LANGPAD: A COMPUTER PERIPHERAL THAT ALLOWS FOR EASY TYPING OF SPECIAL CHARACTERS USED IN LATIN-BASED LANGUAGES

ENGLISH AND PROGRAMMING: FACTORS INFLUENCING THE WIDESPREAD USE OF ENGLISH IN THE SOFTWARE DEVELOPMENT FIELD

A Thesis Prospectus
In STS 4500
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Engineering

By
Christopher Hamilton

Fall 2021

Technical Project Team Members
Rohan Chandra
Emory Ducote
Rawan Osman
Pedro Rodriguez

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISORS

Richard Jacques, Department of Engineering and Society

Harry C. Powell, Department of Electrical & Computer Engineering

General Research Problem

How can the digital language divide be minimized?

The rapid expansion of computer capabilities coupled with increased availability of digital devices has resulted in an unprecedented lifestyle change for all humanity. While digital technology has improved quality of life and accessibility for many, there are countless instances where the benefits of technology are not distributed fairly among different groups of people. Despite the fact that most people in the world do not speak English as their first language, English is the most dominant language on the web (Lu, 2010). One's language determines the amount of information that is available to them and affects his or her user experience with technology (Becker, 2019).

LANGPAD: A Computer Peripheral That Allows for Easy Typing of Special Characters Used in Latin-based Languages.

How can the experience of those who wish to type in languages other than English be improved?

While computer technology has progressed at an extraordinary pace, the primary keyboard layout has remained the same since the early days of typewriters. The standard English keyboard layout, often referred to as QWERTY, has remained largely unchanged since it was designed in the 19th century. It is currently widely-used among people who speak Latin-based languages (Magazine & Stamp, n.d.). Since the QWERTY layout was originally developed for use in English-speaking countries, little thought was given to the use of diacritical marks. Diacritical marks are symbols that modify existing characters, and they are commonly used in non-English languages (Singh, n.d.).

To accommodate the use of these characters, numerous country-specific variations of the QWERTY layout have been created. Many people find it difficult to use these layouts because

users are forced to learn counterintuitive keyboard combinations in order to type special characters. Alternatively, users can access special characters through search features in text editors, but this method is time-consuming and inconvenient.

To address the difficulties of typing special characters used in foreign languages, my ECE capstone group is designing LANGPAD, a standalone touchpad device capable of typing these characters. It will connect to a computer using a USB cable and act as a keyboard input device. The touchscreen display will consist of pages of buttons labeled with specific special characters (fig. 1). Tapping on one of these buttons will type the corresponding character on the connected computer. There will be tabs along the top of the display to allow the user to quickly toggle between languages. By making a standalone device with an intuitive interface, it will provide a user-friendly experience for people who need to type characters in foreign languages.

Figure 1: LANGPAD touchscreen interface

The touchpad device will trigger key presses on Windows computers via a USB connection. A microcontroller will be programmed to register inputs from the touch screen and determine which character a user has pressed. The microcontroller will then communicate this information to the computer using the standardized USB HID protocol, causing the computer to type the corresponding character (fig. 2).

