Declan Urbaniak-Dornstauder
DVU072 / 11284744
CMPT 481
Project Plan

Repository: git.cs.usask.ca/dvu072/cmpt-481-project

Interaction problem

This project addresses the challenge of optimizing text input and command execution on computers, particularly for users who encounter difficulties with traditional keyboard-based methods. Whether due to physical constraints, lack of technical proficiency, or individual preferences, typing may be impractical for certain user groups. The objective of this project is to develop and explore a user-friendly alternative to text input and text-related functionality that accommodates a diverse range of users.

Interaction technique

The proposed interaction technique involves using a external touchscreen device as an alternative input method for computers. Users can write letters and symbols directly on the screen instead of typing on a physical keyboard. The technique offers two main modes: ‘insert mode’ for text input and ‘command mode’ for executing various text related commands and shortcuts. In insert mode, users can write and delete characters at the position of the text cursor on their computer. In command mode, users can perform various touch gestures to control the cursor, select text, and perform actions like copying, cutting, and pasting. This approach provides a more natural and user-friendly way to interact with computers, catering to diverse user needs and preferences.

Full functionality list:

Touch input	Functionality
Double tap	Toggle between insert and command mode
Insert mode	
Draw alphanumeric character or symbol	Insert equivalent ASCII character/symbol at cursor
Two finger swipe left	Delete character/symbol at cursor
Command mode	
Modifier* + one finger swipe left/right	Move cursor one character left/right
Modifier* + two finger swipe left/right	Move cursor one word left/right
Modifier* + draw ‘S’ + one finger swipe left/right	Select character to left/right at cursor
Draw ‘S’ + two finger swipe left/right:	Select word to left/right at cursor
Draw ‘C’	Copy selection
Draw ‘X’	Cut/delete selection
Modifier* + Draw ‘V’	Paste selection at cursor

*Repetition modifier: Draw a number 1 – 9 and repeat the command this many times. If the repetition modifier is not specified, the command will execute once.

Tools and resources

The implementation of this interaction technique should simulate an external input device used peripherally with another device, similar to how a keyboard is used with a desktop computer. This can be accomplished using a mobile device with touchscreen capabilities, such as a phone or tablet. The device will run software to enable the recording of touch gestures and handwritten characters on-screen. Touch gestures are mapped to text commands, and hand-written characters are translated to their ASCII equivalents, then both of these outputs are transmitted externally to a computer (or capable device) via Bluetooth.

For the sake of compatibility and development familiarity, the software enabling the interaction technique will be developed for Android devices using Kotlin. The structure of the application can be broken down into the following layers:

1. *Presentation layer:*
    - Handles the visual UI components and interaction with the user.
    - Developed primarily using tools from Jetpack Compose toolkit
2. *Input processing layer*
    - Processes touch input from the user and translates it into meaningful data for further processing.
    - Uses Google ML Kit or Tesseract OCR to interpret handwritten characters
    - Uses Android GestureDetector module to interpret touch gestures
3. *General logic layer*
    - Core logic of the application used to coordinate different modules and manage application state
4. *Communication layer*
    - Enables communication between the Android device and the computer for transmitting interpreted characters and gestures.
    - Uses Android Bluetooth module

Outline of activities

1. *Research:* Further research is required to ensure appropriate technologies, libraries, etc. are used.

2. *Minimal presentation layer:* Develop the UI to the point of enabling writing on-screen and recording of written input.

3. *Minimal input processing layer:* Implement the functionality to process touch inputs. Recordings of written characters should be processed correctly by the OCR, and touch gestures should be correctly interpreted.

4. *Refine presentation and processing layers*: Refine presentation and processing layers with a focus on usability and fluidity – i.e. ensure there is little to no delay in inputting and processing touch input. Implement insert/command mode system.

5. *Complete communication layer:* Fully complete the communication layer to enable connection and data transmission to computers.

6. Further refine presentation layer: Refine presentation layer to enhance clarity and implement additional UI features as necessary – i.e. how does user know what mode they are in? How do they know what gesture is required for a certain command?

7. *Evaluation and report*: Evaluate the interaction technique and report evaluation findings.

Milestone 1 (March 8): Complete activities 1 – 3.

Milestone 2 (March 22): Complete activity 4, then 5. If enough time is available, complete activity 6 prior to activity 7.

Risks

1. *Touch gestures and character recognition:* Because the Android application supporting this interaction technique makes use of both touch gestures and handwritten character recognition to determine what a user is trying to input, I could run into issues interpreting these types of inputs together. For example, a gesture may be interpreted as a character, and vice-versa. To avoid any inputs being misinterpreted, I will need to carefully design the input gestures and limit how characters can interpreted so that is is little to no cross-over.

2. *Bluetooth module:* I have never worked with this module while developing Android applications, so its possible this could pose some issues due to its importance in enabling the interaction technique. That being said, I have worked with many unfamiliar Android modules in the past, and with enough research and troubleshooting, I am generally able to surmount any obstacles slowing down development.

Evaluation

To evaluate the proposed interaction technique, it will be compared to two use cases of a standard keyboard setup. The first use case seeks to replicate expert keyboard usage through key-shortcuts to execute text commands. For example, moving the cursor to the next word can be accomplished by a two-finger swipe using the interaction technique, which is the same as pressing control with the right arrow key in expert usage. The second version replicates novice keyboard usage. The equivalent function in this scenario is using the mouse to reposition the cursor on the next word. The equivalent expert and novice methods for commands that can be executed by the interaction technique are as follows:

Command	Expert method	Novice method
Move cursor one character left/right	Left/right arrow key	Move to character position with mouse
Move cursor one word left/right	Ctrl + left/right arrow key	Move to word position with mouse

Select character to left/right at cursor	Shift + left/right arrow key	Select character with mouse
select word to left/right at cursor	Ctrl + shift + left/right arrow key	Select word with mouse
Copy selection	Ctrl + C key	Right click + select ‘copy’ from menu
Cut/delete selection	Ctrl + X key	Right click + select ‘cut’ from menu
Paste selection at cursor	Ctrl + V key	Right click + select ‘paste’ from menu

The interaction technique and keyboard techniques will be compared and evaluated through the completion of a series of self-conducted trials. These trials will focus on assessing effectiveness, efficiency, and usability by recording objective metrics such as completion time and accuracy (error-rate).

Each trial will consist of a sequence of standardized tasks representing common text input and command execution scenarios, including the following:

- Inputting and replacing words and characters.
- Navigating through text using cursor movements and selecting text.
- Performing editing functions such as copying, cutting, and pasting text.

Trial task sequences are predetermined and will performed using the same sample text. Each task sequence (trial content) will be tested with all three interaction techniques.

The time to complete each task will be recorded. Until I come up with a better way to do so, this will done manually for each trial using a stopwatch or timer. Errors or inaccuracies in text input and command execution will also be manually recorded.

After all trials have been completed, the average task completion times will be taken for each group of tasks based on the command they use. The averages from the interaction technique will be compared to both use scenarios of the keyboard technique.