How do breaks affect focus / productivity for developers? A tool-based approach

Daniel Gordon Jones
danielgordon.jones@uzh.ch

Jose Antonio Sebastián Dávila
joseantonio.sebastiandavila@uzh.ch

Tilda Karin Maria Myrsell
tildakarinmaria.myrsell@uzh.ch

ABSTRACT

Taking breaks from the computer is an important part of information-centred work contexts. Existing research shows that break reminder tools tend to fall into one of two categories: either they are too intrusive or not intrusive enough. In this article, we present the results from our break reminder tool *BreakBuddy*, which we have developed as a unique and personal approach to reminders. The importance of taking breaks and how important they are for focus, health, and productivity during work has been carefully addressed through existing research, which we attempt to leverage and incorporate into the tool. In this paper, we present results that suggest *BreakBuddy* can help users to take breaks through using the character to graphically represent the need to take a break. We also present the utility of *BreakBuddy*, how it helped participants to be more aware of taking breaks, and how the aspect of fun in the form of the customised approach made the user more inclined to continue using the application.

MOTIVATION

Contrasting approaches have been enacted when attempting to implement research on how best to get people to take breaks from the computer, and when to do it, into tools. Break reminder tools tend to fall into one of two categories: either they are too intrusive or are not intrusive enough. Tools that are too intrusive, such as alerts that take over the screen [7] [8], tend to irritate users and ultimately deter them away from further usage, whilst tools that are not intrusive enough [10] simply go unnoticed, ergo ineffective. In this study, guided by a body of prior research related to break timing [1] [2] [3] [4] [5], we attempt a different and unique approach to tool implementation; the primary goal of which is to get the user to take breaks effectively, without dissuading them from further use (as to fit between the two categories of existing tools).

To try and achieve this, we implement a four-phase intrusiveness level shift, whereby a customisable animated character changes across four equal intervals, gradually becoming more intrusive each phase. The changes are conveyed graphically via the character, primarily through altering its facial expression, to gradually appear more and less focussed. Once the user set break timer is exceeded, the character will enter the most intrusive state, which involves sending out desktop notifications and randomly moving around the screen. For the reminder intervals, our tool leverages in part the approach studied by Luo et al. in their tool, *Time for Break* [1], where break prompts are set flexibly by the user.

INTRODUCTION

Breaks have a positive relationship with how you feel and with productivity at work. The reason why breaks are so important is because it gives the body time to facilitate recovery and it can allow you to boost your performance [11]. Break reminder tools exist to remind the user to take a break when they otherwise wouldn't, which studies have shown is important with regards to both focus and productivity [2] [3] [4]. DeskTime has presented the results of its study that 52 minutes of work and then a 17-minute break leads to maximum productivity [5]. There are also several existing studies and tools that act as a break reminder tool one of these is Strechly which is a tool that takes over the screen for a certain period [7]. Another example of a tool is UbiFit Garden gradually changing the background based on how much of the task has been completed [9]. These tools are either perceived as too disturbing or that they are not intrusive enough to take a break.

BreakBuddy is a flexible and personalised break reminder utility developed in Electron. Upon starting the application, the user is prompted to select the time before a break should be taken, the duration of the break, and to create and personalise their animated character. The character remains on top of the current application at all times and can be dragged around the screen at will. The avatar keeps track of the elapsed time, using the phase-based approach to subtly indicate how far or close it is until a break should be taken. The character gradually becomes more intrusive through changing its facial expression, opacity, position, and size. Once the time before a break should be taken is exceeded, the character enters the most intrusive stage, where it sends out desktop notifications and forcefully repositions itself on the screen every few minutes.

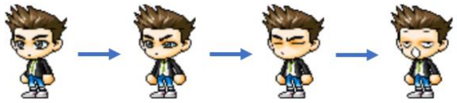


Figure 1: Example four-phase facial expression shift in the avatar

In this study, we analyse data received from a survey of how users experienced our application to understand whether adaptive support and a gradually increasing intrusive approach is effective in getting the user to take breaks. Our analysis suggests that *BreakBuddy* helps users take breaks when the character changed. We discovered that the intrusion factor of the notifications had an impressively high value and overall, the users did not find the application too intrusive. We also found that users found *BreakBuddy* useful and that it helped them be aware of taking breaks, and it was more fun than other break reminders they used.

