Colemak-Im: An Improved Keyboard Layout for the New Age

Ahmed Thahir[a] and Raja Muthalagu[b]

[a, b]*Computer Science, BITS Pilani Dubai Campus, Dubai, UAE*

Colemak-Im: An Improved Keyboard Layout for the New Age

The shift towards digital transformation across the world over the past few decades has made digital ergonomics more important. Current keyboard layouts are either unergonomic or expensive, and hence, there has been a lot of research in this field. Earlier studies seek to improve keyboard layouts based on typing ergonomics and the frequency of letter co-occurrence; however, they do not focus as much on the effort required for navigation, typing numbers, symbols, modifier keys, etc. This report proposes a cost-effective and ergonomic keyboard layout 'Colemak-Im' to address issues in current layouts. However, this keyboard layout is still premature for wide-scale use.

Keywords: Keyboard Layout, Typing, Ergonomics, Colemak, Qwerty

Introduction

Digital ergonomics is the applied science concerned with the efficient and safe usage of digital tools [1]. According to studies, the amount of pressure applied to each key, the travel distance, and the tactile feedback affect typing discomfort and fatigue [2]. Use of better ergonomics at computer workstations has the potential to increase typists' productivity levels while decreasing effort and pain.The following covers the basic concepts relevant to keyboards and typing ergonomics.

Key Rollover

The ability to respond to newly-pressed keys regardless of how many keys are currently held down, i.e. the keyboard's ability to handle simultaneous keystrokes correctly. The rollover of a keyboard can be tested using Microsoft Anti-Ghosting Demo [3].

n-key rollover

Each key is scanned by the keyboard independent of other keys. This means that every keypress is detected correctly regardless of the number of currently-pressed keys [4].

k-key rollover

Only k keys can be scanned by the keyboard independent of other keys. This is done to reduce the cost of the keyboard. The most common rollovers are 2-key and 6-key. With these keyboards, pressing as few as three keys can cause ghosting effects, although care is taken when laying out the matrix arrangement that this does not happen for common modifier key combinations [4].

Keyboard Arrangements (Hardware)

These are the hardware configurations/layouts of the actual keyboard to improve ergonomics.

Split Keyboard

In order to increase the width between both hands, the keyboard is split.

Figure 1. Freestyle2 Ergonomic Keyboard
Source: [5]

Angled Keyboard

In order to reduce ulnar deviation, which is when the wrist is bent outward in the direction of the little finger, and causes wrist damage.

Figure 2. Angled Keyboard
Source: [6]

Kinesis Keyboards

These are considered the golden standard of ergonomic keyboards. However, they are either ~$300, but are single platform (Windows Only, MacOS Only); or else, it costs ~$600, for cross-platform experience. [7]

Figure 3. Kinesis Advantage 2
Source: [7]

Keyboard Layouts (Software)

These are more flexible as it just involves remapping of keys at the software level to improve efficiency.

Qwerty

This is currently the most widely-used keyboard layout. However, it was actually made just to hinder the typing speed of typists to prevent typewriter jams.

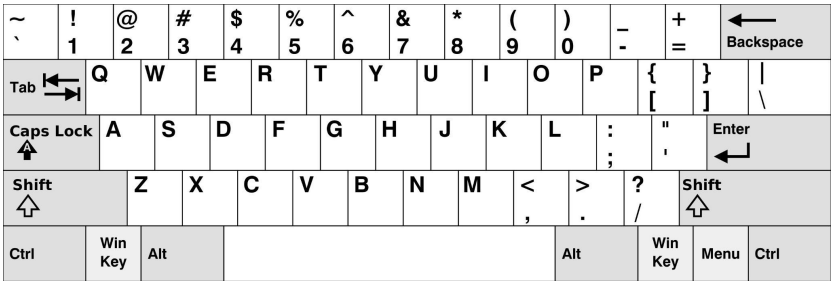

Figure 4. Qwerty Keyboard Layout
Source: [8]

Dvorak, Workman

These layouts were meant to be better than Qwerty. Though they perform well in standard layout testing tools, they have a steep learning curve as there are a lot of variations from Qwerty, making them hard for new adopters. Hence, they are not widely-used by common typists.

