Computational Tool to Support Communication for People with Extreme Motor Impairment

Guilherme M. A. Ramos
Universidade Federal de São Carlos
Rodovia Washington Luís km 235
13565-905 São Carlos SP - Brazil
+55 16 3351-8234
gmatheus95@outlook.com

Victor O. N. Sales
Universidade Federal de São Carlos
Rodovia Washington Luís km 235
13565-905 São Carlos SP - Brazil
+55 16 3351-8234
victor_otavio205@hotmail.com

Cesar A. C. Teixeira
Universidade Federal de São Carlos
Rodovia Washington Luís km 235
13565-905 São Carlos SP - Brazil
+55 16 3351-8232
cesar@dc.ufscar.br

ABSTRACT

People can lose motor capabilities for a variety of reasons, like accidents and infirmities. In extreme situations, as found in patients diagnosed with ALS (Amyotrophic Lateral Sclerosis), individuals lose full motor capabilities, only being able to move their eyes[1]. As a consequence, communicating, for these individuals, becomes a challenge. The use of eye-tracking devices associated with virtual keyboards can solve this issue, however, the difficulty for choosing the characters could derail this solution. This research had the purpose of studying, conceiving, implementing and testing communication alternatives using eye-tracking, which wouldn't demand effort to choose characters with precision. The solutions were conceived based on patterns of eye movements. During tests, it was noticed that the writing speed had to be improved, and to achieve it, an autocomplete feature has been added to the application. An implementation is currently functional, grouping the best decisions and insights so far, and according to the stablished requirements. The tool, as it is, can be used with a low-precision eye-tracking camera, achieving acceptable levels of writing speed and a high learning curve.

Keywords

Alternative communication; Assistive technology; Eye-tracking; Word prediction.

1. REQUIREMENTS

The project was inspired by an acquaintance diagnosed with ALS, so most of the stablished requirements come from his demands and known communication issues.

The requirements are as follows: The characters must be selected with eye movements only; The selection must not depend on blinks (since on advanced stages of the disease, people may not be able to control blinking);

The solution must be cheaper than the ones already available; The interface must be intuitive and easy to use; The interface must not be tiresome to the eyes (because of extended use); The solution must provide a resting screen in case the user doesn't want to write anymore; The solution must provide ways to play alarms (in case the user needs to call someone).

In: Workshop de Ferramentas e Aplicações (WFA), 15., 2016, Teresina.
Anais do XXII Simpósio Brasileiro de Sistemas Multimídia e Web.
Porto Alegre: Sociedade Brasileira de Computação, 2016. v. 2.
ISBN: 978-85-7669-332-1

2. METHODOLOGY

In the available solutions, the selection of a specific symbol is normally accomplished by a blink, when looking right at it, or through a timer (i.e., after a couple seconds looking at the same symbol, it is selected). Both solutions aren't very interesting because both would demand a precise camera, the first one would demand a blink and the second could bottleneck the writing speed after some experience with the tool. Thus, to fulfill the requirements and work around these issues, nine large regions were idealized to distinguish the direction of the gaze, and a certain combination of movements among these regions would generate a character, instead of a timer or a blink.

The characters are disposed according to the most frequently used characters in Portuguese, and according to the alphabet order too (for fast association). The main idea is as follows (see Figure 1): there's a text box in the middle of the screen, where the inserted characters are shown.

The selection of each character requires a round-trip eye movement, starting and ending in the center (for example, the character 'E' is generated when looking up and returning to the center; the character 'D' is generated looking up, then left, then back to the center, as shown in Figure 1).

Because of this system, not all the characters and symbols would fit to 24 different movements, so a secondary keyboard was designed and would be achievable through a timer (the second keyboard holds the least used characters in Portuguese).

According to the requirements and scratches, and adopting a model of iterative and incremental development, solutions would be designed, implemented and tested, adding further features on demand.

