

Assignment-4 : Keyboard Layout

Deenabandhan N
EE23B021

Assumptions

- I have assumed that the distance is just unidirectional i.e the distance is calculated from home row to the respective letter and the cost of coming back is not calculated.
- I have assumed that the input string is given as a string variable in the python ipynb file.
- I have assumed that the layout that is used is imported from a python module and given as an input to the display function.

How to run the ipynb file

- To run the submitted file , you need to import the layout and string to be used.
- If you need to display the animated heatmap generated you have to use the following code snippet.

```
display_anim(string,layout)
```

where string is the input string and layout is the layout to be used in the code.

- If you need to display just the picture of the heatmap generated you have to use the following code snippet.

```
display(string,layout)
```

where string is the input string and layout is the layout to be used in the code.

Approach

- The approach to this assignment involves three steps ,
 - Generating the keyboard
 - Generating the frequency array
 - Generating the heatmap

Generating the keyboard :

- I have used **patches** module in matplotlib to generate rectangular patches and used different dimensions for normal and special letters to distinguish between them.
- The same module is used to store all the coordinates to compute cost/heatmap in the later part of the code.

Generating the frequency array

- The frequency array generation is done by using iterating through each character of the array and adding the respective frequency values.

Generating the heatmap

- The interesting part of the assignment was this part where given a frequency array , we need to generate the heatmap.
- I have used the module **imshow()** in the matplotlib library to perform this operation.
- To use imshow() , we would need a 2-D array which has a unique mapping to the coordinated of the plot.
- I have made the size of such array to be 1000*10000.
- To have a blending heatmap , I have used a gaussian function to determine the color of the region.
- The code snippet is as follows ,

```
heatmap += dat[i]*np.exp(-(distance**2) / (2 * sigma**2))
```

where heatmap is the two-dimensional array and dat is the normalised frequency array and I am computing the gaussian value with variable as distance between the position of the letter and the current point.

- One can change the value of σ , to increase/decrease the blending region.

Insights :

- I have observed that in large texts , space always gets the maximum frequency always.
- Also the median color happens very less frequently.
- The reason I think is when normalised the spcetrum of frequency array , some letters appear most and some letters appear few times , there is not much of letters that appears like an average number of times.

Trying different layouts

- I have used the standard text , **The quick brown fox jumps over the lazy dog** to check which keyboard is optimal.

QWERTY :

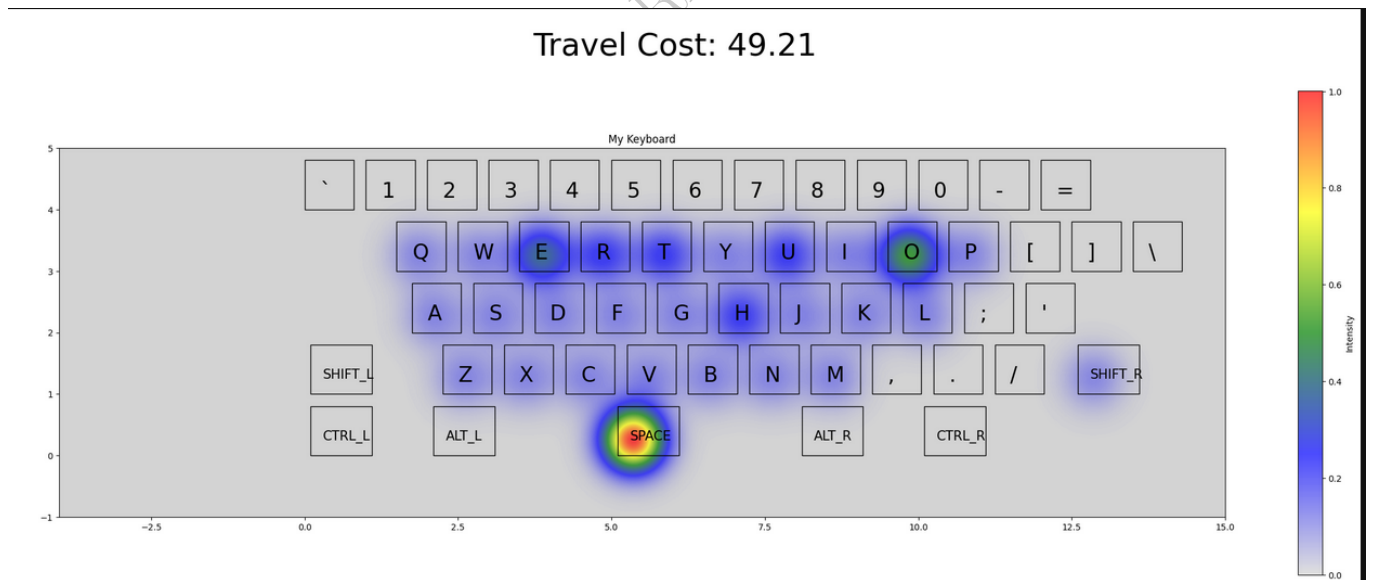


Figure 1: Qwerty_Layout

COLEMAK :

