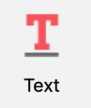
Erasmus+

Project no: 2023-1-TR01-KA210-VET-000165855

Psychopy hands-on exercises

In this exercise you will use Psychopy to create a simple experiment: a Stroop task where a participant responds to the color of a color word, where the word can be congruent (e.g. the word ”green” shown in green) or incongruent (e.g. the word ”red” shown in green) with the color.

1. First you will need a text stimulus, find the text stimulus with the icon shown below in the components menu and click it to add one to your experiment.

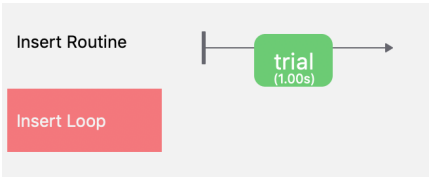


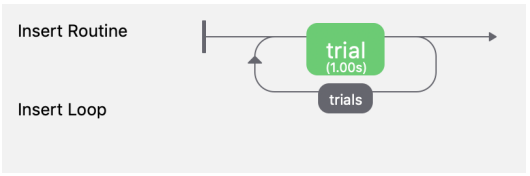
If you want your stimulus to be presented until response, rather than for a fixed duration, you can remove the content of the ”duration” box and leave it empty.

2. Next you will need to specify the text and text color.

For this you will first need to create a table with two columns, one for the text and one for the color. This can be created in Excel or Numbers (on a Mac) or a similar program. Each column should start with a variable name, e.g. ”textcolor” for the color of the text. Specify two text strings: ”red” and ”green” as well as two text colors: red and green. Add every possible combination of strings and colors (4 in total).

In order for Psychopy to use the values from your table you will need to do two additional things.
First, insert a loop around the single routine that your experiment has (for now), like below:

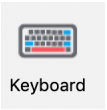




In the properties of the loop, choose the table that you created as the conditions file. If you have done everything correctly you should have 4 conditions with 2 parameters (shown below the conditions box).

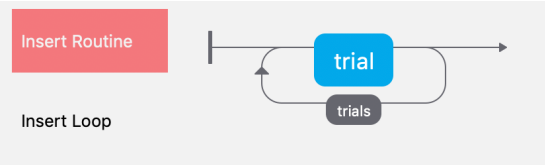
Next, you need to instruct Psychopy to use the variables from your table as text and text color. Change the ”Text” and ”Foreground Color” properties of your text stimulus to the variable names from your table, with an added ”$” sign at the beginning (e.g. ”$textcolor” if you called your variable ”textcolor”). Remember to change these properties to ”Set every repeat” rather than ”Constant” in order to allow them to be changed from trial to trial.

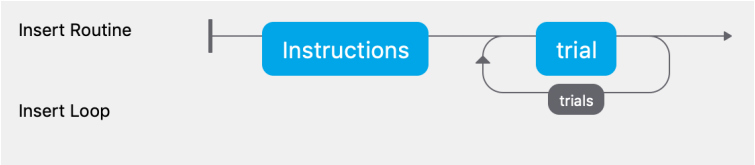
3. Now, you have a sequence of Stroop stimuli, but no possibility of responding. For this, you need to add a keyboard response object:



Choose two keys for the ”red” and ”green” responses and specify these in the ”Allowed keys” property of the keyboard object.

4. At this point you may also want to add an instructions screen. To do this, create a new routine and insert it before the loop:



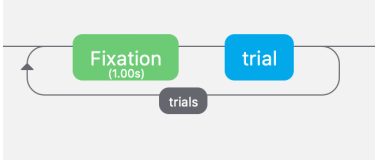


Add a text object to this new routine where you can write some basic instructions, and set the duration to empty (visible until response). Also add a keyboard response, so that participants can press a key to

continue. You could also add an end of experiment screen, with e.g. a ”thank you for participating” message, in the same way.

You now have a basic Stroop task experiment, this may be a good point to attempt to run the experiment and see if it works (if you did not already do this).

5. Perhaps you want a fixation cross before the Stroop stimulus appears on each trial. For this you could add an additional routine before the main stimulus routine inside your loop:



Inside this routine, you could add a polygon stimulus with a fixed duration (e.g. 1 second) and with the ”Shape” property set to ”cross”.

If you wanted the duration of fixation to vary from trial to trial, e.g. to make the onset of the stimulus less predictable in order to discourage anticipatory responses (i.e. preparing for a particular response before the stimulus appears, this is more relevant in experiments where one response occurs more frequently than others), you could add the duration of fixation as another column in your conditions table and use this variable, instead of a fixed value, for the duration of the cross.

Alternatively, you could randomize the fixation duration on each trial with a line of code something like this:

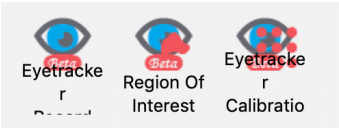
fixdur = 0.7 + np.random.rand()*0.3

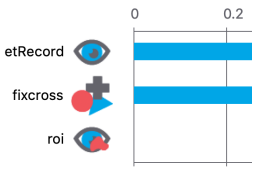
which would give you a random number uniformly distributed between 0.7s and 1s. If you do this, I would recommend also adding this variable to your data file, so that you can know later what the random duration on each trial actually was (without looking in the log-file, which is possible, but much more difficult). You can do this with a line of code like this:

thisExp.addData('fixdur', fixdur)

6. It can also be helpful to specify the correct response on each trial. This way Psychopy will automatically add a column indicating correct and incorrect responses to your data file. You can do this by checking the ”Store correct” box, under the ”Data” tab, in your keyboard response object, and specify the name of a variable with the correct response key in the ”Correct response” box. You could add the correct response key as an additional column in your conditions table, or if you know a little bit of Python, you could create this variable with an if-else statement in the ”Begin Routine” tab of a code component, i.e. check if the text color is red or green and set the response key accordingly.

7. Perhaps you are running your experiment with eye-tracking, and you want to make sure the participant is fixating before the stimulus appears. For this you would first need to select ”MouseGaze” as ”Eyetracker device” in the ”Eyetracking” tab of the experiment settings. Then add a calibration routine before the loop. Also add an ”Eyetracker record” component as well as a ”Region of interest” component to your fixation routine.





In the ROI component, select an appropriate region (e.g. a circle with 2 degree radius centered at the location of the fixation cross). Choose to ”End routine on” looking at the region (”look at”) with an appropriate minimum looking time. Also, change the start time of the ROI so that the start time of the ROI plus the minimum looking time adds up to the minimum duration that you want for the fixation period.

Finally, you can make all the components in your fixation routine have an indefinite duration (duration box empty) so that the fixation routine will only end when the participant is fixating.