

README

Project Title: Change Detection

Name: Hsiang-Yun Wu

Project TA: Byrant

Intro (optional)

Please see the file "Intro" in the folder .

Instructions

How to start:

Enter a working directory that contains these files: main.py, set4.csv, set2.csv.

Run the main project by the command line below:

python main.py

For tests, please run the file: test.py by the command line below

python -m unittest test.py

Features

1. We display arrays of stimuli. The center of the grid and the positions of 49(7*7) checks are calculated and stored in variables called loc and xys. A function("randomposition") is created to randomly choose a position in each quadrant. Another function("whole_exp") is created to display stimuli twice in each trial. This function will assign different colors to checks selected in "randomposition" according to the integers in the color files("set2.csv", "set4.csv"). In each trial, we either display the same squares with the same colors or change one of the colors of one square in the second image. This "whole_exp" function is called 40 times in each session with different colors in each trial assigned by the color files.
2. A function("save_behavioral_data") that saves the experiment data in the form of csv. file. The function collects the participants' data using a dictionary. The dictionary will have three keys: one record the response key of the participant; another record the response time of the participant; the other will record the correctness of the answers. Each key is then attached to the corresponding list. "pd.concat" is used to combine the randomized color files with the participant's responses.

Justification for Complexity

- *This project displays 2D stimuli using visual.ElementArrayStim*
- *Complete process of data reading, processing, and writing*
- *The combination of csv. file and the list of list of colors ("all_colors") to determine the color of stimuli*
- *Efficient use of functional style to prevent repetitive code*

Lists & Script Variables

Lists

- *response_key*: store participant's key response
- *response_time*: store the response time in each trial
- *response_correct*: store the correctness of participant's key response
- *xy*s: store the position of each check (49 checks in total)

Script Variables

- In *function(trial)*:
 - *ran_one*, *ran_two*, *ran_three*, *ran_four*: the position of checks chosen for four quadrants
 - *cross*: the cross stimulus
 - *clock_trial*: the clock object used to time the participant's response in each trial
 - *all_color*: define colors by their *rgb* values

Function Table

Each row will describe the functionality of one custom block/function, with the relevant information placed in the relevant columns (as defined above). You must include a separate row for EVERY custom block/function you create

Block / Function Name	Domain (inputs)	Range (outputs)	Behavior (role in the context of the project)
randomposition()	N/A	Integers, index numbers that are randomly chosen from the lists of index numbers of each quadrant	Randomly choose a position in each quadrant.
check_ans(inputkey, ans)	inputkey: the participant's response (Participants are asked to press M if the encode and probe are identical and press C when they are different) ans: the correct ans (1: they are identical 0: they are different)	integer/string, the correctness of the participant's answer (1:correct; 0: incorrect; "NA": no response)	Check whether the participant's response is correct and return the correctness
trial(stim,encode_c1, encode_c2,encode_c3,encode_c4, probe_c1, probe_c2, probe_c3, probe_c4)	stim: arrays of stimuli defined in line 86 encode_c1,encode_c2, encode_c3,encode_c4: integers that specify the color of each check in the encode period	N/A	This function performs the procedures of a trial. This function is called 40 times in each session.

	probe_c1, probe_c2, probe_c3, probe_c4: integers that specify the color of each check in the probe period		
welcome()	N/A	N/A	This function presents the welcoming words.
session_intro(words)	words: the TextStim object that contains the words we want to present.	N/A	The function presents the introduction before the start of each session.
read_file(filename)	filename: the name of the file that we want to read	dataran: the color file with randomized order. encoding_first, encoding_second, encoding_third, encoding_fourth: integers that specify the color of each check in the encode period probe_first,probe_second,probe_third,probe_fourth: integers that specify the color of each check in the probe period ans: the correct answer (1: identical; 0:	This function reads the csv file, randomizes its order, creates variables of each column and returns the randomized data and columns that will be used in the trials.

		different)	
trial_repeat(times)	times: how many times the function(trial) is called	N/A	Repeat trials by iterating through the columns of color integers in each repetition
save_behavioral_data(response_key, response_time, response_correct)	response_key: the key that the participant presses response_time: the participant's response time in each trial Response_correct: the correctness of the participant's response	N/A	Since the order of trials are randomized and thus different for each participant, the function saves the input colors in the order that is represented to the participant and combines it with the response key, response time and correctness of the participant's response. The combined data is then saved in the form of csv file.

Bug Writeup (optional)

Use this space to explain the bug and the debugging process you went through before the project deadline. Make sure to include what situation the bug comes up in (aka how to replicate the bug) and your best guess about the cause of the bug / what the bug is related to. Include links to any Piazza posts you made regarding the issue.

Extra Credit (optional)

1. Extra feature: A function("read_file") that reads the csv. file and uses the data in that file to assign colors and determine answers. The order of trials are randomized for each participant.
2. Extra feature: A function(check_ans) that checks whether the participant's answer is correct and returns the correctness.
3. The use of higher order functions:
Line 165: In function "whole_exp", the method append is called with input of another function("check_ans").
4. The use of exception handling(try: except:) to keep the project running when the participant presses the wrong key.
5. Significant use of additional libraries. (psychopy, pandas, numpy)