



The Psych-Toolbox-Wrapper

A free and open source Toolbox for the control of behavioral experiments

T. Otto, B. Suchan

299.19/DDD43

tobias.otto@ruhr-uni-bochum.de
www.ruhr-uni-bochum.de/neuropsy

Institute of Cognitive Neuroscience, Ruhr University Bochum, Germany

INTRODUCTION

The Psych-Toolbox-Wrapper is a free and open source Toolbox for the control of behavioral experiments. The major aim of this project was to provide programming novices with a set of easy-to-use functions to control behavioral experiments without limiting the power and flexibility of the underlying toolbox and the programming language.

Since the Psychophysics Toolbox is a powerful software for almost all experiments in neuroscience its concept and programming is hard to implement for beginners. The Psych-Toolbox-Wrapper helps new users to focus on the experiment and the

underlying question without dealing with the technical details of the Psychophysics Toolbox. The modular design of the Psych-Toolbox-Wrapper allows to create simple behavioral experiments on the one hand and advanced hardware dependent experiments like EEG or fMRI on the other hand.

Nearly every other hardware can be implemented as long as it supports MATLAB or standardized and documented communications. Since the Psych-Toolbox-Wrapper acts as a link to the Psychophysics Toolbox, the experiments run with MATLAB and Octave on Mac OSX, Linux and Windows.

IMPLEMENTATION

A few powerful functions form the core of the Psych-Toolbox-Wrapper.

The names of the functions are self-describing, well documented and every function has examples for its most important arguments. The function call itself is standardized throughout the Psych-Toolbox-Wrapper and follows the concept of a self-explanatory string followed by the argument itself.

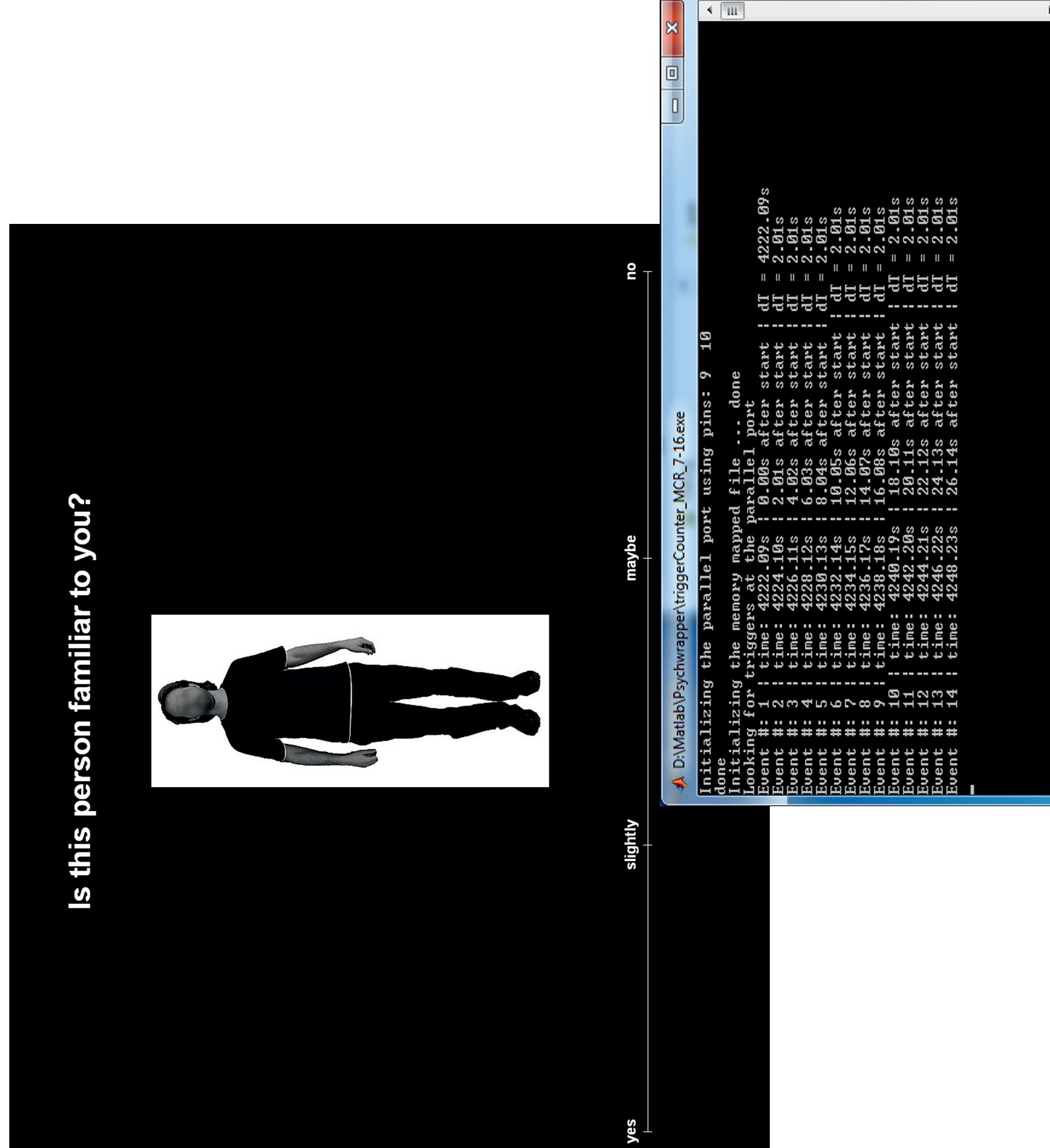
EXAMPLES

```
%> Rating 128 Images with fMRI implementation
start;
initNodes;
mmap = memmapfile('mapfile.dat', 'format', 'double', 'Writable', true);
% Load stimuli
for i=1:128
    stim1 = loadImage(['1\Stimuli\stimulus', num2str(i), '.jpg']);
    % The images are named to stim1, stim2, ...
end

%% Show instructions
showInstructions('instructions.txt');

%% Start rating
fixationCross(1);
fixationCross(1);
showStimuli(stim1, 'dontflip');
showText('Is this person familiar to you?', 'y', 10, 'dontflip');
% Present stimulus
trigger(i);
stimString = mmap.data; % Number of valid scans and time when the last valid scan occurred
% Present scale and wait for a response
out1 = plotScale(5, ['yes', 'slightly', 'no'], 'y', 80);
% Save user response and corresponding fMRI triggers
trigger(i).ansString = mmap.data; % Number of valid scans and time when the last valid scan occurred
trigger(i).ansString = mmap.data; % Number of valid scans and time when the last valid scan occurred
end

%% Save results
% The results file is automatically named to sub0315-2012-10-14-15-09.mat
saveFile(out, trigger, 'subject', 'sub0315');
```