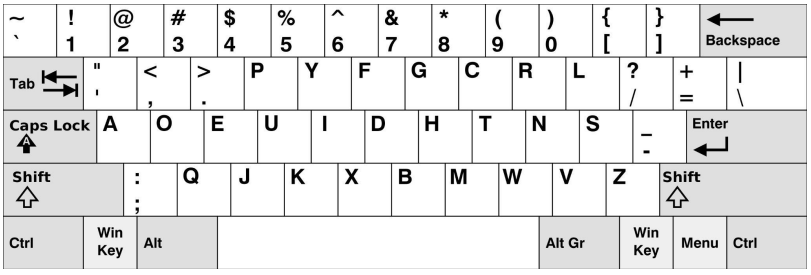

Figure 5. Dvorak Layout
Source: [9]

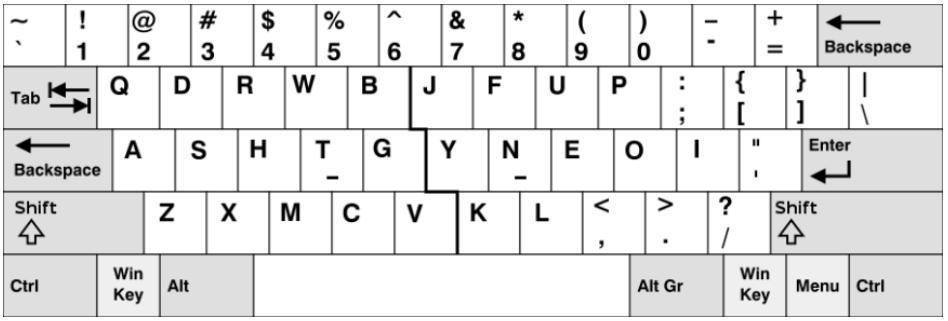

Figure 6. Workman Layout
Source: [10]

Colemak

The QWERTY layout was designed in the 19th century. Colemak is a modern alternative to the QWERTY and Dvorak layouts, designed for efficient and ergonomic touch typing in English. It was created by Shai Coleman in 2006.

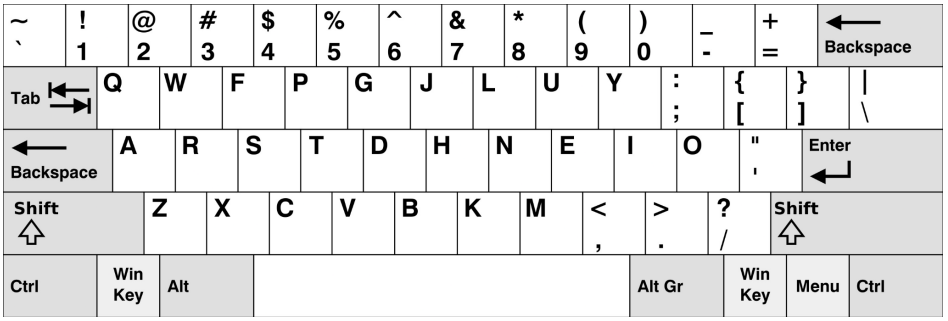

Figure 7. Colemak Layout
Source: [11]

It has had various mods (modified versions) to optimise the layout even further.

- Angle Mod [12]
- Wide Mod [13]
- Extend Layer [14] using a hotkey like to access other keys
- Symbols Layer [15] using a hotkey like to input symbols
- Optimised Modifier Keys [16]
- Curl Mods [17] arranges keys to reduce lateral movement of index fingers for minimising typing effort

These mods have been used as inspiration points for Colemak-Im.

Colemak DHk

Colemak Mod-DHk Wide Angle Mod is an unofficial implementation of Colemak to implement the above mods. Most users prefer standard staggered keyboards as they are cheaper in the market, so this report focuses on the Colemak Mod-DHk rather than Colemak Mod-DHm.

Figure 8. Colemak Mod-Dhm Layout
Source: [18]

VIM's Ideology

VI-Improved is a text editor that is very famous for its approach to using a computer, without the use of a mouse. It has various modes such as insert mode (which lets you enter text) and normal mode (which lets you navigate the interface).

Keyboard Standards

"Standards are technical documents intended to establish common solutions to repetitive requirements." [19] There are 2 different standards for keyboards: ANSI and ISO [20].

