



# Mousey: A Wireless Mouse App for Android

Lakota Morris, Patrick Fox, and Danny  
Chen

Senior Design Project II course Instructor: Paul I. Lin, Professor of  
Electrical and Computer Engineering Technology  
May 2, 2014

## Outline

- Introduction
- System Design Overview and Research
- Software Design
- Unit Testing and System Integration
- Project Management
- Conclusion

# CHAPTER 1

## Introduction

### Problem Topic

- Android Application
- Wireless Mouse
- Project Goal:
  - Alternative to touchpad applications
  - Eliminate the need to carry a mouse

## Background

- Many “touchpad” applications
- Innovative implementation:
  - Mouse moves in response to physical movement of the smartphone
  - First of its kind

5

## Methodology

- Analyze useful open-source applications
- Manipulate existing information
- Develop additional algorithms
- Integrate everything into final system
- Paired Programming

6

# CHAPTER 2

## System Design Overview and Research

### Feasibility

- Not as easily controllable as a physical mouse
  - Accelerometer noise
- High power consumption
  - Use for brief periods or charge while using

## Design Process

- Obtaining data
  - Fused Sensor
- Correcting data
  - Forward/backward drift
- Sending data to server

9

## Legal Aspects

- Open-source code from RemoteDroid and BokiSoft
  - Both released under GNU General Public License v3

1

0

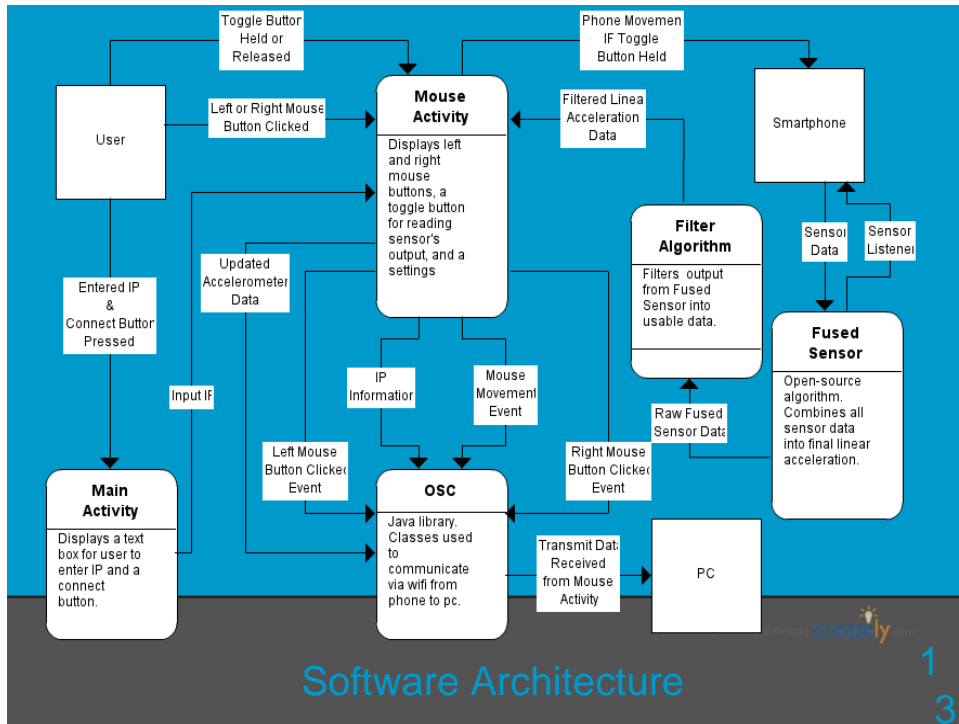
## System Scope

- Data is sent and received via Wi-Fi
- Uses the phone's built-in sensors
- Algorithms manipulate sensor data into usable data