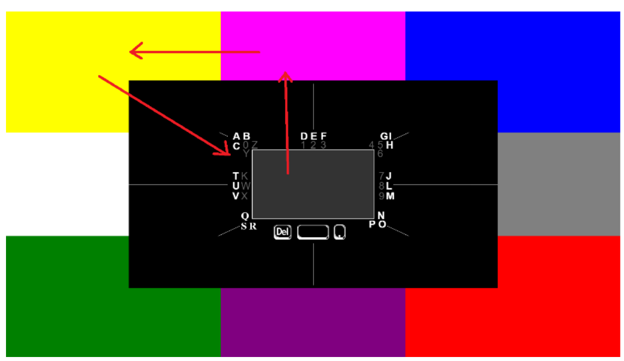

Figure 1. User interface and the movements to select the character 'D'.

3. RESULTS AND DISCUSSIONS

Because of convenience and maintainability, the first versions were developed using Javascript and HTML5. The camera used within the tests was the Eye Tribe Tracker (https://theeyetribe.com/). Considering that one of the goals of this project is to develop a cheap solution, not very dependent on the precision of the camera, this device was chosen for its low price (and therefore, not a very high precision). Thus, it was possible to observe the dependency between the application and precise sensors.

This camera can identify the approximate place on the screen which the person is currently looking at. It's important to note that the solution works with any eye-tracking device. To avoid tiring the user, the interface was changed from the original idea to a cleaner and darker one (see Figure 2), maintaining the small characters around the text box as a reference for the movements that generate them.

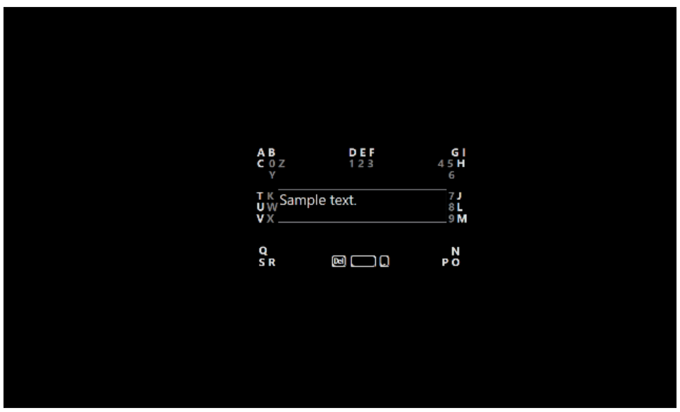

Figure 1. User interface.

Besides this basic structure, some more features have been added in order to attend some of the users' needs and to improve writing efficiency. Blocking screens were added to rest (see Figure 3) and to read full text (see Figure 5), and accessing them is possible through two different complex movements, the first one is accessed looking up, right, down, left and up to the center (see Figure 4), while the second one is accessed looking up, left, down, right and up to the center (see Figure 6).

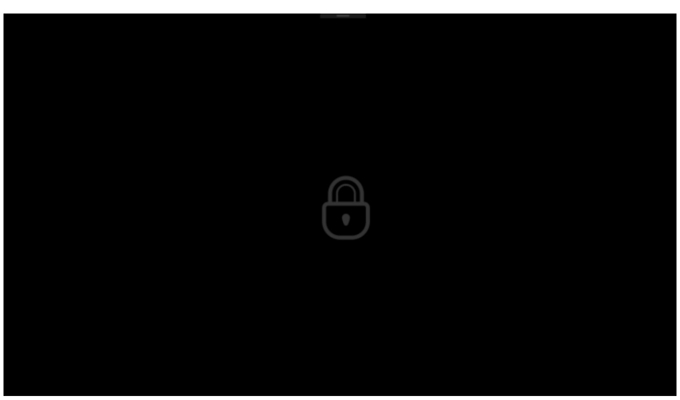

Figure 3. Rest blocking screen

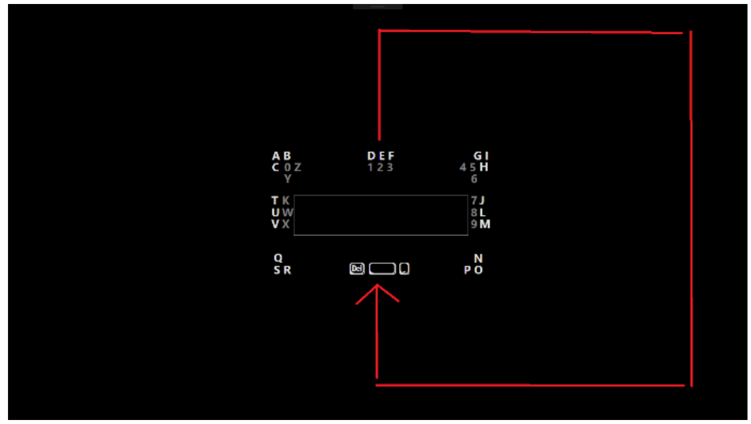

Figure 4. Movement to access the rest blocking screen.

