GUI FOR TRIAL-SET-UP (PYTHON)

User Manual

June 2020

Kerstin Hochegger
kerstin.hochegger@student.tugraz.at

Inhalt

General Information

This document is aimed to assist using the Python GUI for Trial-Set-Up.

The code provided was written in Python 3.7, run and tested in the development environment Spyder 4.1.2.

Python comes with some pre-installed packages such as numpy, time, os, (...). However, not all packages used in this program may be included and have to be installed.

Required Packages:

- tkinter (as tk)
- numpy (as np)
- os
- time
- pickle
- pylsl
- logging
- psychopy

Package Parts

- tkinter: filedialog, X, LEFT
- psychopy: visual, core, event, monitors
- pylsl: StreamInfo, StreamOutlet
- psychtoolbox: PsychHID

The Psychtoolbox is not yet fully available in Python and therefore its functions were replaced by Psychopy. The PsychHID library oriniating from Psychtoolbox is already ported to Python and handles the users' keyboard input. The installation of this part of Psychtoolbox is necessary. The supporting code package can be found at:

https://pypi.org/project/psychtoolbox/

Package Installation

Installation of packages in Anaconda:

- Open Anaconda Prompt
- Insert command: pip install [name of package] for example pip install psychtoolbox
- Run

Files and Functions

The program is split into 10 files, each containing functions for a specific task within the program. The main functions of each file are shortly explained, details about the helper functions can be found within the source code. The two main files are GUI_Functions.py and mainLoop.py.

GUI_Functions.py

All settings for the GUI as well as its source code can be found in the file GUI_Functions.py. Drop Down fields can be extended by adding more options to the respective list (e.g. ParadigmList for paradigms or the formatList for other file extensions).

mainLoop.py

This is the basis of the program. Before the main loop is run, functions from several files (stated below) are imported. Keyboard, figure and timing initialization is done to then execute the main loop (trials).

diary

diary

Functions to write a log-file (or diary). Output originally printed in console is also written into the log-file to access all trial-related information later.

closeHandlersDiary

All handlers for logging are closed.

kb_Functions

kb_init

Creates and starts keyboard queue and initializes keyboard-related variables.

kb_read

Keyboard queue is checked for user input when a specific condition is met (timing).

kb_close

Keyboard queue is released.

lsl_Functions

lsl_init

Creation of stream outlet with stream info object.

lsl_close

Has no function yet.

timing_Functions

timing_init

Uses helper functions for determination of Run Type. Creation of randomized trial sequence and calculation of trial timings.

timing_check

When timing condition is met a new state is set.

fig_init

fig_init

Helper functions build paths to stimulation files, set-up screen and load stimuli. Draws “Start” message to screen.

fig_close

Closes window.

fig_update

fig_update

Helper functions update window when a certain condition is met (e.g. set a screen black, draw cross).

saveAndLoadFiles

loadVariables

Function to load pkl-file. An example of using the load-Function is found on the bottom of the file mainLoop.py (currently commented).

saveVariables

Creates a pkl-file with all variables for the current trial.

GUI

The GUI_Functions.py file contains all functions for the GUI-window. Trial Set-Ups can either be customized or executed with standard parameters.

The run command in the file will open the GUI-window (see Figure 1) and shows all variables to be set.