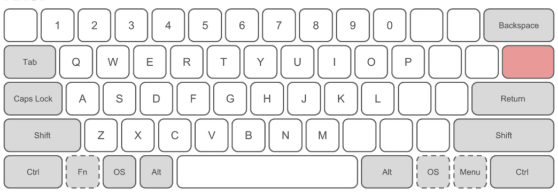

Figure 9. ANSI Standard
Source: [20]

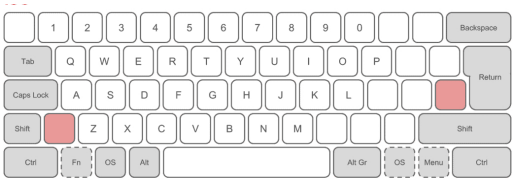

Figure 10. ISO Standard
Source: [20]

Literature Review

This section highlights the recent research in the field of keyboard layouts.

[21] performs a text analysis of words. The analyzer provides the most frequent letters and most common pairs of letters used in a text. The paper aims to implement a layout for the keyboard arrangement problem (KAP) that is based on the frequency of letters and to create the optimal model for a physical keyboard. The design of the ergonomic keyboard is then adjusted using a genetic algorithm to reduce the overall distance travelled by the fingers to access the chosen alphanumeric characters. The results revealed that the proposed keyboard layout's distance travelled is less than the QWERTY keyboard in all sorts of texts, with an average improvement of 6.04%.

[22] uses a deep neural network in conjunction with a genetic search algorithm. The paper proposes a new keyboard layout called MKLOGA. It produced a better keyboard layout than previous algorithms, using the carpalx model for analysing typing effort. It also showed experiments with 300 participants to analyse how typists find new keyboard layouts.

[23] introduces a model to quantitatively analyse typing effort and applies it to (a) evaluate QWERTY and popular alternatives, such as Dvorak and Colemak and (b) find the keyboard layouts that minimise typing effort for a given set of input documents.

[24] has developed a system incorporating factors affecting efficient and comfortable keyboard use. It simulates typing a large volume of standard English text, and rates each layout according to its two most important considerations for keyboard layout design: 1) A base effort value based on the ease with which each key can be typed on average, incorporating finger strength, lateral motion, key size and Fitts's Law; 2) additional penalty accumulated when typing successive keys. My research will be using this model.

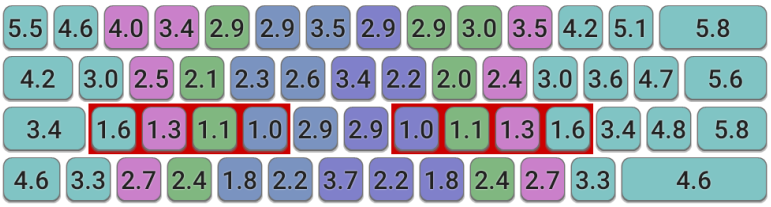

Figure 11. Keyboard effort grid, colour-coded by finger that should be used
Source: [24]

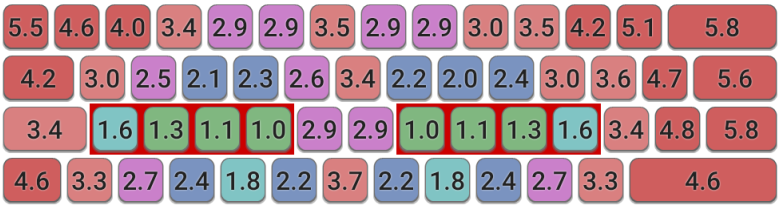

Figure 12. Keyboard effort grid, colour-coded by effort
Source: [24]

Literature Gaps

The current ‘ergonomic’ keyboard arrangements help improve ergonomics but are way too expensive, varying from $300-$1200 on Amazon. Moreover, these keyboard layouts are only

effective, if you actually use them. This would mean that you would have to take it around with you to reap the benefits. Furthermore, you will not be able to use others' devices in an ergonomic manner.

While the current software based layouts help address the above issues as they are portable, there is still room for improvement to reduce unnecessary keypresses, such as reduce pinky effort, better utilisation of the thumbs, etc.

Proposal: Colemak-Im

This report proposes a software keyboard layout that more effectively arranges keys, in order to minimise pain and maximise efficiency through

- minimising finger travel
- distribute typing effort to more fingers
- reducing diagonal movement of pinky fingers

This **may not be** the fastest keyboard layout, as ease of transition is also considered. This is the modelled diagram for Colemak-Im. Yellow buttons highlight the modifier buttons currently pressed.