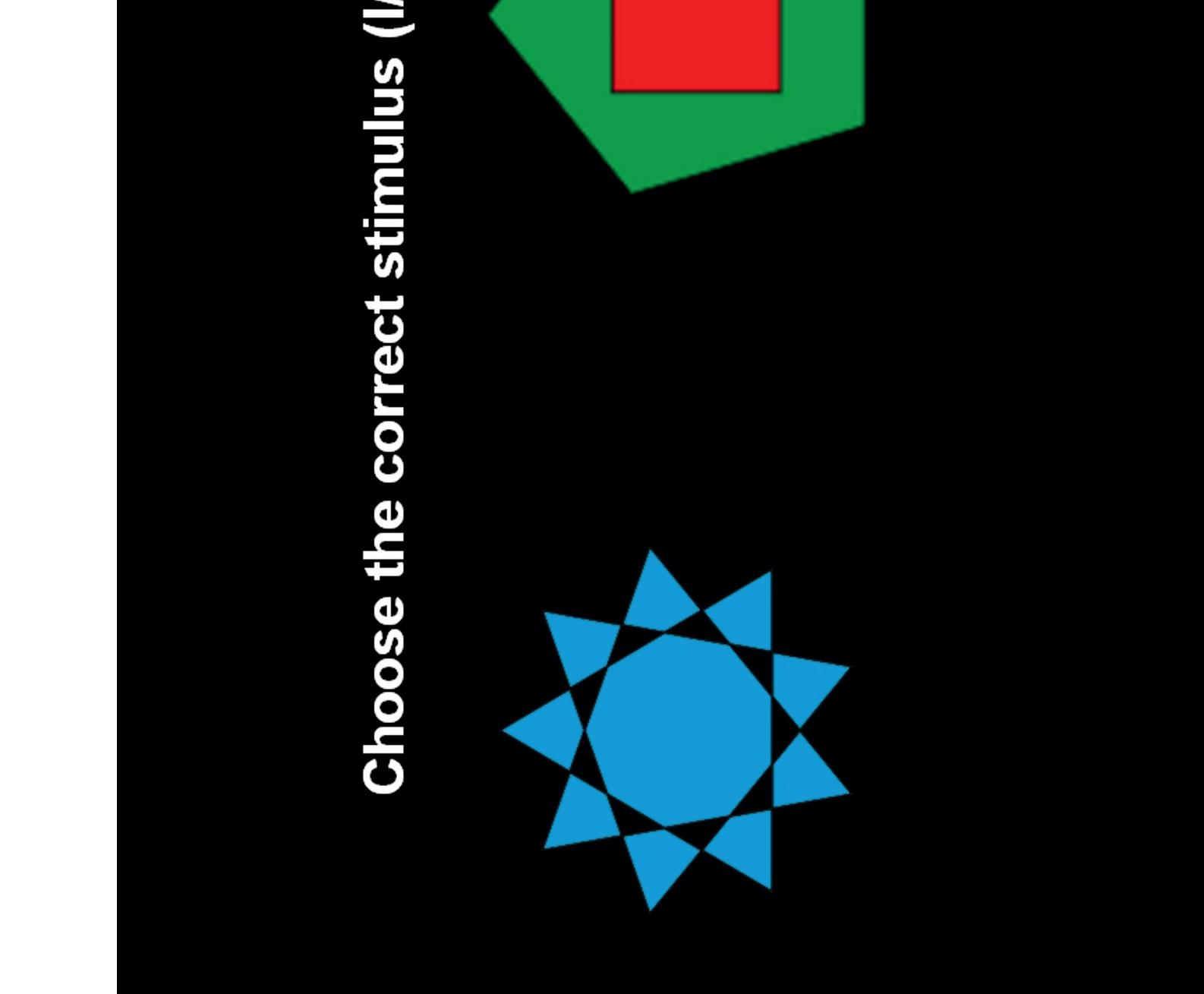


```
%> Initialize Psychophysics Toolbox Wrapper
start;
initWindow;

%% Load stimuli
% Only one line per image is needed to load a picture into the workspace.
stim1 = loadImage('star.jpg');
stim2 = loadImage('poligon.jpg');

%% Present stimuli
% The position of the picture is defined with the optional arguments 'x' and 'y' and the relative position in percent
showStimuli(stim1, 'x', 30, 'y', 50, 'dontflip'); % left upper corner
showText('Choose the correct stimulus!', 'size', 30, 'color', [255 0 0]);
% right upper corner
showText('That was too slow', 'size', 30);

%% Wait for user input and give feedback
% Get answer waits for 5 seconds and then continues. Allows responses are 1(=76) and r(=82), where 1 is correct and r is wrong
out = getAnswers(5, 'goodKey', 76, 'badKey', 82);
if (out.answer == 1)
    showText('Good job! That was correct', 'size', 30, 'color', [100 180 50]); % Display text in green
else if (out.answer == -1)
    showText('Sorry!', 'size', 30, 'color', [255 0 0]); % Display text in red
else
    showText('That was too slow', 'size', 30);
end
```



