Journal of Neuroscience Methods 222 (2014) 250–259

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Basic Neuroscience

The Psychology Experiment Building Language (PEBL) and PEBL Test Battery

Shane T. Mueller[a,*], Brian J. Piper[b]

[a] Department of Cognitive and Learning Sciences, Michigan Technological University, United States
[b] Department of Basic Pharmaceutical Sciences, Husson University, United States

ARTICLE INFO

Article history:
Received 8 August 2013
Received in revised form 24 October 2013
Accepted 30 October 2013

Keywords:
Open science
Iowa Gambling Task
Wisconsin Card Sorting Test
Continuous Performance Test
Trail-making Test
Executive function
Mental rotation
Motor learning

ABSTRACT

Background: We briefly describe the Psychology Experiment Building Language (PEBL), an open source software system for designing and running psychological experiments.
New method: We describe the PEBL Test Battery, a set of approximately 70 behavioral tests which can be freely used, shared, and modified. Included is a comprehensive set of past research upon which tests in the battery are based.
Results: We report the results of benchmark tests that establish the timing precision of PEBL.
Comparison with existing method: We consider alternatives to the PEBL system and battery tests.
Conclusions: We conclude with a discussion of the ethical factors involved in the open source testing movement.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The Psychology Experiment Building Language (PEBL) is a free, open-source software system that allows researchers and clinicians to design, run, and share behavioral tests. At its core, PEBL is a programming language and interpreter/compiler designed to make experiment writing easy. It is cross-platform, written in C++, and relies on a Flex/Bison parser to interpret programming code that controls stimulus presentation, response collection, and data recording. PEBL is designed to be an open system, and is licensed under the GNU Public License 2.0. This allows users to freely install the software on as many computers as they wish, to share their experiments with others without worrying about licenses, to distribute working experiments to other researchers or remote subjects without requiring special hardware locks, and to examine and improve the system itself when it does not suit one's needs.

2. History of the Psychology Experiment Building Language

Development and design of PEBL began in 2002. An initial limited release of PEBL 0.1 was made in 2003, with the first public release to the Sourceforge.net servers in January 2004. The initial motivation for its development and design was a dissatisfaction with the current experiment-running systems available. At the time, there were no robust cross-platform Free systems available, which meant that major vendors would focus on one platform (e.g., Psyscope for Macintosh computers, Superlab or E-Prime for Windows PCs, and no packages available for Linux), or researchers who cared about cross-platform testing would develop one-off cross-platform applications using Java or web interfaces. The original design of PEBL abstracted aspects of stimuli, response-collection, and data structures so that it could be implemented on multiple distinct platforms. However, the first implementation platform was done via the Simple DirectMedia Layer (http://libsdl.org) gaming library, which itself is a cross-platform library. Because of this, versions of PEBL are available on MS Windows, OSX, and Linux operating systems.

Due to its limited capabilities, initial use and adoption of PEBL was fairly modest. For the first year, roughly 250 users downloaded PEBL, and five emails were exchanged on the support email list. During 2005, PEBL's activity improved somewhat, nearly doubling to 450 downloads and 10 emails exchanged. Over this period, five versions of PEBL were released (0.1–0.5). Starting in 2006, we began to release an accompanying test battery, initially consisting of eight commonly-used laboratory tests, which is primarily responsible for the first increase in downloads, and an initial set of publications starting in 2008. Since then, the number of downloads have increased to a stable level of between 1000 and 2000 downloads

* Corresponding author at: Department of Cognitive and Learning Sciences, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931, United States. Tel.: +1 906 487 1159.
E-mail address: shanem@mtu.edu (S.T. Mueller).

0165-0270/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jneumeth.2013.10.024

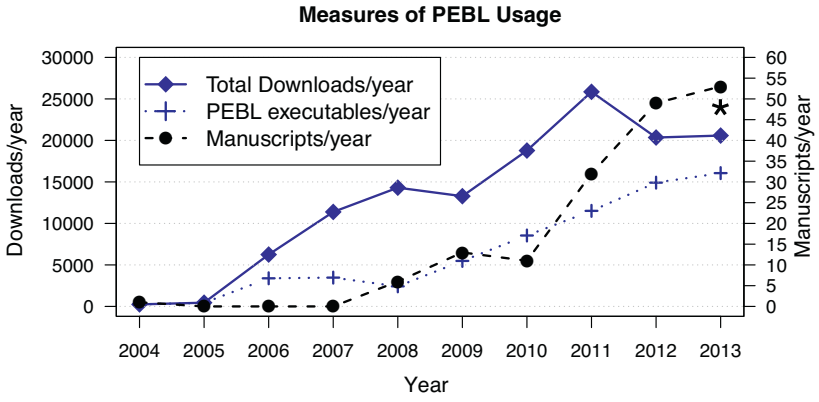

Fig. 1. Three measures of the usage and adoption of PEBL over time: the number of downloads (total and PEBL installer only) recorded via sourceforge.net, and the number of published manuscripts (papers, conference proceedings, theses, etc.). *2013 figures are partial values through November 30, 2013.

per month (see Fig. 1, which also shows the number of published manuscripts that have used or cited PEBL over time[1].) For 2012, the last year for which we have complete records, there were 50 publications citing PEBL, and 20,300 downloads.