Numbering System

Normal Layout

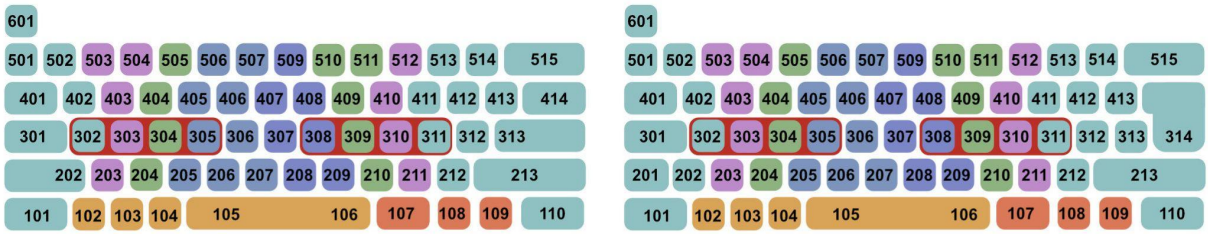

Figure 13. Numbering System for Normal Layouts

Wide Layout

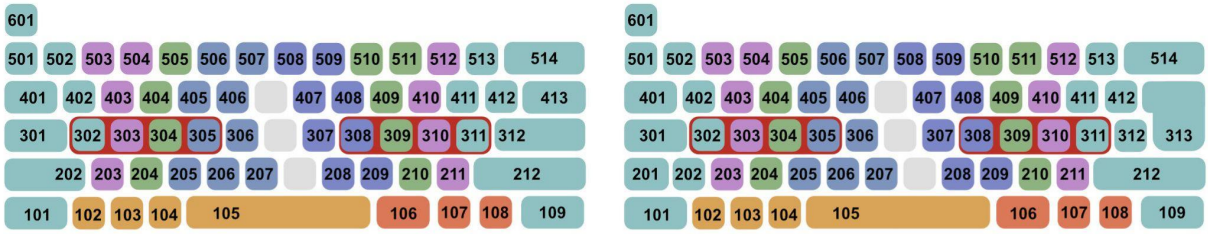

Figure 14. Numbering System for Wide Layouts

Key Mappings

Each coloured key shows a different finger. Yellow-coloured keys show the currently pressed hotkey. The red background shows the position of the home row.

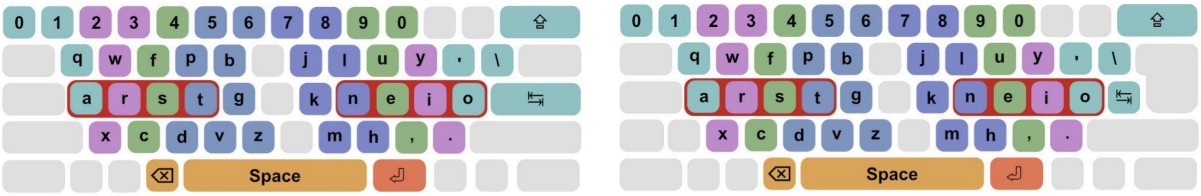

Figure 15. Colemak-Im Single Press Layer (Base Layer)