The research presented contributes to our understanding of how different break reminders affect users.

RESEARCH QUESTIONS

- **RQ1**: Is a gradually scaling intrusiveness approach effective in getting the user to take breaks without frustrating them?
- **RQ2**: Does a glanceable, customisable animated character support taking breaks and the desire to keep doing so?

A SCALING APPROACH TO INTRUSIVENESS

We address **RQ1** by empirically analysing the results obtained through a user survey, focussing on both the implementation of the scaling intrusiveness levels in the application and how frustrated the user felt after using it. We focus on section 2 of the user survey, which asks questions such as “How aware were you of the character changing over time?” (1 – 10 Likert scale). We hypothesise that the blended intrusiveness approach combined with the always-on-display aspect, as opposed to the exclusively low or high intrusiveness approaches, will help to not frustrate the user whilst remaining effective.

EFFECTIVENESS OF CUSTOMISATION

RQ2 is also addressed empirically, through surveying and collecting qualitative feedback from users of the application, focussing on the character customisation element. In the survey, we try to understand how important it was for the user to be able to customise their character and whether this played a role in them continuing to use the tool. We focus on section 1 of the user survey, which asks questions such as “How important was it for you to be able to customise the character to your liking?” (1 – 10 Likert scale). We hypothesise that the personalised aspects of the tool (naming and customising the BreakBuddy) will positively support and tie in with the aspects of what made the user take a break and the likelihood of them wanting to keep doing so through the application.

RELATED WORK

A. Importance of taking breaks and when best to take them

The importance of taking breaks and their implications towards both focus and health whilst working has been rigorously addressed through existing research, with the overwhelming consensus swaying heavily in favour [2] [3]. Scheduling when to take breaks generally corresponds to optimal periods of focus in individuals, who themselves have varying periodic and daily limits [4], making the issue harder to quantify. One popular time tracking application, DeskTime, suggests that 52 minutes working and then a 17-minute break leads to the highest productivity, obtained by studying the most productive 10% of its users [5]. The data analysed by DeskTime is further supported by an exhaustive study by Ariga et al., which advocates for an hourly break [6]. Whilst 30-to-60-minute intervals are shown to work for most people, ultimately, the time a break should be taken varies both per individual and per task, requiring fine tuning to truly perfect. The takeaway from this research with regards to our project caused us to weigh up the pros and cons of a flexible approach versus a pre-defined one, resulting in us trying to achieve both through not allowing total flexibility (instead incorporating a range based on the research).

B. Existing tools of interest

We first examined *Strechly* [7], which is a highly intrusive and polarising reminder utility that alerts the user of when to take a break by completely taking over the screen for a set amount of time. The purpose behind examining this application was to gain insight into how one of the most intrusive applications works and what the pros and cons of such an approach are. To begin with, the application does not actively indicate how close the user is to their break, so the notifications often come as a complete surprise. Additionally, *Strechly* does not allow the user to flexibly set their own schedule, but instead insists on *microbreaks* (every 10 minutes) and *breaks* (every 30 minutes). The application seems to work best for individuals that really seek (and perhaps need) the brute force, intrusive approach to taking breaks. Although examining *Strechly* demonstrated to us how one of the highly intrusive applications can work and be effective, we were not convinced enough about the approach or features of the application to incorporate it into *BreakBuddy* – instead opting to head more into the opposite direction.