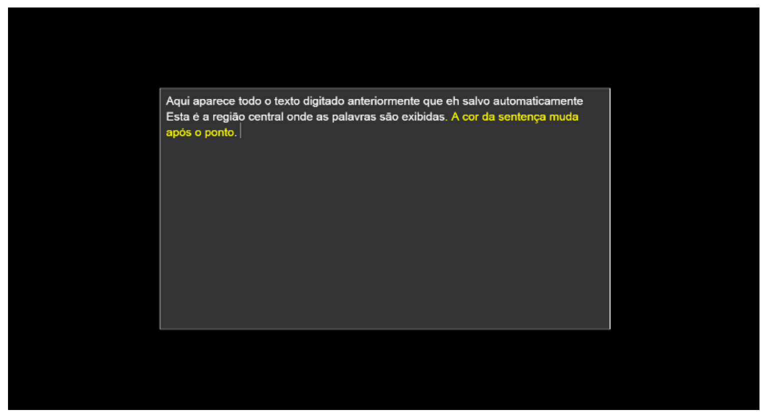

Figure 5. Reading blocking screen.

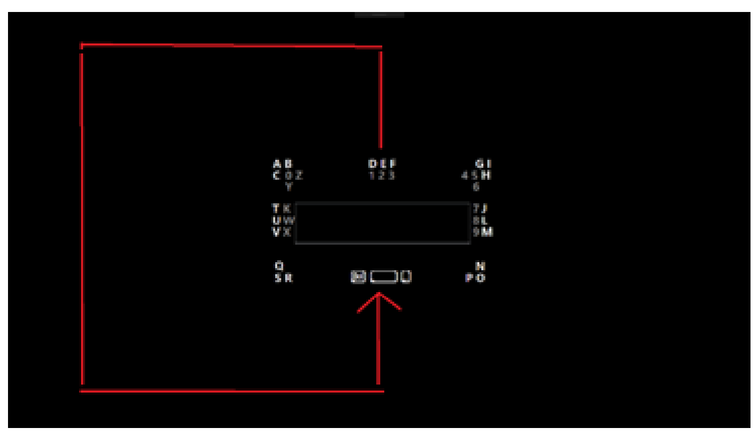

Figure 6. Movement to access the reading blocking screen.

In order to be able to increase the speed of the writing with the user ability, the process of writing is independent of timers, a character is written when the user finishes the motion. Making the communication faster depending on the user. The eye movements must be short and fast, the shortest e fastest movements should be avaible for the most common characters.

The next thing to figure was a way to play an alarm when needed, and since there were not many movements left to explore, the idea of writing something specific to trigger an alarm was the most interesting one.

This combination of characters should not be a word, because sometimes the user might want to write it and not to ring the alarm, and it must be easy to insert, since in a moment of urgency, the alarm must be rung as quickly as possible.

To fulfill these requirements, the combination "jj" was chosen to play a ringbell, while "jjj" was chosen to play a very loud alarm and the solution now completely fulfills its goals and requirements.

Some tests were made and some interesting results could be found: users not familiarized with the tool could select up to 20 characters (roughly four words) per minute, whilst within 20 minutes of continuous use, users could select up to 34 characters (roughly six words) per minute. It's obvious that the writing speed is very low and, although the solution works, improvements had to be made.

After several brainstorms, the solution to improve the writing speed was chosen: Implement a suggestion feature, similar to the ones found on smartphones, word prediction mechanisms (auto complete and suggestions) are used with the objective of increasing the user tipping speed [2], which would suggest words based on the written prefix and would rank the most usual words to be suggested first.

