**Python-based virtual keyboard descriptive language interpreter**

R. C. V. Dias1, J. R. Oliveira1, M. B. Pelosi2

1DCA/FEEC, UNICAMP, 2FCM, UFRJ

**Introduction:** This study proposes the use of brain signals for alternative communication by interfacing it to virtual keyboards, making possible not only to express words, phrases and texts but also enabling transparent end integration with other daily-use computer programs. In alternative communication context virtual keyboards make a set of options available to the user, normally related to communication context (letter, words, and sentences) but not limited to, and these options can be accessed in many ways, and with different types of input. In most restricted scenario, a virtual keyboard with N options can be used and controlled only with a binary signal input, using scan techniques. The objective of this study is to implement a Python[2] based interpreter that can be used to instantiate different virtual keyboards and at the same time has its inputs connected to the output of processed brain signals, enabling effective end-user communication.

**Materials and Methods:** As shown in the work “Descritive language for virtual keyboards with focus on augmentative and alternative communication”[1], virtual keyboards can be described and defined using data structures based on ontologies. That work also defines a description language supporting main virtual keyboards features, which will be used in this work as the format for defining and inputting virtual keyboards into the interpreter. Also, this interpreter will have an interface for inputting the brain output data, enabling the control of this virtual keyboard. Tests will be made in simulated environment, since the interpreter expects processed brain signals outputs, and with some basic virtual keyboard options.

**Results:** The interpreter was developed in Python[2] and expects input in XML format defined by the virtual keyboard descriptive language1. Two basic keyboards were proposed:

Basic Words

Options (buttons): “Yes”, “No”, “Maybe”, “Stop”, “Pain”

Free Writing Keyboard

Options (buttons): all letters of the alphabet (“A”, “B”, “C”, “D”, “E”, “F”, etc)

The interpreter implemented two options for input:

* Binary Signal: the interpreter makes a time-based scan of all options, so the signal must be inputted when the desired option is on focus.
* Direct Mapping: each input is mapped directly to one option of the keyboard.

**Discussion:** The results have shown that virtual keyboards can be used for effective end-user communication from brain signal when these signals are already processed in order to give a more clean output. The developed interpreter based on the virtual keyboard descriptive language makes possible to adapt it to specific needs without any change in computer code, since only the high-level description of the keyboard must be changed. Also, as this description of the keyboard is very close to the characteristics of the keyboard itself, it’s easy to researches make changes to fit it to their needs.

**Conclusion:** The present work confirmed that virtual keyboards can be used to write and express feelings, words and sentences with minimal – for example only one binary signal - from brain. But virtual keyboards can also be used to interact with other programs or systems, as its options can be structured as control panels or even “cockpits”. So, this interpreter can be evoluted in order to make integration with other systems possible.

**References:** [1] DIAS RCV, “Descritive language for virtual keyboards with focus on augmentative and alternative communication”, MA thesis, 2016; [2] Python www.python.org