PauseBoard: A Force-Feedback Keyboard for Unintrusively Encouraging Regular Typing Breaks

Lewis Bell
University of Bristol
Bristol, UK
lewis.bell@bristol.ac.uk

Charlie Harding
University of Bristol
Bristol, UK
charlie.harding@bristol.ac.uk

Jay Lees
University of Bristol
Bristol, UK
jay.lees@bristol.ac.uk

Ben Lee
University of Bristol
Bristol, UK
bl16266@bristol.ac.uk

Will Smith
University of Bristol
Bristol, UK
ws16062@bristol.ac.uk

Abstract
Maintaining positive digital well-being has become essential as we spend more and more time working at desks in offices, interacting with computers and typing for hours at a time. In this paper we present PauseBoard: a computer keyboard designed to unintrusively encourage users to take regular breaks. Through the use of motorised linear potentiometers, the force required to activate each key is gradually increased, until a maximium point, after a period of productivity. Preliminary testing shows that 75% of users respond well to this novel gentle encouragement, being reminded to take breaks while still being able to concentrate and finish their current task.

Author Keywords
Keyboard, force-feedback, digital well-being, stress, productivity

Introduction
In recent decades, there has been a large increase in the number of hours the average office worker spends working at a computer, with some studies putting this number as high as 1700 hours per year [9]. Studies have also shown a strong correlation between overworking and both increased levels of stress and decreasing job satisfaction [13]. This project aims to provide a functional solution for computer users in maintaining a healthy working life by reminding

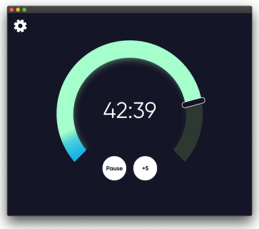

Figure 1: Main PauseBoard UI indicating time elapsed, time remaining and when the resistive force will increase.

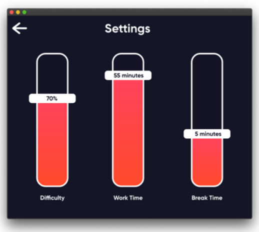

Figure 2: Settings UI where the system timings can be modified, along with the relative resistance.

them to take regular breaks.

Several solutions to this problem have been proposed by both academic and commercial organisations, the majority of which utilise a timer to alert users in a disruptive manner that it is time to take a break [1, 2, 17, 18]. Such methods have proven to be effective in getting users to take breaks, however they may intrude on work users are trying to complete; demanding their attention be taken away from their work immediately.

To combat this issue we propose PauseBoard, a keyboard that uses custom hardware to unintrusively remind users to take breaks at regular intervals. By using motorised linear potentiometers under each key, the voltage across each motor can be individually adjusted to increase the resistance it generates, thereby increasing the required force to activate each of the keys. By gradually increasing the force required to press the key over time, the user is slowly encouraged to take a break. However, they are still able to continue their work without distraction until they reach a natural stopping point, as the keys reach a high, but not impractical, level of resistance. When they return from their break, the user will find the keys have returned to their regular resistance, so they can continue working as normal until the next break. A prototype of PauseBoard can be seen in Figure 3.

Walkthrough

PauseBoard is designed to be a direct replacement for a standard USB keyboard, requiring only the PauseBoard software to be installed on the computer. The main user interface is shown in Figure 1, showing how much time is remaining in the current working period. The colours indicate the time when the keyboard will become gradually harder to use, culminating in the user taking a break. After

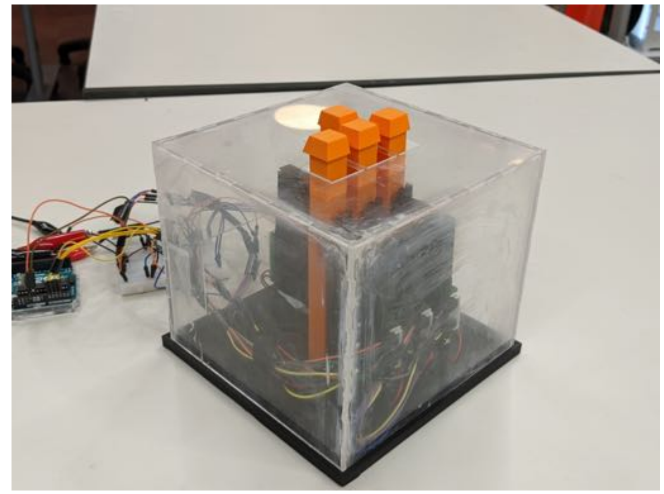

Figure 3: PauseBoard Prototype

taking a break from typing, the timer resets and the user can return to work. In the final implementation, the user will be able to configure time intervals that best suit their working environment, as well as how much force should be applied to the keys. This will be done through the settings user interface, as shown in Figure 2.