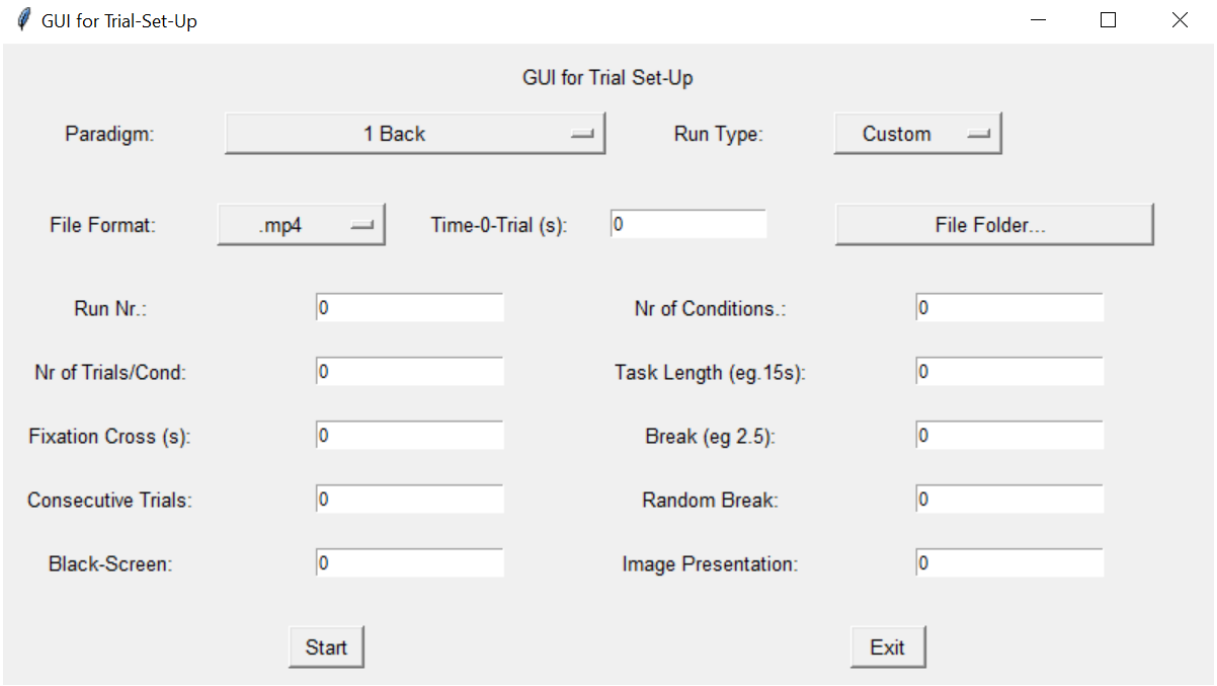

Figure 1 GUI-Window

Parameters

Mandatory Parameters

Note: Mandatory variables must be set regardless of the chosen run type.

File Format

Options: png and jpg (image formats) and mp4 (video format).

In case the required extension is missing, another format can be added. Please extend the files:

- GUI_Functions.py: formatList (add your extension to the list)
- fig_init.py: Add your extension to line 79 for a video format or line 82 for an image format.
- fig_update.py: Add your extension in case of a video format to line 66.

File Folder ...

Opens a dialogue box to search directories. The folder containing stimulus for trials (images or videos) must be chosen.

Run Nr.

Run Nr. is the number of the ongoing trial and is also part of the file names (variable file and log/diary file).

In case the Run Nr. already exists, the trial will be stopped before any existing data is overwritten.

Run Type

"Custom" is auto set and requires several parameters. Standard Run Types can be chosen. This parameter is explained in detail later.

Paradigm

Options: 1 Back, 2 Back, N-Back and Oddball.

If none of these fulfills your requirements you can add the desired paradigm name to the original code (GUI_Functions.py, line 10).

Run Type

There are 4 different run types available:

Run Type 1

nDiffCond = 6
nTrialsPerCond = 1
nMaxConsecTrials = 1
t_cross_mi = 15
t_break_min = 2.5
t_mean_rand_break = 0.5
t_zero_trial = 1.5
t_cross_ref = 2
t_instr_mi = 0

Run Type 1 is set-up for 6 different conditions (e.g. 6 videos or 6 images). Trials per condition is set to 1, so that each condition is presented once with one consecutive trial max. The task length is 15 seconds. The break after the task is 2.5s, the random break 0.5s, the cross-presentation period is 2s. Time period before the first trial is 1.5s. The duration of showing the black screen is linked to the variable t_instr_mi but is usually 0s.

Run Type 2

nDiffCond = 6
nTrialsPerCond = 30
nMaxConsecTrials = 2
t_cross_mi = 0.080 # image presentation
t_break_min = 1.5 # break after task
t_mean_rand_break = 0
t_zero_trial = 1.5
t_cross_ref = 0.5 # reference cross/dot presentation
t_instr_mi = 0

Run Type 3

nDiffCond = 10

nTrialsPerCond = 10
nMaxConsecTrials = 2
t_cross_mi = 0.080 # image presentation
t_break_min = 1.5 # break after task
t_mean_rand_break = 0
t_zero_trial = 1.5
t_cross_ref = 0.5 # reference cross/dot presentation
t_instr_mi = 0

Run Type Custom

All parameters can be customized.

Time-0-Trial:

Period before the very first trial is started.

Nr of Conditions

Number of images/videos presented for one experiment.

Nr of Trials/Cond

Number of times a condition is shown (trials).

Task Length

Length of the task.

Fixation Cross

Presentation time of fixation cross in seconds.

Break

Break after task.

Consecutive Trials

How many trials after each other are required?

Random Break

Random break.

Black Screen

Usually set to 0. (Presentation of black screen in seconds).

Image Presentation

Must be set for Run Type Custom if the stimulus is an image and should be shown for a given amount of seconds.

Start

Start executes the trials. In case any parameters are missing, a note occurs.

Exit

Closes GUI.

Determination of certain parameters

Task Length

This parameter should be the sum of the stimulus duration (length of the video or Image Presentation period) and the task time (e.g. Moto Imagery).

Example: Standard Run Type 1:

nDiffCond = 6
nTrialsPerCond = 1
nMaxConsecTrials = 1
t_cross_mi = 15
t_break_min = 2.5
t_mean_rand_break = 0.5
t_zero_trial = 1.5
t_cross_ref = 2
t_instr_mi = 0

The overall task length is 15 s. Each video is approximately 4 to 5 seconds long and the resulting Moto Imagery period is approx. 10 seconds.

