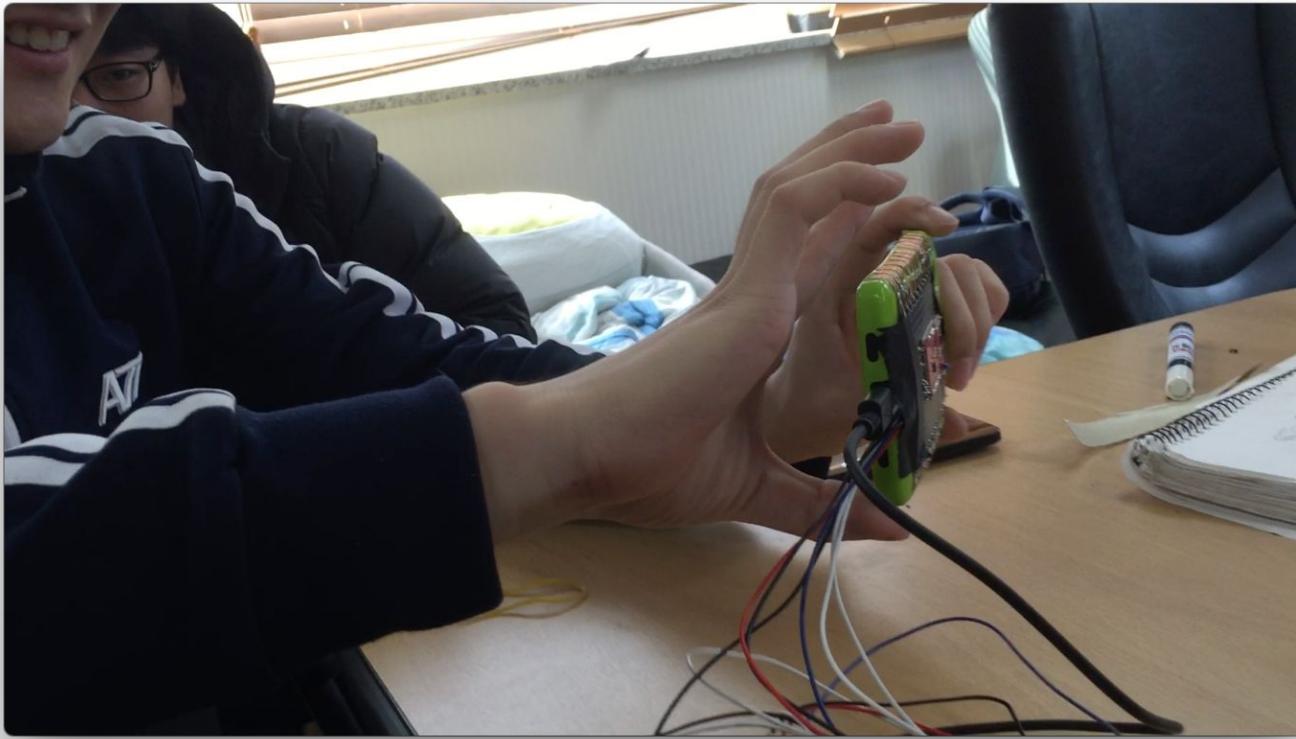
- **Questionnaires & Interview**
  - 8 Questions capturing the subjective ratings in likert scale.
  - Some questions are referenced from PSSUQ\* Questionnaires.
  - Users gave additional comment about their experience.

Simplicity	It was simple to use this system
Efficiency	I was able to complete the tasks and scenarios quickly using the system
Learnability	It was easy to learn this system
Occlusion	Fingers didn't overlap the screen
Usability	Using the interface was better than existing apps
Expectation	The system had functions and capabilities I expected
FUN	It was fun to use the system
Overall satisfaction	I'm satisfied with this system