Travel Cost: 41.95

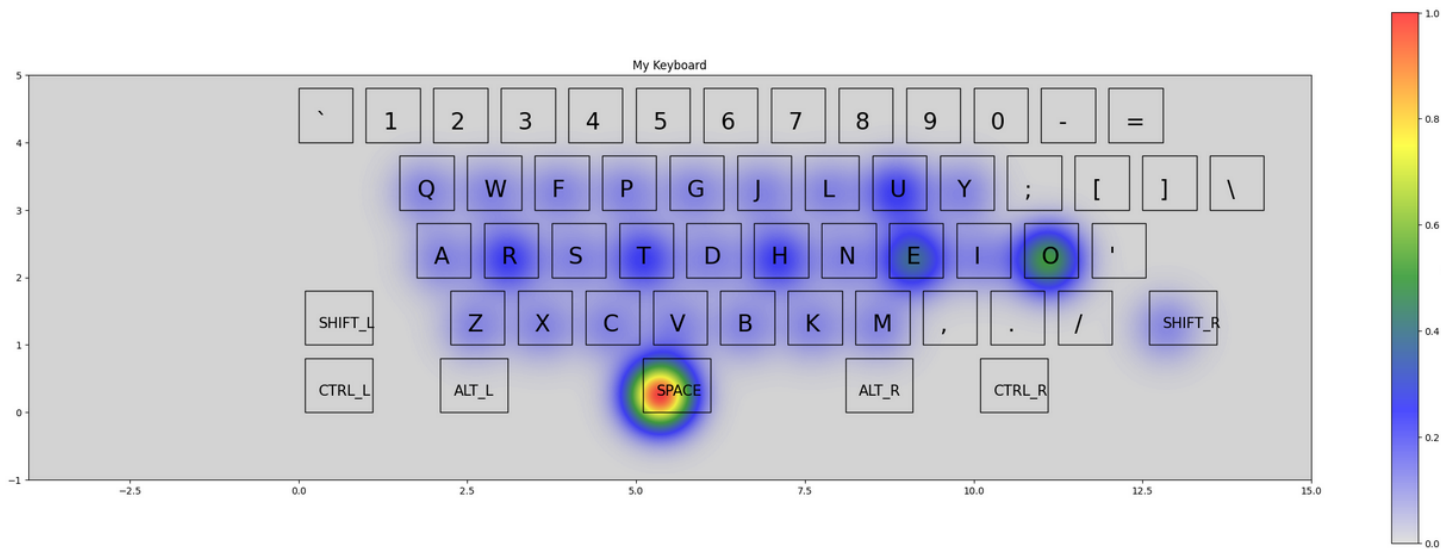


Figure 2: Colemak_layout

DVORAK :

39.92256697918758

Travel Cost: 39.92

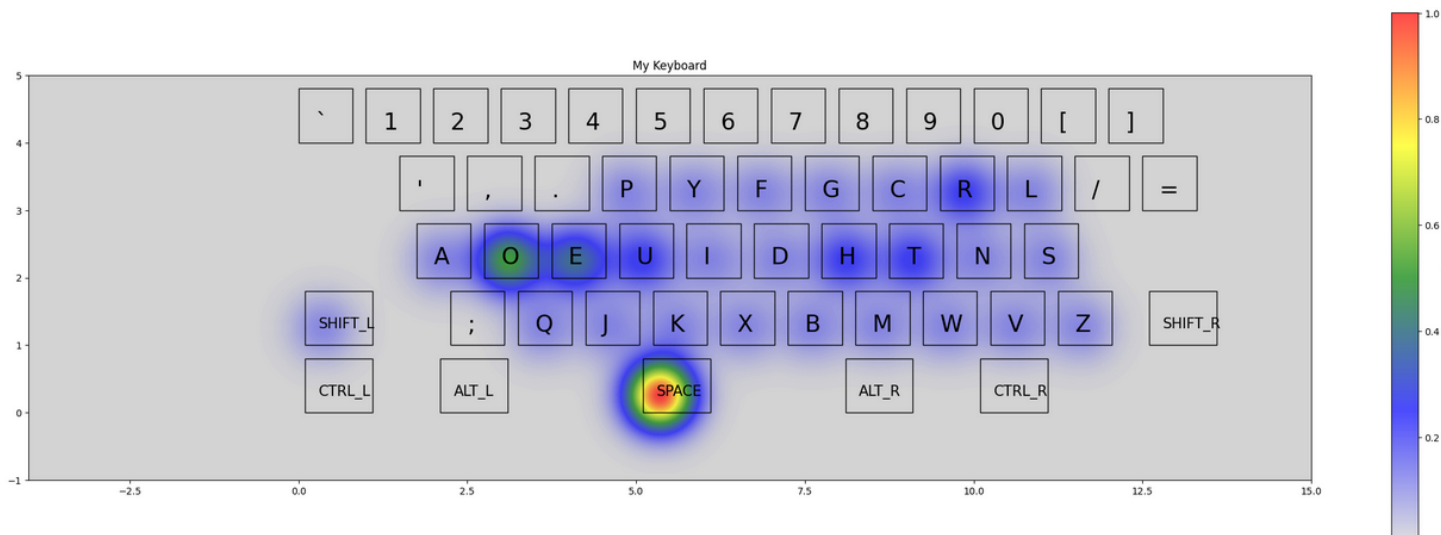


Figure 3: Dvorak_layout

- I have changed the keyboard according to my own intuition.
- If someone change the hom row of the file the output will change.

References :

- I have referenced the matplotlib documentations.