The word prediction mechanisms should anticipate the user actions, making prediction from the user previous data. The past can be recent, instants before the first characters were tipped or the context information were obtained, analyzing word in the current phrase or text [3].

Older information, for example, frequency of use of words tipped by the user or relevant people, are also important. The possibility of correct miss tipping mistakes, by offering the suggestions with one or two miss tipped characters by the user is also a common characteristic in the mechanisms.

An extensive search throughout available solutions exposed that none of them could directly be applied to our system, but adapted and refined beforehand. The chosen solution was to develop an autocomplete feature using Lucene[4], a free and open-source information retrieval software library, because of its capabilities and vast community and documentation.

Because of convenience and to integrate the application with the camera application (designed in .NET Framework), the feature was implemented in C#, as well as the whole solution, previously developed in Javascript and HTML5, was transliterated to the same language.

Once the suggestion solution was built and tested, it had to be integrated to the communication tool. To do so, three things had to be figured out: how many suggestions should be shown, where to show them, and how to accept one.

To solve these issues, the solution was to use a composition of three adjacent corners, showing therefore, up to four suggestions in the center of the screen, so that the user can choose one without looking at any of the corners.

Figure 5 shows the suggestions to the prefix "pa" and figure 6 explains the movement to accept one of them (the others follow a similar movement).

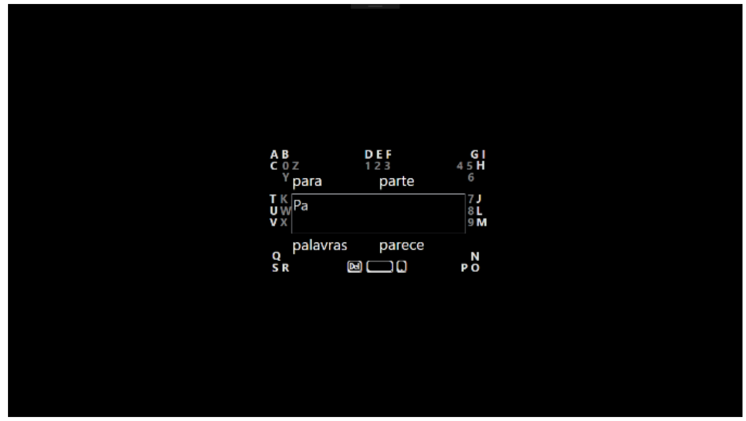

Figure 7. Suggestions to the prefix "pa".

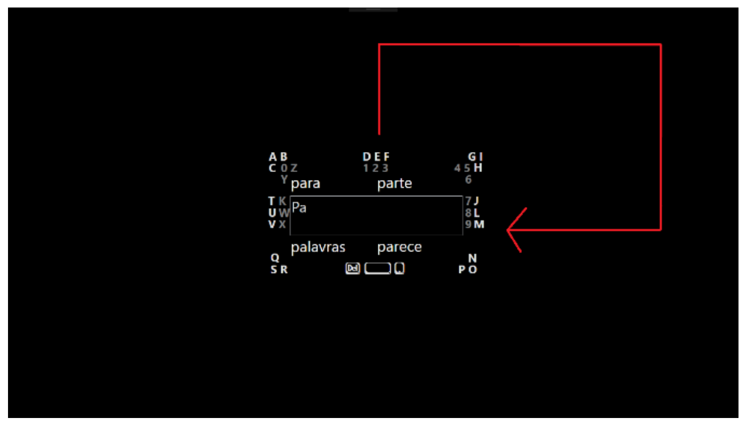

Figure 8. To accept the suggestion "parte", this movement must be executed.

Considering a pre-built base containing ranked words for suggestion (i.e. interesting suggestions will appear as soon as possible), we could achieve interesting results when analyzing a sentence written using the right suggestions as soon as they appeared.

The Figure 9 shows a graphic representing the number of characters per sentence, comparing the written ones and the ones that are actually in the sentence (the more suggestions the user accept, less characters will have to be selected to form a sentence).