Note: The Task Length must be set according the duration of the stimulus and the desired time for the task itself (e.g. Moto Imagery). A short Task Length leads to a full video/image presentation but leaves no time for set presentation of the cross as well as Moto Imagery. The cross and "Imagine" window appear but quickly disappear.

Difference between video and image presentation

A video file is played entirely when called, there is no need for a specific parameter stating the duration of a video.

For testing purposes videos with 4 to 5 seconds each originating from the Point Light Walker Experiment were used to test the code and its ability to meet the timing conditions.

A different approach for image representation is required as an image itself has no duration, is flipped onto the current window but vanishes immediately.

The parameter Image Presentation must > 0 to determine the duration of image presentation.

Messages and Notifications

Several messages and notifications can and will occur during program execution in order to facilitate the workflow.

Message 1

Invalid Input: Directory or Trial-Settings

Appears if wrong or no file folder is set. The folder must contain files with the extension set in the GUI.

Message 2

Set: Nr of Conditions, Nr of Trials/Cond, Task Length, Consecutive Trials

Appears if required settings are left with initial value ('0').

Message 3

!!! Error filename already existing. Stopping!!!

*** finished ***

File name construction: SUPER_[Run Type]_run_[Run Nr]

In case the file name already exists, the process is stopped and exited before any data may be overwritten. The Run Nr is important to distinguish files and must be set correctly.

Message 4

Error

An error within the loop was encountered, the program stops.

Message 5

*** Main loop finished. Exit screen and save data with ESCAPE ***

The loop successfully terminated and the program can be closed by pressing "esc".

Message 6

ESCAPE and Exit GUI

The esc-Button will close the GUI and end the program.

Start Program:

Correct parameter settings will allow the GUI to start the trials, however the user has to manually start the main loop by pressing the 'Enter'-key as soon as the subject is ready and the text "The measurement is about to start ... " appears.

Trial Set-Up

- Black Screen
- White Cross
- Stimulus Presentation (Video or Image)
- Task (e.g. 'Imagine')

Keyboard Input

The program allows user feedback during trials and the following keys are recognized:

- 'Enter': Starts the trial (main loop).
- 'esc': Stops the trial and exits GUI.
- '1': Like - the user likes the stimulus, that is currently presented.
- '2': Dislike - the user dislikes the stimulus, that is currently presented.

Program Termination

The 'esc'- keypress stops the trial and main loop. It automatically closes all outlets and exits GUI. The data received until 'esc'- keypress is stored in the log- and variable files.

'esc'-keypress:

*** Main loop finished. Exit screen and save data with ESCAPE ***

- Closing figure.
- Closing LSL stream.
- Closing keyboard queue.
- Saving data.
- Exit GUI

Successful termination

*** Main loop finished. Exit screen and save data with ESCAPE ***

ESCAPE and Exit GUI

User Input: 'esc'

- Closing figure.
- Closing LSL stream.
- Closing keyboard queue.
- Saving data.
- Exit GUI

After successful termination of all trials, the user is asked to manually exit the program by pressing 'esc'. Outlets are closed and data is stored in the log- and variable files.

Note: The user is obliged to only press the 4 keys stated above. The behaviour of other keys is not explicitly handled in the program and may lead to an error or a continuous loop.

Error Handling

The program may be computationally expensive in some cases and lead to unexpected behavior. This can be caused by background activities or undefined user inputs.

In case of an error or a continuous loop execute the following steps:

1. Close the console in which all messages and outputs are shown.

   Recommendation: Spyder 4.1.2 closes the current console and opens a new one. A new connection to the kernel is set-up.

2. Run the program again. Error should be gone.

Additional Options:

*) If the error is still occuring, close the program (GUI, working environment ...) and restart it.

*) Another option is restarting the computer and the program.

Notes

As previously stated, the program was tested and written using Spyder 4.1.2 and Python 3.7. It has not yet been tested with a full trial set-up. It his highly recommendable to execute a full test run to determine the required functionality of the program.

Especially lsl-specific functions found in lsl_Functions.py (StreamInfo, StreamOutlet) and corresponding lsl functions found in fig_update.py (push_sample) are to be checked for correct functionality.

The rle-Function located in timing_Functions is also required to be tested with a full set-up. Both, the commented and uncommented code parts were successfully tested. The uncommented part was used for all testing purposes of the program.

A full trial set-up may lead to a different input to the rle-Function and therefore the commented part may be required.

The source code is open for any modifications and can be fully customized.

Helpful Links

Further information to certain packages and helpful supporting pages can be found here:

- Psychtoolbox: https://pypi.org/project/psychtoolbox/
- Python Help Page: https://www.python.org/about/help/
- Programming Help Page: https://stackoverflow.com/
- PsychoPy Page: https://psychopy.org/