1  
1

# CHAPTER 3

## Software Design

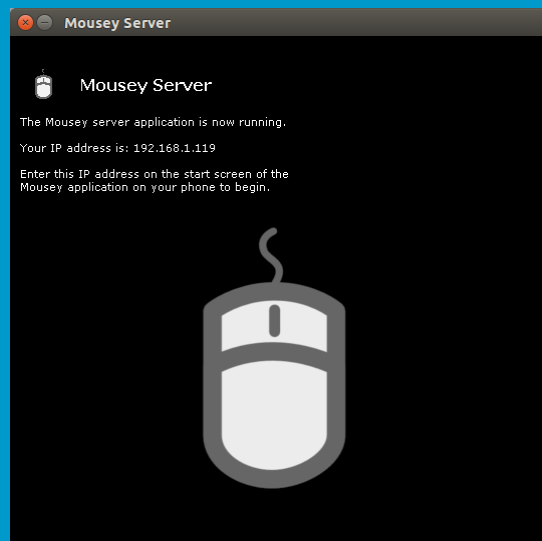


## Programming Language and IDE

- Programmed in Java using Android API/Libs
- Integrated Development Environment
  - Android SDK
  - Eclipse IDE with ADT plugin

## Main Components

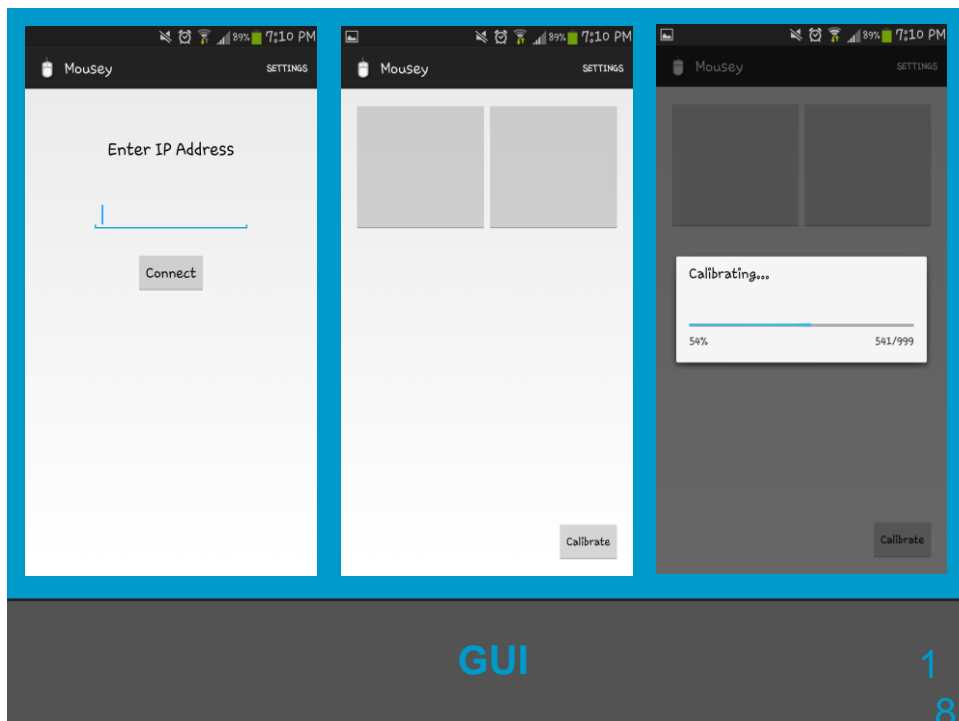
- Server
- Application
  - Network connection
  - Fused Sensors
  - Calibration
  - Filtering
  - Data interpretation algorithm
  - Physics algorithm
  - GUI