The true usage of PEBL is likely to be much broader than may be reflected by citation counts for several reasons. First, PEBL gets used frequently in research methods and other undergraduate courses, resulting in studies that cannot typically be published. Second, PEBL has been used in many academic theses (including bachelor's-level honors theses up to Ph.D theses), and for research studies that are only presented at conferences, and these are not systematically indexed or publicized. Third, because it is free, PEBL gets used internationally, where research studies may be more likely to be presented only at regional conferences or in language-specific journals that are not indexed. Finally, even the top journals systematically fail to require authors to cite work related to the software they use to conduct the study. Many publications that use PEBL have merely referenced its website in a footnote or parenthetical comment. These footnoted references do not appear in standard citation indexes, and it is likely that there are published articles that have used PEBL but made no reference or mention to the system by name.

3. Features of PEBL

The features of PEBL are far too numerous to describe in detail here. We provide a 200-page reference manual that can be freely downloaded or purchased on-line which details the programming system and its functions (Mueller, 2012a,b). As a basic overview, PEBL 0.13 supports a number of stimulus types, including images (in a variety of image formats), text rendered in TrueType fonts using both single-line stimuli and multi-line text objects; many rendered shape primitives (lines, circles, rectangles, etc.); audio recordings, video recordings, and simple generated sounds. For response collection, PEBL supports keyboard, mouse, gaming device input, communication via TCP/IP, serial, and parallel port, and a software audio voice key. In addition, timing of stimuli and responses can be recorded and controlled with a precision dictated primarily by the limits of the hardware and operating system used. As we will show later in the manuscript, internal event timing can be scheduled/performed with 1-ms precision, keyboard responses can be recorded with roughly 5-ms precision, and stimuli can be displayed in increments of the video update frequency.

PEBL provides a library of functions for general computing as well as ones devoted to the design of experiments. These include a wide selection of functions for randomization, sampling, and counterbalancing; data handling and statistics; standard experimental idioms (e.g., built-in functions for messages, multiple-choice questions, many types of commonly-used visual stimuli), and both restricted-set (e.g., press one of several keyboard buttons) and multidimensional response collection (e.g., free-form typed input).

PEBL experiments are typically run via a software launcher that allows users to select aspects of how the test is conducted (screen resolution, participant code, etc.) and also allows "experiment chains"; tests that can be run in sequence. The launcher is itself written in PEBL, and so achieves cross-platform execution on any platform PEBL is available on. A screenshot of the PEBL launcher is shown in Fig. 2.

In comparison to other similar systems, PEBL has both advantages and weaknesses. For example, a number of similar systems employ a special-purpose GUI that can be used for visual programming to create simple experiments using drag-and-drop metaphors, including OpenSesame (Mathôt et al., 2012); PsyScopeX (http://psy.cns.sissa.it/) E-Prime (http://pstnet.com), Presentation (http://neurobs.com), and others. In contrast, PEBL experiments are implemented via a flexible, full-featured programming language, which limits is accessibility to some users, although enabling more elaborate experimental designs. Detailed instructions and tutorials for programming in PEBL are available elsewhere, but we have included the code for a simple choice-response experiment as an Appendix. In addition, PEBL is designed to avoid many object-oriented and event-focused programming metaphors found in full-featured GUI programming toolkits which are sometimes confusing to novices.

For use as a scientific tool, the open-source nature of PEBL has advantages over many closed-source solutions. These advantages begin with the ability to inspect, alter, and redistribute the source code, so that experimenters can verify and change aspects of an experiment, and an experimentation tool can live on even if the original developer abandons it. Another advantage is that the development model enables using a large number of existing open source libraries and source code developed by others, reducing the complexity of PEBL. Finally, the open-source nature means that executables can be freely distributed, allowing experimenters great flexibility in how they conduct their tests. Nevertheless, it should be recognized that commercial software generates revenues that can help to support norming studies, continued software development, documentation, bug fixing, and can provide dedicated support to customers.

[1] A complete listing is available at http://pebl.sourceforge.net/wiki/index.php/Publications_citing_PEBL.

Download English Version:

https://daneshyari.com/en/article/6268928

Download Persian Version:

https://daneshyari.com/article/6268928

Daneshyari.com