Different examples of randomizations included in the Toolbox

```
>> out = nBackBuffer(20, 3, 5)                                >> out = randomOrder(3,20, 'ratio', [1.2 .2 .6])      >> out = randomOrder(5,1)
out = [3 2 6 8 1 3 9 10 3 5 7 4 5 7 4 8 5 7]                out = 3 4 5 2 3 4 5 2 1
buffer: [0 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0]                      out = 3 3 3 1 3 2 3 3 2 1 3 1 1 3 3
stimulus: [0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0]                   out = 3 4 5 2 3 4 5 2 1
hit: [0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0]
```

IMPLEMENTATION

Two functions providing the most used randomizations are already included in the Psych-Toolbox-Wrapper.

Two functions providing the most used randomizations are already included in the Psych-Toolbox-Wrapper. The function "randomOrder" allows multiple optional arguments to provide maximum flexibility to the programmer: e.g. normal distributed random numbers, random numbers with a defined ratio or simply a defined count of random numbers with given values. The second function "nBackBuffer" provides the experimenter with a complete set of randomized n-back numbers for one experiment.

INITIALIZATION

Most of the functions in the toolbox require an initialization. This is performed by calling "start" followed by "initWindow" at the very beginning of each paradigm. The first call initializes the pseudorandom numbers generator of MATLAB, resets timers and adds the necessary paths depending on your software version and operating system.

"InitWindow" opens the window for user interaction and stimulus presentation. The Psych-Toolbox-Wrapper supports multiple monitor setups and is capable of creating several stimulus presentation windows in a single monitor setup. Supporting multiple monitors is a helpful feature for more complex applications to inform the experimenter during an experiment about first results in real time.

LOADING AND PRESENTING STIMULI

Presenting stimuli is one of the most used features in behavioral experiments. Therefore "showStimuli" contains a lot of optional arguments: the size and position of each stimulus is defined via arguments that work with every screen size, because all dimensions are specified in relative values and not in absolute pixels. When loading the images with "loadImage" not only the image itself is saved, but also more properties such as the path and filename, the size and much more.

CONCLUSION

The toolbox has been used in our lab for three years in about 30 experiments and has proven reliable. We hope for rich participation of the scientific community in the project. By hosting the project on an open source platform, we wish to extend the scope of the project and give rise to a reliable and free set of tools that are independent of



software-licenses or the doings of a specific company. With this software toolbox no commercial software is needed to design experiments and present visual or auditory stimuli.

Download the Psych-Toolbox-Wrapper at
<http://psychwrapper.sourceforge.net>



www.ruhr-uni-bochum.de/neuropsy | psychwrapper.sourceforge.net