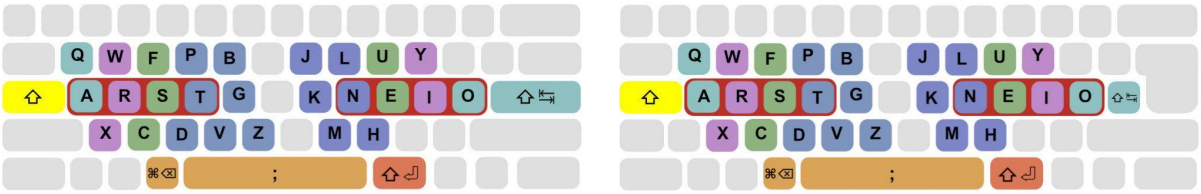

Figure 16. Colemak-Im Shift Layer

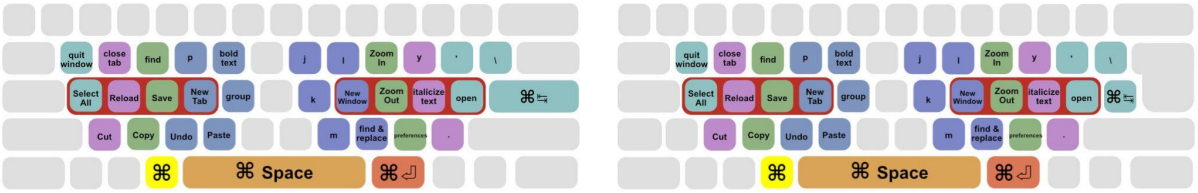

Figure 17. Colemak-Im Left Meta Layer

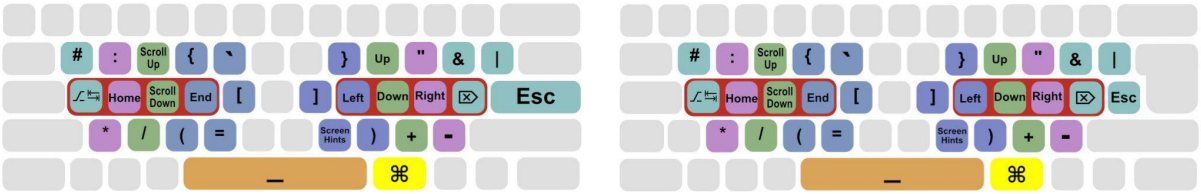

Figure 18. Colemak-Im Right Meta Layer

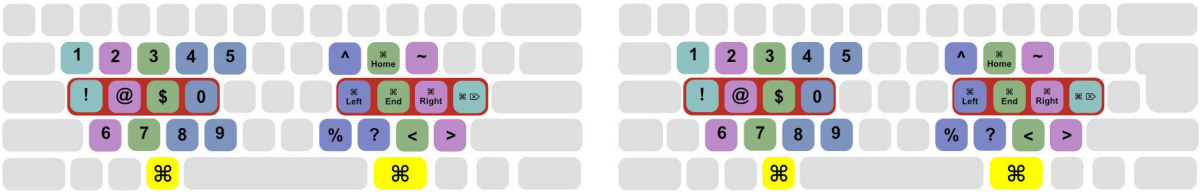

Figure 19. Colemak-Im Right Meta, Left Meta Layer

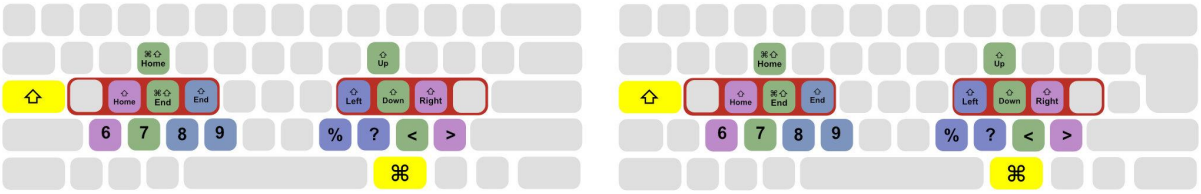

Figure 20. Colemak-Im Right Meta, Shift Layer

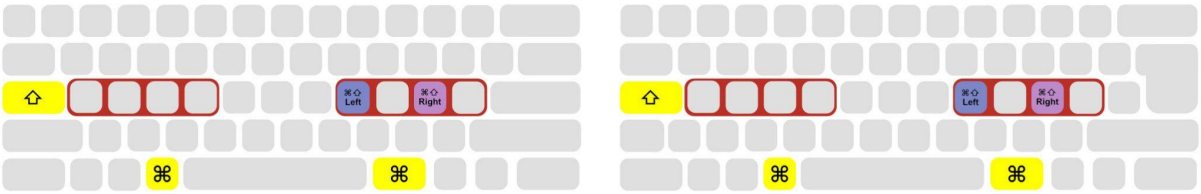

Figure 21. Colemak-Im Right Meta, Left Meta, Shift Layer

Input	**Output**	**Explanation**
Single-Click LAlt/LCmd Single-Click RAlt/RCmd	Backspace Enter	Backspace and Enter are the two of the most-used keys, for all purposes - regular typing, photo-editing, etc. to delete mistakes, create new lines, submit your input, and many more. Hence, it only makes sense to make them easily-accessible, while ensuring that comfort of typing words and sentences are not disrupted. The thumbs are usually under-used apart from clicking the space bar, so the left and right Alt/Cmd keys are ideal for this.
Enter	Tab	Especially in programming, tabs are a simple way to perform indentation. Already it is an improvement over typing Spacebar 4 times. However, it involves using the left-pinky finger in an awkward diagonally-upward direction. Shifting the load to the right pinky (which is now free as the enter and backspace have been moved to the thumbs) greatly reduces overall effort. This also helps reduce the effort when switching between Tabs in applications such as browsers, using Ctrl-Tab and Ctrl-Shift-Tab.
Caps Lock	Shift	Shift is a key that is used regularly in regular passage typing as well as a modifier key for special characters. Even this involves moving the left-pinky to an awkward diagonally-downward direction. Shift-ing it upward by one position greatly reduces the effort required.
LAlt/LCmd-D	Ctrl+Z	Undo/redo using Ctrl-Z and Ctrl-Shift-Z are the most frequent operations that most people use, especially when photo/video editing. I did not add a separate shortcut for redo, as it would affect common shortcuts in the neighbouring area. Ctrl-D is not very commonly used in most programs. However, through an unbiased perspective, this should be considered as a limitation.
