

CS3307 Group Project – Stage #1

Touch Pad Gesture

Group# 47

Member:

- Fan, Ziheng (zfan88)
- Jin, Yifang (yjin367)
- Liu, Zeyu (zliu973)
- Yu, Xiaoke (xyu429)
- Zhang, Xiaowei (xzha783)

Description

During years of development in the laptop industry, a touchpad has currently become a standard feature on all devices. However, for desktop user, they don't have the access to a similar touchpad hard wear. Hence, we are developing a Raspberry Pi based hard wear to fill this demand.

The basic function is through gathering finger landing data though a touchable screen, calculate the correct gesture, and output corresponding key-input or mouse-movement.

Features

Required Features

- Data transfer through local area network
 - All the data calculated through Raspberry Pi forms a string of instructions, which will be sent through local area network with protocols like TCP or UDP depending on the situation.
- Finger landing data and gesture understanding
 - The data gathering process should be though Raspberry Pi touchscreen, (exp. <https://www.amazon.ca/SunFounder-Raspberry-Touchscreen-1024%C3%97600-Capacitive/dp/B07Y889J3X/?th=1>)
 - Due to hard wear and budget limit, we will be mimicking this process with an Android phone and sending the data to the Raspberry Pi (no matter what platform we will use to simulate a Raspberry Pi hard wear, either through a PC or Raspberry Pi its self, we will always implement the function of transmitting data from a phone to Raspberry Pi)
 - For required features to lunch, gesture understanding will be fairly basic, only analyzing the basic movements such as scrolling up and down.

Optional Features

- Human-computer interaction optimization
 - Although this is not a required feature, it's still a core functionality of this program. An optimized algorithm will significantly increase user experience. Turning a basic functional to usable.
 - Some examples of the optimization would be finding the best travelling distance during a scroll gesture, or washing some trash data so that it would not miss trigger as much.
- Multi platform compatibility
 - To be able to deploy this program to different devices, it have to understand different system settings. Such as, MacOS have different finger gestures as Windows, and they have different sets of keyboard input to implement various functionality.
 - Different screen resolution is also another issue for different devices, so the mouse

movement should be a ratio of the screen rather than hardcoded pixel length.

- Client\Hard wear side UI

- The client side on PC could use a better user interface help our user with different functionality. Such as paring Raspberry Pi with QR code, customizing gestures and more. (Both wish list feature)

Wish-list Features

- Gyroscope

- This is suppose to be a optional feature during the original planning, later through research, with the understanding of it's complexity, we moved it to Wish-list, only finishing this if we have extra time.
 - The core concept of this feature is to use Raspberry Pi as a laser pointer to operate the cursor.
 - A further extend of this feature would be implementing a simple machine learning model to stableize the movement of the laser pointer, and other gesture reconization.

- Multiple port orginization

- With the above features finish, the program will run perfectly between one Raspberry Pi and one PC. However, once it encounters one Raspberry Pi to multiple PC scutation, current program could not handle multi processing.
 - We will need to implement a auto handling function to connect between different PC.

- Optimize messaging system

- The security of the message sent between Raspberry Pi and PC is not encrypted, it will be easily hijacked by anyone if they want
 - Using a third party encryption system will resolve this issue

- QR code

- Using QR code to simplify the process of connecting between Raspberry Pi and PC

Risks

- C++ does not have a large library like python, which supports almost every functionality that you want.
 - The only solution will be just write them yourself, with the downside of taking longer to develop the program
 - Multiple Port interrupting each other
 - As listed above in the Wish-list section, this would be solved through an auto handling function
 - TCP message not encrypted, could be hijacked
 - As listed above in the Wish-list section, this would be solved through using a third party encryption system
 - Multiple Platform have different API and Pixel ratio
 - As listed above in the Optional section, this would be solved through implanting different function for different platform
-

Other Notes

- The required hard wear for this project is:
 - An Android phone for finger and gyroscope data collection
 - A Linux environment on a device for processing data
 - A PC to receive the final result of the data calculation
-

References

- <https://github.com/>
- <https://stackoverflow.com/>
- <https://developer.android.com/docs/>
- <https://www.raspberrypi.com/documentation/>