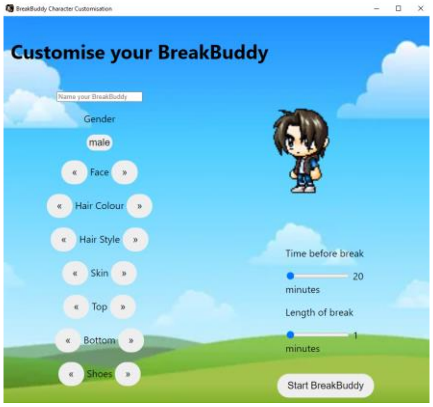
Workrave [8] takes quite a different approach to the previously discussed tool and, whilst still rather intrusive, is much less so than *Strechly*. The tool utilises a variety of methods to alert the user, for example, a break timer window that is constantly displayed, showing the progress of three different types of breaks (microbreaks, rest breaks, daily limit), and a pop-up alert. During the break, the application presents the user with a stretching exercise demonstration, which the user is encouraged to take part in. The application is built around the theme of preventing users from developing occupational diseases, such as carpal tunnel syndrome and repetitive strain injury, which naturally does not set out to address the same problems as us. Despite this, we were able to obtain good takeaways from this tool – in particular the constantly displayed timer, which we incorporate as the animated character in *BreakBuddy*. Based on reviews of the application on *Sourceforge*, the users generally appear to appreciate the flexibility of approaches and overall functionality of the tool, which was also important for us to consider due to the similarities of our approaches.

UbiFit Garden [9] is a legacy tool that uses a phase-based approach to indicate degrees of task completion. The application works by displaying a garden in the background of the phone, gradually growing more beautiful as tasks are marked completed. The core idea of the application is that, instead of getting the user to complete tasks through reminders or notifications, the user glances over at their background (which they know is linked to the number of tasks they have completed) and is motivated to continue completing tasks, growing their garden as a result. Consequently, the application is very unintrusive, as it only ever statically occupies part of the phone’s background. With *BreakBuddy* the problem of reminding the user to take a break is approached in a similar manner; using the animated character to visually represent the need to take a break.

1 APPROACH

After finalising the fundamental requirements of the project, we had to figure out which technologies made the most sense. Since one of the hard requirements of the project involved tracking mouse movement and keyboard presses globally (as to detect whether user was active or not), the initial choices were already limited to desktop native applications. Another problem was then deciding on which platform to develop the application on, where committing to either Windows or OS X would only serve to further limit the number of potential users and testers from an already rather low projected number. Since software developers were a particular target group of the application, one early idea to circumvent the problems was to develop the application within the VS Code IDE as an Extension, though we quickly realised this could be improved upon. By utilising the same tech stack as VS Code, Electron, cross-platform support could be achieved without compromising on form or functionality.

Before the core functionality could be implemented, the start screen was designed and developed as the first point of access to the application from the user. In the start screen, the user can name and personalise many different visual elements of their character. The user is also able to set two important values related to the latter functionality – the time before a break should be taken (default 60 minutes) and how long the break should last (default 3 minutes). The overall design philosophy was to create a very simple and intuitive screen, opting to sacrifice descriptions and additional text for simplicity. We took this approach primarily to facilitate our research questions, looking at the customisation and testing the flexibility hypothesis [1] (with regards to allowing the user to set their own timers).

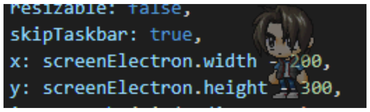


Time before break
20
minutes
Length of break
1
minutes

Figure 2: The starting screen of the application

Like the ideas and core concepts presented in *UbiFit Garden* [9], the goal of *BreakBuddy* was to always be accessible to and in view of the user. An additional requirement was that it should be easy to reposition the character, such that it does not end up in the way of something. These decisions were taken in-line with our research questions and to help reduce frustration overall. To achieve such functionality, a customised window was created that was: transparent (as to only display the character), always on top, could not be closed, resized, or minimised, but could be moved around.