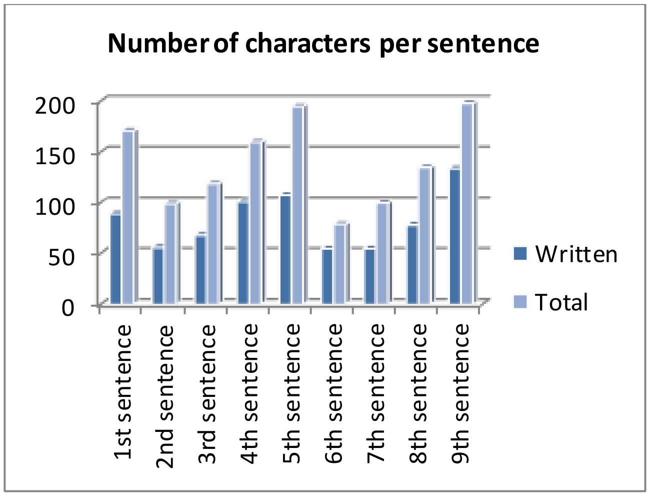

Figure 9. Graphic representing number of characters per sentence

Tests with users after 20 minutes of training with the tool and without the word prediction showed an average of 34 characters per minute, or six words per minute. With the word prediction, this value almost doubled.

On average, 3.05 characters are needed to generate the correct suggestion among the four, which means that, writing approximately 59% of the word is enough to achieve the correct suggestion, increasing therefore, the writing speed.

The tests previously described were accomplished using the following environment: Processor: Intel Core i5 460M (2.53ghz 2 cores, 4 threads); RAM: 8GB DDR3 1066Mhz Dual-channel; Hard drive: 1TB 32MB Cache 5400RPM SataII; Graphics Card: AMD Radeon HD 6470M 512MB GDDR3; OS: Windows 10 Pro; Debugging Tool: Microsoft Visual Studio 2013; Eye-tracking device: The Eye Tribe Tracker.

4. CONCLUSION

Taking into account the current results, the initial objective of building a tool to help the communication of people with severe motor limitations, using "eye tracking" devices with low precision was successful, and its results are quite satisfactory.

Currently the tool is being tested in order to analyze how different people adapt to the tool, the learning curve, the medium speed of writing, and gather opinions from the subjects.

All of this data being collected will be used in order to further improve the tool.

Looking in the incremental side of the model, new functionalities can be added to the tool in the future, with the objective of either improving the speed or the accessibility of the tool, or adapting it to help people with similar needs.

An example of a recently added functionality is the implementation of the suggestion solution.

The prediction tool increases considerably the writing speed of the user, and the extensive use of the application will improve the accuracy of the suggestions, based on user previous data, adapting itself for the user vocabulary.

The link of the repository with the tool is available bellow: https://drive.google.com/file/d/0B2STwzp26nphenRQWEgtVkhPZU0/view

The tool can be tested by using the mouse, emulating the eye movements of the user. When using eye tracking devices, the devices are used to move the mouse cursor of the PC.

5. ACKNOLEDGEMENTS

We would like to thank FAPESP (processes 015/13931-8 and 08/57705-8) and CNPq (processes 573972/2008-7 and 312692/2013-7) for the indispensable endorsement in this research.

6. REFERENCES

[1] Oliveira, A. S. B, Pereira - Amyotrophic lateral sclerosis (ALS): three letters that change the people's life. For ever. Arquivos R. D. B. Neuro-Psiquiatria Print version ISSN 0004-282X Arq. Neuro-Psiquiatr. vol.67 no.3a São Paulo Sept. 2009 http://dx.doi.org/10.1590/S0004-282X2009000400040, último acesso em 07 de agosto de 2014.

[2] Tam, C. ; Wells, D. - Evaluating the Benefits of Displaying Word Prediction Lists on a Personal Digital Assistant at the Keyboard Level. Assistive Technology, 21, 2009, p. 105-114.

[3] Witten, I. H. ; Darragh, J. J. - The reactive keyboard. Cambridge, UK: Cambridge University Press, ISBN 0-521-40375-8, 1992, p. 43–44

[4] McCandless, M. ; Hatcher, E. ; Gospodnetic, O. - Lucene in Action: Covers Apache Lucene 3.0. Second Edition. Connecticut: Manning Publications Co., 2010, 475 p. ISBN: 1933988177 9781933988177