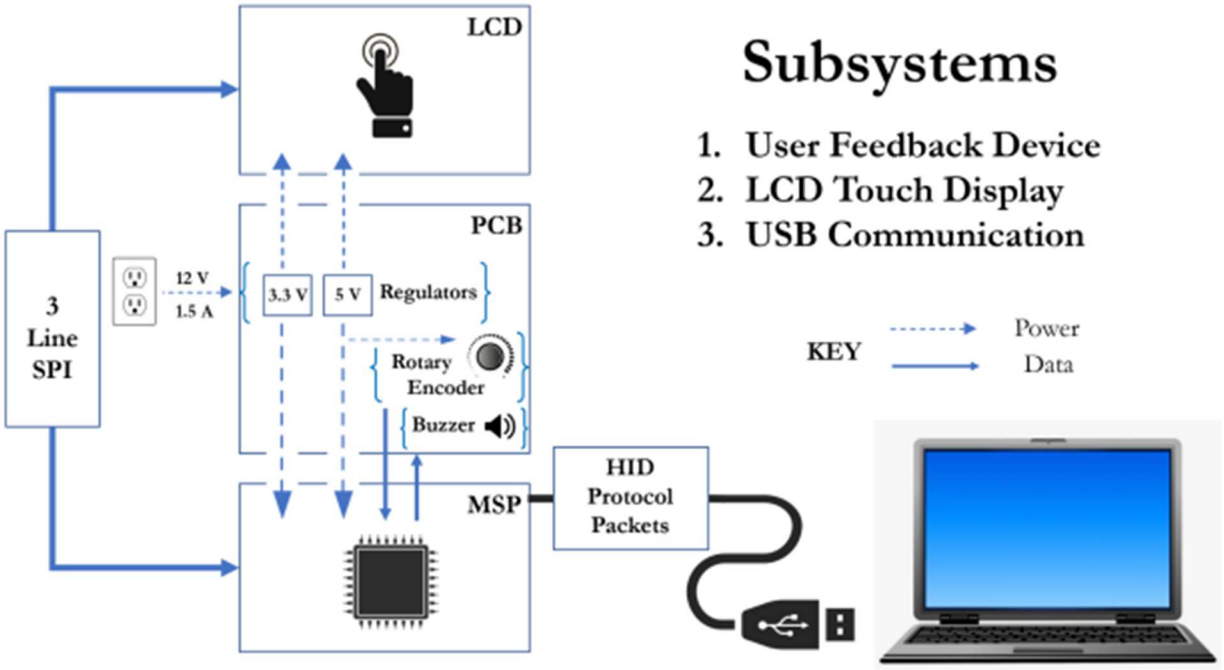

Figure 2: LANGPAD block diagram

This project will require two main areas for technical development, hardware design and embedded software. The hardware side will require a design of a custom circuit for the electrical connections between the different components (fig 2). NI Mulitsim will be used to develop the circuit schematic and NI Ultiboard will be used to design a custom PCB that will then be manufactured. On the embedded software side, a TI MSP family microcontroller will be used to execute the control software that we write. Code Composer Studio will be used for writing code and providing a debugging interface for the microcontroller.

The PCB design and the embedded software are planned to be completed before November 12th, 2021, providing over a month for integration testing. The project will be completed by the scheduled date for final ECE capstone demonstrations, December 17th, 2021. Collaborators for this project include Rohan Chandra, Emory Ducote, Rawan Osman, and Pedro Rodriguez. The technical advisor for this project is Harry C. Powell of the Electrical and Computer Engineering Department.

In order for our project to succeed, we will need to create a functioning prototype of the device we are designing. Making a prototype will require a realistic design and will establish a proof-of-concept for our idea. Beyond the capstone project, we are hoping that the device will become a commercially-available product that will make the experience of typing in non-English languages more intuitive.

English and Programming: Factors Influencing the Widespread Use of English in the Software Development Field

How does one's native language effect his or her ability to learn programming?

As there are numerous ways that language affects one's experience with technology, the impact of language on software engineering is often-overlooked. With technology's rapidly increasing role in modern life, computer programming is becoming an essential skill. Almost all of the popular programming languages used today are implemented using English keywords, and most programming curriculum is taught in English (Veerasamy, 2014). This brings into question the experience of non-native English-speakers learning to code. I expect to document how the widespread use of English in the programming field impacts these people, and I plan to investigate ways in which their experience can be improved.

Many of the most widely-used programming languages were developed in the United States by companies such as Bell Labs and Sun Mircrosystems (Nnass, 2020). While some popular programming languages like Python and Ruby were developed in countries where English is not the native language, these programming languages are still implemented in English (McCulloch, n.d.). Programming languages are made up of predefined keywords that a developer uses to write a program. The choice of these keywords is functionally arbitrary, but words such as “for” and “while” are commonly used because of their English meaning (Nnass, 2020). By assembling a language of meaningful keywords, programs are easier to comprehend. Those who cannot interpret the meaning of these words will not be able to take advantage of the intrinsic readability of programs. In a study conducted at CQUniversity, many Lybian students indicated that they had difficulty interpreting error messages (Nnass, 2020). Error messages are generated while running computer programs and communicate important information to developers. Meticulous reading of these messages is often required to interpret them correctly. Since error messages are predominantly generated in English, one’s English skills will affect their ability to use these messages effectively.

Another point of difficulty for those who do not speak English as their primary language is most programming classes use English as the language of instruction (Veerasamy, 2014). For example, Computer Science curriculum in India is taught in English, placing an increased burden on students lacking strong English skills (Soosai Raj et al., 2018). Non-native English speakers face the challenge of mentally translating content to their native language, increasing their cognitive load and decreasing their comprehension of the material (Guo, 2018). To substantiate the claim that there are significant difficulties pursuing education outside one’s native language, data mining has shown that non-native English speakers have high drop-out

rates in English undergraduate education (Kumar & Pal, 2012). Considering that 95% of the world's population does not speak English as their native language, the accessibility of programming education is potentially a widespread issue (Guo, 2018).