Related Work

Keyboard design and interaction have been the subject of many academic papers [11, 13, 14], where multiple aspects of the device have been monitored and changed to determine the effect on users.

Several papers have considered how existing keyboard technology can be modified in order to obtain further information from a keyboard, in addition to which keys have been pressed. Dietz et al. present a keyboard [11] that is

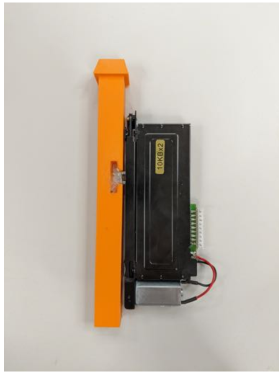

Figure 4: Motorised linear potentiometer and 3D-printed keycap

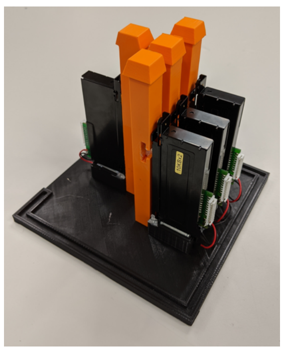

Figure 5: Motorised linear potentiometers positioned vertically in 3D-printed base

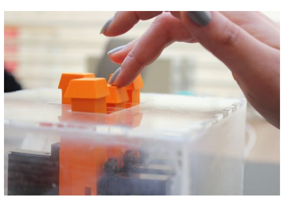

Figure 6: Keys on the finished prototype

capable of monitoring the precise force level on every key, suggesting potential applications in music composition or gaming. However they do not provide a formal analysis of an individual use case.

Previous papers have also considered using variable resistance keys to minimise typing mistakes. Hoffman et al. present TypeRight [14], a keyboard that prevents typing errors by increasing resistance on keys that would lead to a spelling error. They conduct a formal study between TypeRight and a traditional keyboard, which concludes that the addition of tactile feedback successfully decreases error rates by 46 %. This positively suggests that modifying keyboard resistance is an effective way to interact with users unintrusively.

Finally, additional research has been conducted into the content of typing breaks - specifically Morris et al. consider the effect of combining interactive activities with existing “break reminder” software [17]. Their focus is mainly on increasing the uptake of such breaks, as well as retaining users for extended periods of time. In their initial research, they find that users “ignore [or] turn off existing software shortly after installing it”, suggesting existing break reminder tools have plenty of scope for improvement. Their paper concluded that increasing interactions within typing breaks increased uptake when combined with off-the-shelf break reminder software.

Design

The PauseBoard design is comprised of three components: the hardware to enable variable key resistance, the enclosure to hold the hardware, and the software to detect key presses and control resistance.

Hardware

When trying to find a way to change the pressure of keyboard keys, the first solution considered was to use electromagnets under the keys, which would then repel the keys back up when they were pressed. Whilst this may work for a single key, on a full keyboard there would be the problem of magnets interacting with one another, creating forces that would put strain on the keyboard housing. However, by fitting shielding to counteract this, the added weight would make the system prohibitively heavy.

Instead, other ways were considered to change the force applied to the keys. As the keypress state will need to be measured, for which linear potentiometers would be useful, motorised linear potentiometers [5] were investigated to both measure the keypress and supply the resistance against it. Four such motorised potentiometers are positioned vertically in a 3D-printed base, with 3D-printed keycaps attached to them (Figure 4). The potentiometers are connected to an Arduino Uno [8], which reads how far down each key is pressed, and alters the resistance as signalled by the host computer.

