Assignment 4: Keyboard Analysis
REPORT

Dhruv Prasad, EE23B130

1st September 2024

1 Introduction

Python program that analyzes keyboard usage patterns for a given text input. The program:

- Generates a heatmap visualization of key usage
- Calculates the total distance traveled by fingers while typing.

2 Usage

- Extract **ee23b130.py** and **kbd_layout.py** from my .zip submission and save them to the same directory
- Run the script with **python ee23b130.py**
- A file **keyboard_heatmap.png**

Note: To test my code with a different layout, either modify my layout file or replace it with another in the same format with the same file name.

Note: To test my code with a different text input, modify the sample_text string in my script.

2.1 Assumptions

- The positions of the keys must have integral y-coordinates for the keys to not overlap, as I have assumed key height to be 1. On the other hand x-coordinates can be anything.
  Note: The image will still be generated
- The string must not have any newlines in it

3 Methodology

3.1 Key Press Frequency and Finger Travel

- For each character in the string, the program tracks the corresponding key presses on the keyboard and calculates the Euclidean distance between that key and it's corresponding key on the home row. The sum of these distances represents the finger travel required to type that key and is printed out.
- Additionally, a frequency count of each key press is maintained in a dictionary.

3.2 Layout display

- First I get row wise data of each key in an array and sort it by x-coordinate assuming key heights to be 1 and widths to be some constants.
- Special attention is given to adjusting the size of certain keys, such as Shift and Space, which occupy more space than regular keys.
- If any key does not have enough place to be drawn to their default size, I draw it upto the next key in the row. This makes my keyboard robust and able to plot weird layouts properly, as shown in the example outputs.

3.3 Heatmap Generation

- First I construct a 2D grid over the keyboard. Each key centre is given a value equal to it's frequency, and the neighbouring values in a circle of fixed radius around it are decided by a Gaussian function.
- Additionally, a frequency count of each key press is maintained in a dictionary.

Note: I set the frequency of 'Space' to 0 because it usually has much higher frequency than other keys and skews a linear heatmap to the point of other keys being too light

3.4 Plotting and Visualization

- I create a custom colormap from the matplotlib default 'Spectral' colormap by inverting and scaling it.
- I use imshow() function to display the heatmap grid. Bicubic interpolation option is applied to further smoothen the heatmap.
- The keys are drawn as rounded rectangles with text in the centre
- A colorbar with scale is drawn at the side to indicate the absolute frequency of key presses.

4 Outputs and images

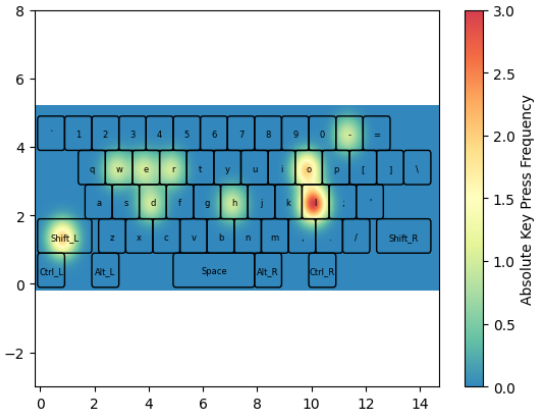

Figure 1: QWERTY with 'Hello_world'
Finger travel = 12.200575343245989

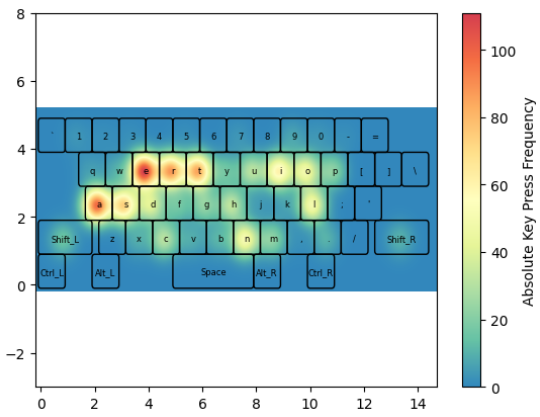

Figure 2: QWERTY with default sample_text
Finger travel = 711.7114244494628

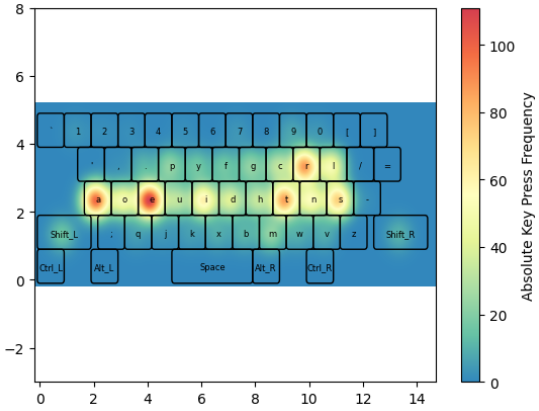

Figure 3: DVORAK with default sample_text
Finger travel = 1209.6110760776944

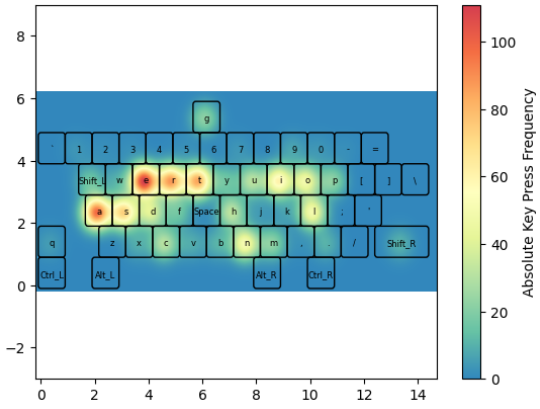

Figure 4: Custom layout with default sample_text
Finger travel = 621.1846533323435

5 Conclusion

- DVORAK was less efficient than QWERTY for my default sample_text.
- By moving the space and Shift_L keys closer to the center I was able to get better finger travel.