# Evaluating User



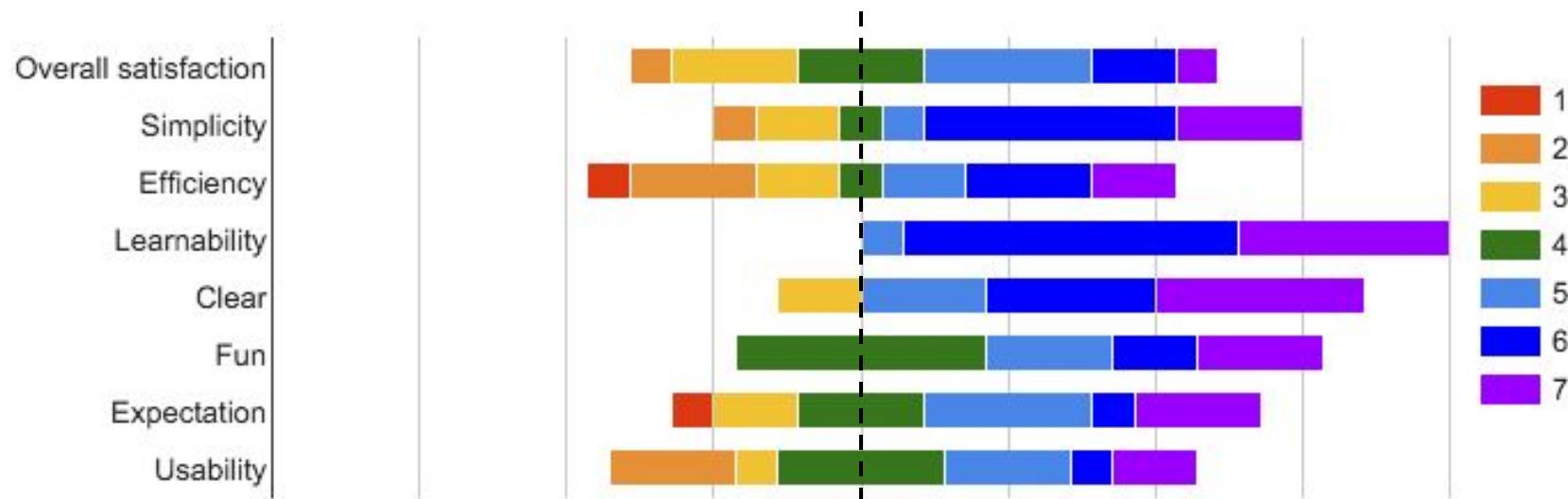
# Evaluating User



## User Evaluation Result

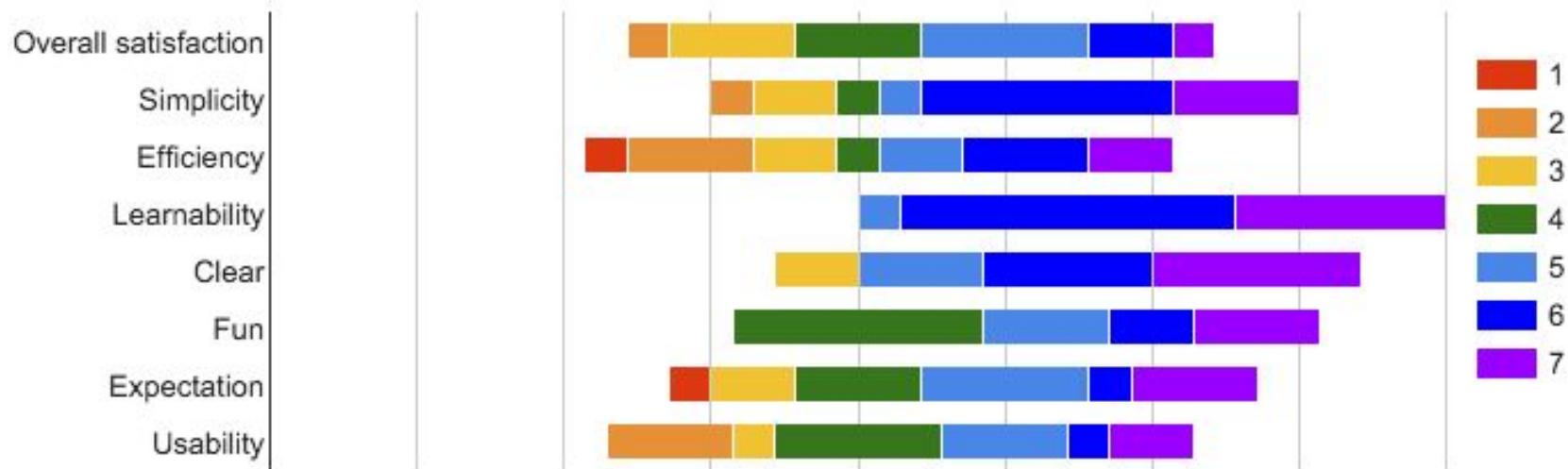
- Total 14 subjects took part in the experiment
- Most of the participants were KAIST students

# User Evaluation Result



# Analysis

- Participants gave relatively high scores at Simplicity, Learnability, and Fun.
- Participants agreed that DSCE solves the occlusion problem.
- Users did not either agree or disagree that DSCE had high usability or efficiency.



# User Comments

- **About prototype(hardware) issues**
  - “Copper sensors are excessively attached.”
  - “Copper sensors hurt”
  - “Since it is prototype, this is acceptable.”
  - “The locations of sensors were uncomfortable.”
  - “There were noise during experiment”
- **About gesture & interaction**
  - “There should be more functionalities.”
  - “It was easy to understand, but hard to use precisely”
  - “I think the screen occlusion problem is surely solved”
  - “Since its prototype user experience wasn’t good enough but understandable”

# Conclusion

- We have introduced a dynamic soft control on edge.
- We wanted to provide additional gestures depending on foreground apps.
- While building our prototype, there were lots of hardware challenges.
- DSCE solves occlusion problem.
- It is proven to be easy to learn and fun to use.
- But we're not sure if it is efficient enough.

# Q & A