To change the resistance of the keys, the voltage across each motor of the linear potentiometers is varied. Since the operating voltage of the motor in the potentiometers is 6 V to 11 V and the Arduino can only provide a voltage of 5 V [4], transistors were considered to source power from pairs of 9 V batteries. However, due to the types of transistors we had available, it was not possible to supply the full 11 V to achieve the key activation resistance required to adequately signal a break. Because of this, a Wizard of Oz approach [15] is taken to achieve a greater force: for demonstration, a variable power supply is used, which allows switching between preset voltages.

Figure 7: User Study Setup

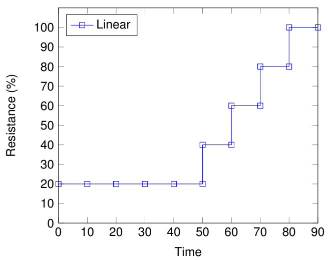

Figure 8: Resistance over time for the linear experiment

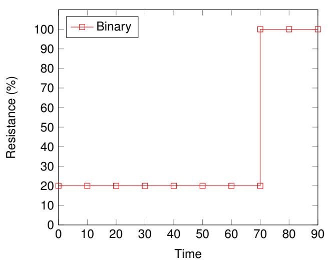

Figure 9: Resistance over time for the binary experiment

Enclosure
The electronics are housed in a laser-cut transparent acrylic enclosure measuring 180 mm × 180 mm, with the key caps extending out of holes in the top surface.

The complete prototype can be seen in Figure 3.

Software
The Arduino detects the analogue input from each key to determine if it is pressed, i.e. if the resistance reading from the potentiometer is over a preset threshold. If so, it sends a signal to the connected computer, repeating that signal at regular intervals until the key is released. This communication is done over a serial connection through the Arduino's USB host port.

Attached to the Arduino is a host computer, with custom software installed [7] listening for connected Arduinos. Once connected, the serial port is read for keypresses, which are then relayed to the operating system using AppleScript [3]. A graphical user interface is also provided, as seen in Figure 2, which allows configuration of timer settings.

Evaluation
We completed an informal user study during term time with sixteen participants from the University of Bristol, firstly to investigate the hypothesis that increasing the resistance of keys is an effective way to encourage users to take breaks, and secondly to ascertain whether a linear or binary variance works best to encourage breaks. The user study was a qualitative assessment, and was split into two parts. Users were first asked to play a short game of Tetris using the keyboard without any information provided regarding the purpose of the device. This was to investigate what users believed the purpose of the device to be after using it for a short period of time, and for the users to get some hands-on experience using the keyboard.

Table 1: Comparing the sentiment of questionnaire responses for the two different experiment configurations

	Response type		
	Linear	**Step**	**Combined**
Strictly Positive	67.2 %	59.4 %	63.3 %
Neutral	6.3 %	6.3 %	6.3 %
Strictly Negative	26.6 %	34.4 %	30.5 %

The users were split into two groups, each playing the game with a different keyboard resistance configuration. The first group would play with a 'linear' increase in key resistance, as seen in Figure 8, meaning that the resistance would increase in small increments after a set amount of time before reaching the maximum resistance. The other group used a binary resistance, as seen in Figure 9, meaning the resistance would immediately increase from minimum to maximum after a set amount of time. The experiment was a Wizard of Oz experiment [15], with the variance and time being controlled manually, however this was not noticed by any of the participants.

After playing the game, users were then asked to answer a short questionnaire which consists of twelve questions, partly inspired by the System Usability Survey [10]. Most of these questions used a Likert scale [16], others used longer answers. Overall, the Likert scale questions showed positive feedback to the idea (Table 1, Figure 10). 63 % of responses overall were strictly positive and when switching to the linear configuration this number jumped to 67 %.