The difficulties associated with non-native English speakers learning programming skills have various parties of interest. The primary group of interest are those who do not speak English as their primary language. Those who cannot speak English will not be able to comprehend many aspects of English-based programming languages and will have difficulty consuming English-based education materials. Concrete efforts to improve the learning experience of these people place increased burden on the developers of programming languages and computer science educators. Beyond the directly affected parties, software employers must consider a potential hire's grasp of the English language and how it affects their ability to perform in a software engineering role.

Research in the area of language's impact on programming is limited, and the issues discussed above have not been thoroughly investigated (Alaofi, 2020). Given that the learning of computer programming depends on people's prior experiences and abilities, it is difficult to isolate one's language mastery as the variable controlling their learning outcomes (Nnass, 2020). Despite this limitation, there is ongoing work to address the problem of English-speaking ability affecting people's experience learning programming skills. Some major programming languages have alternate language configurations, such as the Chinese version of Python (McCulloch, n.d.). A group at West Virginia is attempting to provide empirical evidence of the impact of English-speaking ability on programming performance. They hope to justify the creation of an Arabic programming language for teaching programming fundamentals in students' native language (Ben Idris & Ammar, 2018). Additionally, PseuToPy is an attempt to create a natural

programming language with support for numerous languages (Wang, 2021). While the aforementioned solutions do not address the issue in its entirety, there is an argument for their validity. By teaching people programming fundamentals in their native language, students will be able to focus on learning core programming concepts instead of English comprehension.

Conclusion

In recent times, English has developed into a major world language. It dominates the field of technology, leaving those who speak other languages at a disadvantage. With tech giants like Google and Microsoft predominantly operating in the United States, a large amount of technology is designed with language support as an afterthought. People's language controls the quality of their experience with technology and the amount of information available to them. With technology's major role in modern life, people should have equal digital opportunities, regardless of what language they speak. To achieve this goal, it will require active efforts from various parties to develop accommodations that make the digital world inclusive of all languages.

References

Alaofi, S. (2020). The Impact of English Language on Non-Native English Speaking Students' Performance in Programming Class. *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, 585–586. https://doi.org/10.1145/3341525.3394008

Becker, B. A. (2019). *Parlez-vous Java? Bonjour La Monde != Hello World: Barriers to Programming Language Acquisition for Non-Native English Speakers*. 13.

Ben Idris, M., & Ammar, H. (2018). The Correlation between Arabic Student's English Proficiency and Their Computer Programming Ability at the University Level. *International Journal of Managing Public Sector Information and Communication Technologies*, *9*, 01–10. https://doi.org/10.5121/ijmpict.2018.9101

Guo, P. J. (2018). Non-Native English Speakers Learning Computer Programming: Barriers, Desires, and Design Opportunities. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3173574.3173970

Kumar, S., & Pal, S. (2012). Data Mining: A Prediction for Performance Improvement of EngineeringStudents using Classification. *World of Computer Science and Information Technology Journal*, *2*, 51–56.

Lu, L. (2010). *Digital Divide: Does the Internet Speak Your Language?* 4022–4025. https://www.learntechlib.org/primary/p/35226/

Magazine, S., & Stamp, J. (n.d.). *Fact of Fiction? The Legend of the QWERTY Keyboard*. Smithsonian Magazine. Retrieved September 27, 2021, from https://www.smithsonianmag.com/arts-culture/fact-of-fiction-the-legend-of-the-qwerty-keyboard-49863249/

McCulloch, G. (n.d.). Coding Is for Everyone—As Long as You Speak English. *Wired*. Retrieved September 27, 2021, from https://www.wired.com/story/coding-is-for-everyoneas-long-as-you-speak-english/

Nnass, I. (2020). *Identifying and solving issues with acquiring skills in computer programming for non-English speakers* [Thesis, CQUniversity]. https://figshare.com/articles/thesis/Identifying_and_solving_issues_with_acquiring_skills_in_computer_programming_for_non-English_speakers/13406825/1

Singh, V. (n.d.). *How the QWERTY keyboard shapes the way we communicate*. The British Academy. Retrieved September 27, 2021, from https://www.thebritishacademy.ac.uk/blog/summer-showcase-2020-how-qwerty-keyboard-shapes-way-we-communicate/

Soosai Raj, A. G., Ketsuriyonk, K., Patel, J., & Halverson, R. (2018). *Does Native Language Play a Role in Learning a Programming Language?* 417–422. https://doi.org/10.1145/3159450.3159531

Veerasamy, A. K. (2014, February 14). *Teaching English based programming courses to English learners/non-native speakers of English*.

Wang, P. (2021). PseuToPy: Towards a Non-English Natural Programming Language. *Proceedings of the 17th ACM Conference on International Computing Education Research*, 429–430. https://doi.org/10.1145/3446871.3469787