Assignment 4 Keyboard Heatmap

Deepak Charan S ee23b022

September 2024

1 Introduction

I am Deepak Charan S and this is my report for the fourth assignment of EE2703 course (Applied Programming Lab).

Disclaimer: Before the change in layout, I had made two files; *main_kybd.ipynb* and *kbd_animate.py* (Since animation doesnt run on jupyter, I had to make an alternate script for animation)
Now, I have made a notebook, *kybd_alt.ipynb* to test against the new layout but the rest of the report assumes *main_kybd.ipynb* to be the main submission.

2 Assumptions:

1. the given layout always has 4 rows, with the 3rd row being the home row (and each rows has atleast 9 keys in it)
2. All keys to the right side of the keyboard are pressed by the little finger
3. For paragraph-style strings which are quite long, Space key heavily dominates and **skews the distrbution** of my heatmap so I have sidelined it (kept it as grey)

3 Overall Approach:

I used a class 'kbd_analysis' to perform the desired operations (takes in ***QWERTY_LAYOUT*** as default layout)

3.1 Methods Used:

- **get_key_position** : takes in a key as an input and returns the coordinates of it (using the layout).
- **euc_dist**: takes in two keys as inputs and returns the Euclidean Distance between them.

- ❊ **key_dist**: Creates a map of each key and their corresponding travel distances (using *euc_dist* and stores it in *'self.travel'* dictionary)
- ❊ **visualise_kyb**: plots the keyboard
- ❊ **travel_dist**: takes in a string as input, goes through the string and calculates the frequency of each key pressed. It returns the total travel distance and calls 'visualise_kyb' as well

Upon creating a class object and calling the *travel_dist* method for a string using a layout; my class first maps each key to its travel distance. Then it runs through the string and computes the overall travelling distance and keeps a count of how often each key is pressed. Finally, it plots the heatmap corresponding the usage of each key

3.2 How To Run

1. ***main_kybd.py*:** Run the prerequisite cells (import libraries, QWERTY_LAYOUT, and the class
2. The user can provide their own layout (or use 'qwerty' by default) to create a class object
3. User can then run this object across multiple strings using the *travel_dist* method (I have given some example strings to see and also gave an option to check for any arbitrary string)
4. I have also implemented **DVORAK** and **COLEMAK** layout and ran all these 3 keyboard layout against a very long string
5. Finally, I have kept a cell to check for any input layout and string provided by the user and compute the travel distance
6. ***kbd_animate.py*:** Run this code on VSCode preferably (animations dont show up in Jupyter). I have attached some sample strings to check for and also kept an option to check for any input strings
7. ***kybd_alt.py*:** Run this code similar to how we ran *main_kybd.ipynb*. I have attached a *qwerty_layout.py* script which this notebook imports from and computes travel distance with (User is free to attach their own keyboard layout and check)

4 Sample Outputs:

Attached these in my main notebook (Colormap used is attached in my main notebook). I have kept the travel distance for the correspondign heatmaps above the plot

I Link to a sample output of my animation is given here

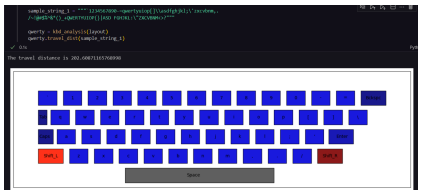

Figure 1: Every key being pressed

Figure 2: some arbitrary string

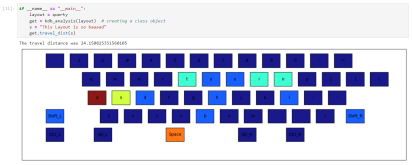

Figure 3: New Layout

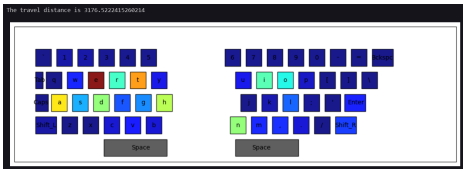

Figure 4: Split Keyboard

5 Findings:

upon checking it with a very long string (*attached in my main notebook*, Qwerty keyboard seemed to be the least efficient (high travel distance (TD)) with Dvorak and Colemak being much better and similar

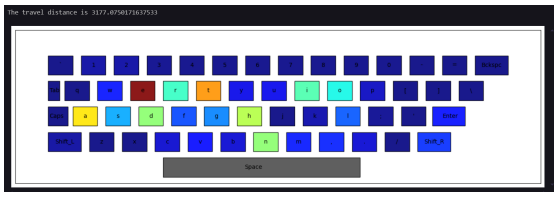

Figure 5: Qwerty (3177 TD)

Figure 6: Dvorak (1847 TD)

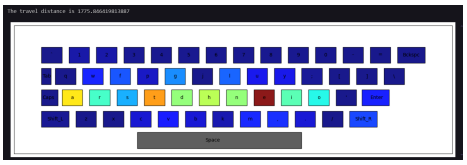

Figure 7: Colemak (1775 TD)

We can see how most of the heat is concentrated on the homerow for Dvorak

and Colemak (And subsequently having much better travel distances)

6 References

- How color maps work and the different kinds of colormap
- To check how to use a colormap and accrodingly use it as a heatmap for the keyboard
- How to add rectangles on a matplot figure
- How to add text on a matplot figure