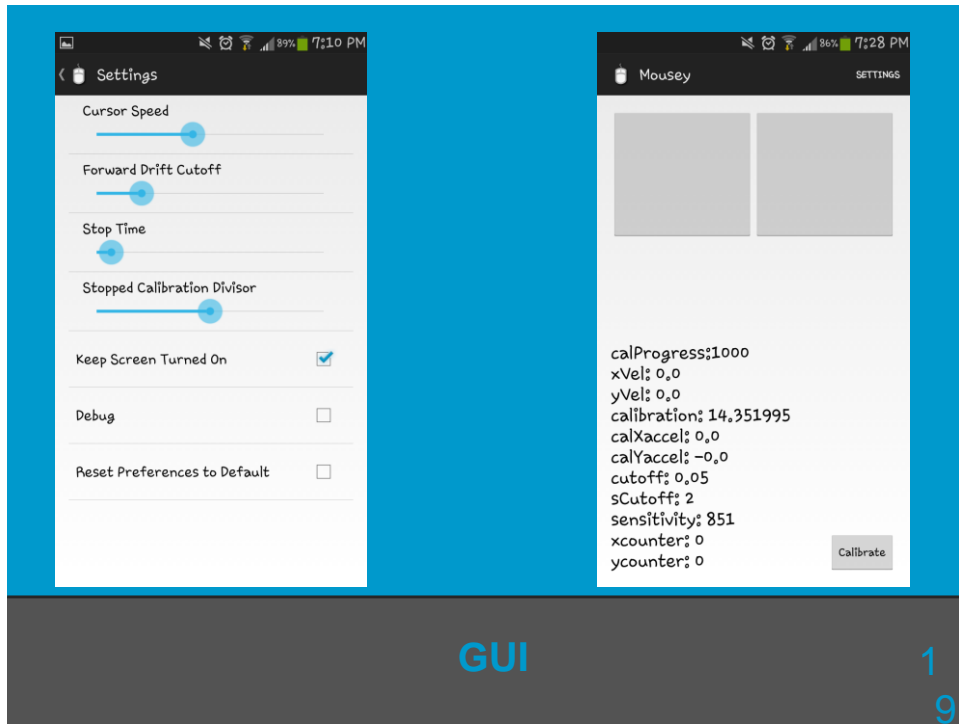
1  
5

Server

1  
6







## CHAPTER 4

### Unit Testing and System Integration

## Software Testing and Validation

- Connection is made
- Left-click and Right-click
- Click-and-drag
- “Touch-to-move” function

2  
1

## System Integration, Testing, and Validation

- Changes
  - from Bluetooth to Wi-Fi
  - from constantly active to “touch-to-move” function
- Moving the mouse
  - extensive testing with acceleration, velocity, and a variety of different filters

2  
2

# CHAPTER 5

## Project Management

### Schedule and Time Management

- 11 October, 2013
  - 28 October, 2013
  - 25 November, 2013
  - 27 November, 2013
  - 16 December, 2013
  - 27 January, 2014
  - 24 February, 2014
  - 27 March, 2014
  - and
  - 2 April, 2014
  - 14 April, 2014
  - Evaluation
  - 21 April, 2014
  - 2 May, 2014
- Project Charter
  - Project Requirements
  - Project Plan Outline
  - Project Schedule
  - Final Project Plan
  - Progress Report 1: Updated Project Schedule
  - Progress Report 2: Iterative Design Phase
  - Progress Report 3: Design Project Prototyping Development
  - Progress Report 4: Testing and Evaluation
  - Progress Report 5: Updated Testing and
  - Final Report Draft
  - Final Report and Project Presentation

## Resource and Cost Management

- All software was open source
- All hardware was already owned
- Only cost was time

Activity	Labor (Hours)
Design	15
Develop Code	30
Test/Debug	145
Write Report	30
Generate Presentation	30
<b>Total Cost</b>	<b>250 hours</b>

2  
5

## Risk Management

- Accelerometer performance varies between phones
  - Mitigated with calibration settings
- Bluetooth requires root
  - Mitigated by moving to Wi-Fi
- Backups!

2  
6

## Project Procurement

- Free software
- Open source code
- Pre-own smartphone
- Pre-own computer

2  
7

## CHAPTER 6

### Conclusion

## Conclusions

- Sensors are noisy
- Integrating noise = drift
- Drift is hard.
- Works well enough to show it can be done, not well enough for everyday use
- Better sensors in the future will make this much easier

3  
0

Q/A?