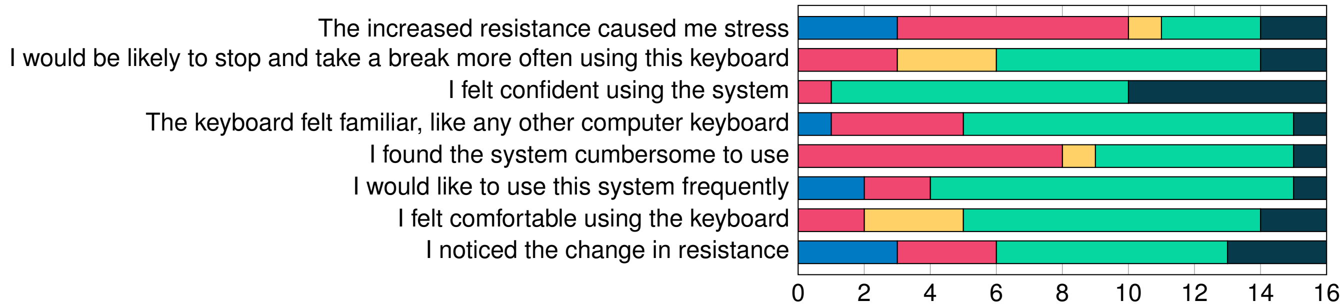

Figure 10: Overall responses to short-answer questions. Legend: Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree

"Easy to use and beneficial for taking breaks."
Participant 1

"Could definitely feel the resistance and made me think about how much I was working."
Participant 9

"Could see it as a positive way to help people take breaks, I like the possibility for different use cases via customisation."
Participant 11

Figure 11: Quotes from participants

For the most part, varying the resistance configuration yielded little difference in the distribution of the responses to the Likert scale questions. However, for the question regarding the likelihood of taking a break, the linear resistance type produced a significantly higher likelihood, with 87.5 % of participants agreeing in the linear case, compared to only 37.5 % for the binary case. This is perhaps because the introduction of the high resistance was more gradual and therefore less intrusive until maximum resistance was reached.

Regarding users guessing the purpose of the keyboard, all the correct guesses came from people who were using the linear test, none from the binary test. The variety of guesses suggested that users could envisage various use cases for the keyboard. They ranged from correctly guessing encouragement of breaks, to making Tetris more difficult, to finger exercise. This supports the opinion of Dietz et al. [11].

All but one participant had an overall positive reaction to the concept. 75 % of participants responded that they would like to use this keyboard frequently given a full and polished design. The feedback from the longer answer questions frequently alluded to the possibilities for different use cases. Participants were highly interested in configuring the resistance of the keys to suit their needs. One participant noted that by having an increased resistance on the keys, they were made more aware of the load on their fingers and therefore the amount of time they were typing. This suggests that the keyboard could potentially have an impact on both mental and physical well-being. On the other hand, one user noted that if the resistance were too high for too long, this may lead to repetitive strain injury [6].

Negative feedback usually referred to problems with the prototype rather than the concept. A significant number of these comments implied that the resistance of the prototype keyboard was not high enough to have a strong influence on them stopping work. Other comments made it clear that users have different preferences on key travel, resistance, and even sound. When commenting that this keyboard may not be effective for them, participants almost always believed it may still be effective for others. A few people commented that they would find this keyboard stressful when nearing deadlines of high-stress tasks, however this feedback could be applied to any current break-encouraging system. The implementation of configurable software could combat this with the ability to suspend the reminder. Some people simply do not want to take breaks at regular intervals. One comment suggested we were re-inventing the sand timer, however the sand timer is passive and can be more easily ignored. A couple of people found the higher resistance stressful, though this may have been related to playing a game such as Tetris.

Although the user study returned positive results, these may be negatively skewed by participants making judgments based on the prototype which was far from the final product. This was a result of the questions not differentiating between the prototype and the concept.

Future Work

It would be incredibly beneficial to extend the prototype PauseBoard to a full set of keys in order to run a full side-by-side trial against a standard computer keyboard and a standard break timer in a realistic work environment.

More research is required to find the optimal levels of resistance, though this is likely to be specific to each user. Additionally, a typing exercise should be investigated within a workplace environment, as different levels of force are exerted when playing a game such as Tetris on a restricted set of keys.

Beyond the obvious incremental changes, other possible applications for the technology can be considered.

One such application could be the ability to develop a keyboard which adaptively models other kinds of mechanical key switches. One global leader in key switch manufacturing, Cherry MX, currently develop 13 unique varieties of switches [12], each with their own properties, popular with different individuals.

