Keyboard Alternative Technology

"Sally" Ji Hyun Kong and Stephen Lane
SIG Center for Computer Graphics

ABSTRACT

The goal of this project is to develop an alternative keyboard input system that will allow users to be able to type in their mobile devices without the interference with the front display that touchscreen keyboards produce. To achieve this goal, we designed a keyboard app that could be operated using four fingers from the back of a mobile device. To simulate having input from the back of the mobile device and displaying the output in the front screen, we used the Unity Engine to develop and export an application for a mobile device and a desktop computer, and implemented a peer-to-peer network connection between the two.

APPROACH

1. Gesture Recognition and Finger Object Implementation

For this project, three finger gestures are recognized: Press, Pull and Push. When an initial contact is made, a finger object is created. When the finger is in contact with the screen, the finger object keeps track of the finger position and determines what gestures were made based on a specific time period and distance traveled. There are seven combinations that are recognized: [Push, Release], [Push, Press, Release], [Press, Push, Release], [Press, Release], [Press, Pull, Release], [Pull, Press, Release], [Pull, Release].

2. Quadrant Calibration

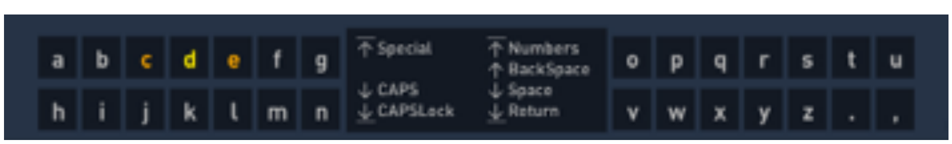
This interface is designed to be used with four fingers: index and middle fingers of both hands. Hence, a quadrant was calibrated based on the contact position of the four fingers which later labeled these fingers with L1, R1, L2, or R2.

3. Reservation Queue Implementation

When a finger object is created and labeled as a valid finger by the quadrant, it is inserted to the reservation queue. A reservation queue only outputs the key of a finger object and destroys the finger object when it is in front of the queue and completed one of the seven recognized gesture combinations. This keeps track of the order the fingers were used, and thus allows simultaneous typing with different fingers.

4. Keyboard Visualization

A keyboard display that highlights current intended character and suggests next possible moves was implemented to help in ease of use.



5. Network Socket Implementation for Mobile and Desktop App

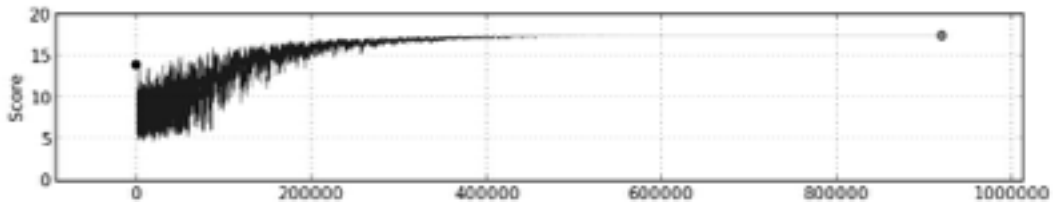
To simulate a mobile device that uses haptic input from its back and shows output in its main display, the keyboard application was exported as a mobile app and a desktop app. A port in a shared WiFi connection was used to send the necessary data from the mobile device to the desktop app.

Keyboard Alternative Technology

"Sally" Ji Hyun Kong and Stephen Lane
SIG Center for Computer Graphics

6. Optimal Keyboard Layout Research

The keyboard layout was represented as a 2 x 14 array of characters that had a specific score based on the hand, finger, and gesture combination. Using the New York Times Corpus of individual letter and bigram (i.e two letter combination) frequencies, we computed an overall score based on each finger's gesture and letter layout. Then to find the optimal keyboard layout, we used an optimization approach based on the Simulated Annealing Algorithm.



7. Typing Tutorial Game

To help users familiarize themselves with this new keyboard interface and also to keep track of their learning curve, we created a simple typing tutorial game.

FUTURE WORK

- User Testing
- Modifying the network socket so that the desktop app can also send data to the mobile app
- Integrating app with Google Glass