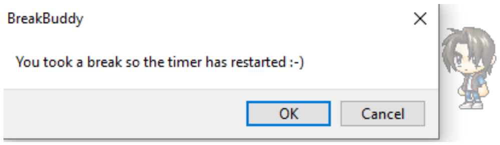
Whilst not a flawless solution (there are a few known bugs that probably cannot be fixed without a proper native solution), the custom window fit all the requirements and performed well during the tests and study, across the different operating systems.



```
resizable: false,
skipTaskbar: true,
x: screenElectron.width - 200,
y: screenElectron.height - 300,
```

Figure 3: Example of the always on top functionality within an IDE

After the character could successfully be displayed, the next challenge was to gather computer interaction, which was achieved through tracking mouse and keyboard input and then using this to either keep the break timer running (and therefore triggering the character shifting events) or to stop it. If no interaction was detected for the amount of time set for the "Length of break", the break timer was reset and a popup displayed, acknowledging that a break had been taken. Later in the development a feature to manually reset the timer was implemented, which was not only implemented to help remedy bugs, but also for general use. With the successful implementation of this core functionality, all that remained was to set the changing behaviours of the character whilst time passed i.e., increasing the intrusiveness in phases.

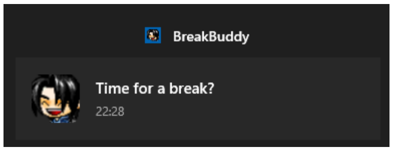


BreakBuddy

You took a break so the timer has restarted :-)

Figure 4: An example of the popup displayed, acknowledging that a break had been taken by the user

To implement the shifting levels of intrusive, first the phases are calculated according to: $P = \left\lfloor \frac{T}{3} \right\rfloor$, where T is the "Time before break" in minutes and $P_3 = (P + T) - P * 3$. In the first phase, the character is set to the default size and is reduced to 40% opacity. Each subsequent phase increases the size of the character by 5% and the opacity by 20%, such that by the final phase, the character is completely opaque and 15% larger than default. Once the final phase, P_3, has finished (indicating that the time before break has been exceeded), the character enters the so called "nuisance mode". In this mode, the character moves to random points on the screen unprompted and sends out desktop tray notifications reminding the user to take a break. Whilst the values could be further tweaked, we found during tests that these values anecdotally yielded a nice compromise between noticeability and intrusiveness (where the first 2 phases are not very noticeable and the last 2 more so), which was later further supported via the user survey.



BreakBuddy

Time for a break?
22:28

Figure 5: An example of the desktop tray notification sent out once the time before a break has passed

The last feature to be implemented was the taskbar context menu, which is created as soon as the "Start BreakBuddy" button is clicked. In the taskbar context menu, the user can close the application completely, re-open the character customisation screen (restarting the ongoing timer), and reset the timer. The taskbar icon and context menu were implemented in one part due to necessity (there needed to be a way to close the application whilst active) and the other so that the character customisation screen could be completely closed – ensuring the only window generated and displayed was the one with the character, which was hidden from menus such as the alt + tab one.

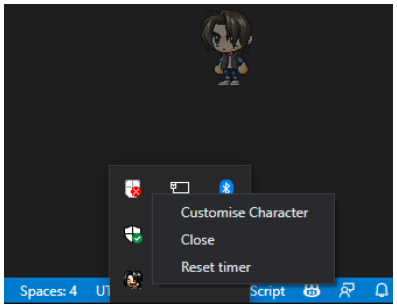


Figure 6: The taskbar context icon and menu on Windows

2 RESULTS

We created a user survey (link in the appendix) with a mixture of Likert-scale (1 – 10) questions and text input that was sent out to all the users after at least 3 days, once they communicated to us that they had installed the application. In total, there were 11 participants that had agreed to both install the application and answer the survey at the end. Of the 11 participants, 8 were male and 3 female, ranging from 21 to 33 years of age (with a mean age of 24.6). The participants comprised of both classmates from a graduate course in Software Engineering and colleagues of the authors.

To help answer **RQ1**, we focus on section 2 of the survey.

First, we can observe that all the users were, to some extent, aware of the character changing over time, as initially hypothesised. 3 of the users didn't find the changes useful, scoring the relevant question (*How useful did you find the changes as an indicator that you should take a break?*) 3, 4, and 5 points across a 1 – 10 Likert-scale. In contrast, the other 8 users collectively rated the feature with a mean of 8.5, showing that there was clearly some amount of people who found it very useful.