RAlt/RCmd+,	(Forward) Delete	In conjunction with the movement keys, many times typists use delete. This mapping eliminates the need to - move right hand all the way to the delete key - use extra arrow keypresses and use backspace

RAlt/RCmd + Arrow Keys	Up, Down, Left, Right Scroll up, Scroll down Home, End	Using RAlt/RCmd as a modifier key to access these keys helps eliminate the use of the arrow keys and far-off keys such as Home, End, which prevents the need for the typist to move their hands away from the home row, thus eliminating overall effort. This also prevents the need to use a mouse/trackpad.
Alt-a	Alt-Tab	This is easily one of the most-used shortcuts, especially by those working while referring to documentation. This mapping eliminates the need of the typist from shifting their hands away from the home row.
Capslock + Space	;	To access ; we need an extra keypress of Shift, which is a drawback, but it is a reasonable trade-off, as many modern programming languages (such as JavaScript, Python, R, Julia) don't use even use ;
Shift-RAlt/RCmd	Shift-Enter	Greatly helps in Integrated Development Environments such as Jupyter Notebook and Google Colab, where the default execution shortcut is Shift-Enter. Moreover, Shift-Enter is the default shortcut to create a single line break in many document formatting softwares, such as Word, Google Docs.
RAlt/RCmd+LAlt/LCmd+Keys	Numbers Symbols	Not very frequently used, but this combination of keys avoids the need to move palms away from the default position.
RAlt/RCmd+Shirt+ Bottom Row Keys (Duplicate Layer)	6789%?<>	This layer serves as a fallback for keyboards with only 2-key rollover

Table 1. Mappings Explanation

Benefits

1. Cross platform (one can switch from Windows and MacOS with ease)
2. Acts an angled keyboard
3. Pinky fingers have minimal usage

Layout Implementation

Operating System	Format	Tool

Windows	.exe	AutoHotkey[25]
MacOS	.json	Karabiner-Elements[26]
Linux (Not yet implemented)	N/A	N/A

Table 2. Layout Implementation

Results

	Windows_ANSI ColemakIm	Windows_ANSI Qwerty	Windows_ISO ColemakIm	Windows_ISO Qwerty
median	3.85	6.20	3.85	6.95
mean	4.02	6.32	4.02	6.71
std	1.65	3.29	1.65	3.33
min	1.00	1.00	1.00	1.00
max	7.80	11.90	7.80	11.90