With its high resolution of key-depth sensing and the ability to make micro-adjustments to the voltage across the keys, PauseBoard could be adapted to emulate the particular profile of any key. This could be achieved by increasing the resistance as the key is pressed until an 'activation point' is reached, at which point the pressure is temporarily decreased. This profile could be adapted in real time according to user desires, and could switch for the context of the task they are conducting.

As one respondent mentioned, our experiment could come with the risk of increased repetitive strain injury [6]. Further research could therefore be conducted to determine whether increasing the key resistance for short periods of time does increase this risk, and what changes could be made to mitigate this.

Conclusion

This paper has presented PauseBoard, a force-feedback keyboard that varies resistance to unintrusively encourage users to take regular breaks. This differs to current approaches which tend to be more disruptive of user workflow. Despite the limitations of the demonstration prototype, users' responses are largely positive, indicating promise for the concept. Further studies need to be conducted to evaluate the idea in a real-world context. There is potential to further explore, particularly in allowing users to dynamically adjust the tactile properties of their keyboard.

REFERENCES

[1] 1994. (Aug 1994). http://www.lcdf.org/xwrits/

[2] 2002. Workrave. (Sep 2002). http://www.workrave.org/

[3] 2016. AppleScript Language Guide. https://developer.apple.com/library/archive/documentation/AppleScript/Conceptual/AppleScriptLangGuide/introduction/ASLR_intro.html. (Jan 2016).

[4] 2018. The Full Arduino Uno Pinout Guide [including diagram]. (Nov 2018). https://www.circuito.io/blog/arduino-uno-pinout/

[5] 2018. PSP60-082A-103B1 - Potentiometer, 10 kΩ, 500 mW, ± 20 %, Pro Audio PSP Series, Linear. https://uk.farnell.com/bourns/psp60-082a-103b1/motorized-slide-pot-10k-20-0-5/dp/2773948. (Jan 2018).

[6] 2018. Repetitive strain injury (RSI). (Nov 2018). https://www.nhs.uk/conditions/repetitive-strain-injury-rsi/

[7] 2019. PauseBoard: Source Code. (Dec 2019). https://github.com/jaylees14/PauseBoard

[8] Arduino. 2011. Arduino Uno Rev3. https://store.arduino.cc/arduino-uno-rev3. (2011).

[9] Grant Bailey. 2018. Office workers spend 1,700 hours a year in front of a computer screen. (Jul 2018). https://www.independent.co.uk/news/uk/home-news/office-workers-screen-headaches-a8459896.html

[10] John Brooke. 1996. SUS: A quick and dirty usability scale. (1996).

[11] Paul Dietz, Benjamin Eidelson, Jonathan Westhues, and Steven Bathiche. 2009. A practical pressure sensitive computer keyboard. 55–58. DOI: http://dx.doi.org/10.1145/1622176.1622187

[12] Cherry GmbH. 2019. CHERRY MX - Our best keyboard switches. (Jul 2019). https://www.cherrymx.de/en

[13] Javier Hernandez, Pablo Paredes, Asta Roseway, and Mary Czerwinski. 2014. Under Pressure: Sensing Stress of Computer Users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 51–60. DOI: http://dx.doi.org/10.1145/2556288.2557165

[14] Alexander Hoffmann, Daniel Spelmezan, and Jan Borchers. 2009. TypeRight: a keyboard with tactile error prevention. 2265–2268. DOI: http://dx.doi.org/10.1145/1518701.1519048

[15] John F. Kelley. 1985. *CAL — a natural language program developed with the OZ paradigm: implications for supercomputing systems*. Research Report RC 11324 (#50219). IBM Thomas J. Watson Research Center, Yorktown Heights, NY. 19 pages.

[16] Rensis Likert. 1932. A Technique for the Measurement of Attitudes. *Archives of Psychology* 140 (1932), 1–55.

[17] Dan Morris, A. Brush, and Brian Meyers. 2008. SuperBreak: Using Interactivity to Enhance Ergonomic Typing Breaks. *Conference on Human Factors in Computing Systems - Proceedings* (Jan 2008), 1817–1826. DOI: http://dx.doi.org/10.1145/1357054.1357337

[18] Arthur Saltzman. 2011. New Comprehensive Healthy Computing Module. (Sep 2011). http://paratec.com/