9 out of the 11 participants reported taking a break at least one time as a direct result of the program - 4 due to the notifications, 2 because of the character changing state, and 3 because of the notifications in some cases and the character in the other.

Although most participants took at least one break, not all seemed to find either the character changing or the notifications effective. 4 participants voted the relevant effectiveness question (*Overall, how effective did you find the application was in getting you to take a break?)* 2, 4, 4, and 5. In contrast to this, the other 7 users graded it with a mean of 8.7, with one participant commenting that they found it "*Good too* [sic] *have both character and* [desktop] *notifications*".

Overall, the participants didn't find the application too intrusive. The total intrusiveness mean score was 4.9, which increased to 6.55 when they were questioned about the intrusiveness of the most intrusive state. Such results are indicative of reaching a fair middle ground between low and high intrusiveness, which is what we had originally set out to achieve.

Regarding the frustration aspect of the research question, most of the participants did not find the application frustrating. 4 out of the 11 participants felt some degree of frustration, which they largely attributed to being distracted by the character during periods of high focus. One participant commented: "*When the character suddenly appears in the middle of the screen it might be frustrating when I am highly concentrated on something. It changes the focus of attention*". This can be linked to the flexibility afforded by the application, where users can set their own working period but ultimately cannot be sure it will be effective. Such an approach would benefit from a more dynamic state of the art solution, which should make predictions on focus at the individual level and change the intrusiveness levels accordingly.

To help answer **RQ2**, we focus on section 1 of the survey.

With our survey, we found that only one out of the 11 participants didn't customise their character. For the other 10 participants, 4 of them spent less than 1 minute customising it whilst the other 6 spent between 2 and 5 minutes. When inquiring as to the importance of being able to customise the character, we ultimately found differing opinions, suggesting that this was a matter of subjectiveness (at least for our demographic).

To help retrieve additional insights into the matter, we asked a follow-up question ("*How much would it have bothered you if the character was preselected and you therefore couldn't customise it?*"), which netted the close to the same distribution of results as in the previous question.

We then asked the participants whether they had named the character or not - 4 of the participants did not, leaving us with 7 that did. For those who named it, seeing their chosen name appear on the screen when the pop up appeared seemed to positively contribute towards getting them to take the break seriously - "*Felt like a caring, imaginary friend telling me to take a rest*" and "*I was not expecting it so I noticed it more when I saw the name I set so I think because of this It made me want to take more a break*".

Finally, when asked if they had used the application more than once, and the reason for it, three main reasons were common: utility, greater awareness, and fun. Whilst the character customisation was never explicitly stated, we believe that it at least plays some role in the fun aspect, as stated.

3 LIMITATIONS

The application was implemented in a relatively short amount of time with limited peer consultation throughout and was sent unpolished in the interest of collecting feedback over a longer period. Consequently, 3 users mentioned they had encountered bugs, with 1 of the 3 users encountering a bug that managed to impair their experience significantly. Whilst the application was successfully deployed cross-platform, we were unable to perform comprehensive tests on OS X, which was the operating system reporting the impairment bug.

Comprehensive research on break reminder utilities and their applications are lacking, so many of the decisions we made during development were based off our own intuition and personal experiences, rather than documented approaches.

With a low amount and specific demographic of people that were surveyed, the results obtained only contextually answer our research questions. The results serve more as a proof of concept, that there is at least a specific demographic that positively received the application, than anything else. Further testing across a wider range would be needed to validate the results at scale.

4 FUTURE WORK

Section 3 of our survey was structured to obtain feedback from the users which could be used to improve our program in the future.

The main dislikes of the application were centred around bugs and missing functionality, in particular the ability to manually pause the timer. Regarding the bugs, due largely to time limitations, we were unable to extensively test the project, especially with regards to testing on non-Windows operating systems. Fortunately, through the feedback obtained during the survey, we were able to fix quite a few of the critical bugs – though we were not able to hotfix them for the users still using the application.