Table 3. Difficulty of different layouts

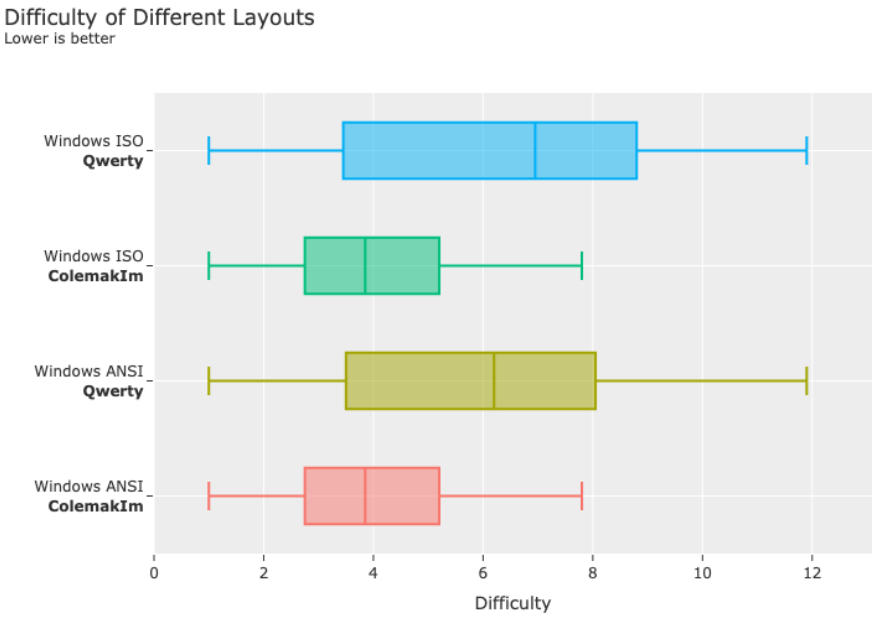

Figure 22. Difficulty of different layouts

Limitation

1. This keyboard layout is not the most efficient keyboard layout, as ease of transition from Colemak is kept in mind
2. Adopters of this keyboard layout have to learn the new changes

3. Due to a learning curve, reaching the speed of one's previous layout (such as Qwerty), may take a few months. It took me 3 months just to reach the same typing speed on Colemak as Qwerty. (Colemak-Im was made over months of research)
4. In keyboards with 2 key rollover, large effort is required for typing 6, 7, 8, 9, %,?, <, >
5. 2 keypresses are required for ; which may affect programmers of languages such as C, C++, Java
6. Further testing may highlight more limitations

Future Scope: Evaluation

A tester must be developed to quantitatively prove how a layout is better than another. Various types of scenarios should be tested, such as:

1. Regular prose typing
2. Programs
3. Compiler languages such as C, C++, Java involving ; , {}
4. Interpreted languages such as Python, Javascript with a simpler syntax
5. Random paragraphs

Conclusions

List of Figures

List Of Images

- Figure 1. Freestyle2 Ergonomic Keyboard
- Figure 2. Angled Keyboard
- Figure 3. Kinesis Advantage 2
- Figure 4. Qwerty Keyboard Layout
- Figure 5. Dvorak Layout
- Figure 6. Workman Layout
- Figure 7. Colemak Layout
- Figure 8. Colemak Mod-Dhm Layout
- Figure 9. ANSI Standard
- Figure 10. ISO Standard
- Figure 11. Keyboard effort grid, colour-coded by finger that should be used
- Figure 12. Keyboard effort grid, colour-coded by effort
- Figure 13. Numbering System for Normal Layouts
- Figure 14. Numbering System for Wide Layouts
- Figure 15. Colemak-Im Single Press Layer (Base Layer)
- Figure 16. Colemak-Im Shift Layer
- Figure 17. Colemak-Im Left Meta Layer
- Figure 18. Colemak-Im Right Meta Layer

- Figure 19. Colemak-Im Right Meta, Left Meta Layer
- Figure 20. Colemak-Im Right Meta, Shift Layer
- Figure 21. Colemak-Im Right Meta, Left Meta, Shift Layer
- Figure 22. Difficulty of different layouts

List of Tables

List Of Tables
- Table 1. Mappings Explanation
- Table 2. Layout Implementation
- Table 3. Difficulty of different layouts

Bibliography

[1] "Definition of ERGONOMICS." https://www.merriam-webster.com/dictionary/ergonomics (accessed Apr. 19, 2022).

[2] M. J. Gerard, T. J. Armstrong, A. Franzblau, B. J. Martin, and D. M. Rempel, "The Effects of Keyswitch Stiffness on Typing Force, Finger Electromyography, and Subjective Discomfort," *Am. Ind. Hyg. Assoc. J.*, vol. 60, no. 6, pp. 762–769, Nov. 1999, doi: 10.1080/00028899908984499.

[3] "Keyboard Ghosting Interactive Demonstration | Applied Sciences | Microsoft." https://www.microsoft.com/applied-sciences/projects/anti-ghosting-demo (accessed Feb. 15, 2023).

[4] "Keyboard Matrix Help." https://www.dribin.org/dave/keyboard/one_html/ (accessed Feb. 12, 2023).

[5] "KINESIS Freestyle2 Ergonomic Keyboard for PC (9" or 20" Separation) (9" Separation)." https://www.amazon.com/Freestyle2-Ergonomic-Keyboard-Standard-Separation/dp/B00CMALD3E (accessed Feb. 10, 2023).

[6] "Amazon.com: Microsoft Ergonomic Keyboard - Black. Wired, Comfortable, Ergonomic Keyboard with Cushioned Wrist and Palm Support. Split Keyboard. Dedicated Office Key. : Electronics." https://www.amazon.com/Microsoft-LXM-00001-Ergonomic-Keyboard/dp/B07Y414QXJ (accessed Feb. 15, 2023).

[7] "KINESIS Advantage2 Quiet LF Ergonomic USB Keyboard (KB600LFQ)." https://www.amazon.com/Kinesis-Advantage2-Ergonomic-Keyboard-KB600LFQ/dp/B07K1SMRGS (accessed Feb. 15, 2023).

[8] N. machine-readable author provided D. assumed, "Qwerty Keyboard Layout," Jan. 12, 2006. https://commons.wikimedia.org/wiki/File:KB_United_States.svg (accessed Mar. 21, 2023).

[9] "Dvorak Keyboard Layout." https://commons.wikimedia.org/wiki/File:KB_United_States_Dvorak.svg (accessed Mar. 21, 2023).

[10] "Workman Keyboard Layout." https://commons.wikimedia.org/wiki/File:Workman_keyboard_layout.png (accessed Mar.

21, 2023).
[11] “Colemak.” https://colemak.com/ (accessed Apr. 19, 2022).
[12] “Ergonomic Keyboard Mods: Angle Mod.” https://colemakmods.github.io/ergonomic-mods/angle.html (accessed Feb. 21, 2023).
[13] “Ergonomic Keyboard Mods: Wide Mod.” https://colemakmods.github.io/ergonomic-mods/wide.html (accessed Feb. 21, 2023).
[14] “Ergonomic Keyboard Mods: Extend Layer.” https://colemakmods.github.io/ergonomic-mods/extend.html (accessed Feb. 21, 2023).
[15] “Ergonomic Keyboard Mods: Symbols Layer.” https://colemakmods.github.io/ergonomic-mods/symbols.html (accessed Feb. 21, 2023).
[16] “Ergonomic Keyboard Mods: Modifiers.” https://colemakmods.github.io/ergonomic-mods/modifiers.html (accessed Feb. 21, 2023).
[17] “Ergonomic Keyboard Mods: Curl.” https://colemakmods.github.io/ergonomic-mods/curl.html (accessed Feb. 21, 2023).
[18] “Colemak Mod-DHm.” https://colemakmods.github.io/mod-dh/#revision-history (accessed Apr. 10, 2023).
[19] T. Cellucci, “Developing Operational Requirements”.
[20] J. Harrington, “ANSI vs ISO: Explained,” *Switch and Click*, Feb. 03, 2020. https://switchandclick.com/ansi-vs-iso-layout/ (accessed Feb. 11, 2023).
[21] A. H. H. Onsorodi and O. Korhan, “Application of a genetic algorithm to the keyboard layout problem,” *PLOS ONE*, vol. 15, no. 1, p. e0226611, Jan. 2020, doi: 10.1371/journal.pone.0226611.
[22] K. Nivasch and A. Azaria, “Keyboard Layout Optimization and Adaptation,” *Int. J. Artif. Intell. Tools*, p. S0218213023600023, Jan. 2023, doi: 10.1142/S0218213023600023.
[23] M. Krzywinski, “Carpalx keyboard layout optimizer,” 2005.
[24] “Colemak Mod-DH - Layout Analysis Model.” https://colemakmods.github.io/mod-dh/model.html (accessed Apr. 09, 2023).
[25] “AutoHotkey v2 Quick Reference.” https://www.autohotkey.com/docs/v2/ (accessed Mar. 07, 2023).
[26] “Karabiner-Elements Documentation,” *Karabiner-Elements*. https://karabiner-elements.pqrs.org/docs/ (accessed Mar. 09, 2023).