We then asked the participants about additional features that they would like ("*Are there any feature/s you would like to have added to the app?").* The main features users asked for pertained to customisation and additional timer functionality (pausing the timer). Unfortunately, again due largely to time limitations, the customisation had to be left in a very bare state but is certainly something we could extend should we work on the project again. The pause timer functionality was an oversight we failed to recognise but would certainly implement with more time – the technical nature of the problem meant a quick implementation could not be deployed in time.

Lastly, we asked the participants about changing the existing features ("*Are there any feature/s you would change/remove from the app?*"). Much of the feedback related to the way the user was notified, though one participant provided a very interesting idea about including a soft sound that could be played, instead of popping up a desktop tray notification. Through such an approach, the user would know that the time to take a break was nigh but could still finish their existing task in peace. This is certainly something we would look to incorporate into the application in the future and test.

5 CONCLUSION

Based on the varying results obtained through our survey, we can generally conclude that each user has their own different view for what they regard as more or less intrusive. The survey we deployed reinforces the statements of the two research questions we made at the beginning. Yes, a gradually scaling intrusiveness tool can help a user to take those breaks whilst not being frustrating, but intrusiveness should be implemented to neither distract during periods of high focus nor go totally unperceived. Additionally, the customisable aspect of the character in our approach seems to serve as a nice solution to further emphasise the need to take a break and make the user want to keep doing so, as the users we surveyed perceived it rather as a "fictional friend" than as something non-sentimental. Whilst this study presents its fare share of problems and difficulties, we believe that we are at least able to demonstrate a good proof of concept approach to a previously lesser studied problem, with plenty of room for further improvements.

REFERENCES

[1] Luo, Y., Lee, B., Wohn, D. Y., Rebar, A. L., Conroy, D. E., & Choe, E. K. (2018). Time for break: Understanding information workers' sedentary behavior through a break prompting system. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1-14).

[2] Kühnel, J., Zacher, H., De Bloom, J., & Bledow, R. (2017). Take a break! Benefits of sleep and short breaks for daily work engagement. *European Journal of Work and Organizational Psychology*, *26*(4), 481-491.

[3] Henning, R. A., Jacques, P., Kissel, G. V., Sullivan, A. B., & Alteras-Webb, S. M. (1997). Frequent short rest breaks from computer work: effects on productivity and well-being at two field sites. *Ergonomics*, *40*(1), 78-91.

[4] Rock, D. (2010). Your brain at work: Strategies for overcoming distraction, regaining focus, and working smarter all day long. *Journal of Behavioral Optometry*, *21*(5), 130.

[5] DeskTime (2018). The secret of the 10% most productive people? Breaking! https://desktime.com/blog/17-52-ratio-most-productive-people/

[6] Ariga, A., & Lleras, A. (2011). Brief and rare mental "breaks" keep you focused: Deactivation and reactivation of task goals preempt vigilance decrements. Cognition, 118(3), 439-443.

[7] Hovancik, J. (2021). Strechly. https://hovancik.net/stretchly/

[8] Penners, R., & Caelers, R. (2021). Workrave. https://workrave.org/

[9] Consolvo, S., McDonald, D. W., & Landay, J. A. (2009). Theory-driven design strategies for technologies that support behavior change in everyday life. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 405-414).

[10] Lanier, J. (2011). Awareness. https://iamfutureproof.com/tools/awareness/

[11] The importance of taking breaks. The Wellbeing Thesis. (2020, January 14). Retrieved December 9, 2021, from https://thewellbeingthesis.org.uk/foundations-for-success/importance-of-taking-breaks-and-having-other-interests/

APPENDIX

- GitHub - https://github.com/danielj0nes/BreakBuddy
- Character assets - https://maples.im/
- Survey questions - https://forms.gle/jqg1bpJxh2NuwVxW8
- Video demonstration - https://youtu.be/V0FjnRhxxZo