***# Python-Hackathon*** *created on Jun 28, 2018 by python course hackathon group*

*OPEN ISSUES:*

1. *RT is negative*
2. *Output csv needs further check – there might be a problem with the assignment of response to a trial (wrong split)*

*\*\*both bugs probably derives from the same reason*

1. *Function create\_stim\_list in SBDM\_Data.py is now workingwell when csv is transferred and not loaded from .csv file*

*This experiment is composed from 3 parts and based on this paper https://www.nature.com/articles/nn.3673:  
 Part 1:  
 ordering user prefrences and allocating them to go/no go show/no show sanity high/low valus.   
   
 Part 2: Training stage  
 The main part of the experiment consists of the presentation of different snacks to the user, sometimes accompanied by another stimulus(image or sound), usually a tone which then the user should response by click or not (based on cued or not).  
 The pairing of snacks and stimulus relies on the input from the first part of the task.  
 The output of this part is multiindex DataFrame where each block section containts (snack name,RT from cue,success input count, which key pressed) this data from the block   
 the game calls the block,  
 in each each block runs all the trials (snacks) exsists in the dataframe produced from the first part  
 each trial is composed of:  
 fixation cross -> image(snack) -> cue/not cue-> fixation cross  
 differet time intevl between those step varies as a result of the user response  
   
 Part 3:Preference test  
 In the last phase of the experiment, the computer displays the snacks again and asks the volunteer to re-evaluate their value in their eyes.   
 This part relies on the input from the previous two phases of the experiment.   
   
   
 parameters for experimenter to change:  
 break\_text: "BREAK:\n Press any key to continue when you are reday to start the next step"  
 cue\_image\_name: astriks\_gabor.jpg  
 cue\_position: must be tuple for psychopy package purposes  
 cue\_size: list. in pixels  
 cue\_sound\_value: musical note liike "G"  
 cue\_time:   
 cue\_type: sound  
 end\_exp\_img: end\_exp\_img  
 images\_folder\_path: current folder-Images/  
 iti: 1 sec  
 itis: 0.3 sec  
 key\_to\_press: keyboard key  
 pre\_cue: 0.7 sec  
 stim Position: must be tuple for psychopy purposes  
 stim\_size: in pixels  
 stim\_time: 1 sec  
   
 The main function   
 draws parameters file into dictionary,   
 collects user data throu a dialog box (subject ID, Age, Gender)  
 and saves the output of each part  
   
 part 1:  
 1.a.  
 'main' is the function that runs the BDM   
 input: subject ID  
 output: txt rankig called bid   
 \*\*\*\*\*Important Notes:\*\*\*\*  
 1) BDM requires folders called Output, Images in the same folder its function is.  
 2) list of strings of images name (e.g. 'snickers.jpg') is hard coded into the BDM function  
 so any changes happening with snacks stock should be considered in there too.*

*3) subject IDs should be unique. Otherwise it doesn’t save the new subject data but the old one.   
 1.b.  
 takes the txt file and excel file which is pratically a lookuptable of keys stating which rank should get which value  
 of cue/not cued (Go/No\_go) and more.  
   
 output: DataFrame   
 that contains stim name (snack name), Bid(1-10) (user input of snack value) , ranking(1-num of snacks), go\_no/go(bool), sanity high/low(Enum objects), show/no\_show(bool)   
   
 Part 2:  
 Class Block  
 this class runs all the trials togeter (in a random order) for each block,   
 saving the user input (drown from each trial) and stacks all trials into dataframe  
   
 Class Trial   
 this class runs each stimulus given from the block section in the following general scheme:  
 fixation cross -> image(snack) -> cue/not cue-> fixation cross  
 Differet time interval between those step varies based on user RT (e.g. itis, pre\_cue)  
 these sre changed with Block class by success and faliure count  
 other time intervals are changable through parameter file (stim time, cue time etc.)  
 the images are also changing betweeen trials   
 cue type can be changed per game(tone or image of gabor with stricks)  
 from each trial you can get: (snack name,RT from cue,success input count, which key pressed)  
 which will later be drown by block and stacked to final result file  
   
 Class Stimulus  
 the class creates a stimulus object which will represent any image that should be presented on screen  
 with the following attributes:  
 show - bool. should this stim be shown  
 cue - bool. should this stim accompanied with a cue (sound or image)  
 still - bool. if pic is still and waits for user cue ->stil=True  
 if pic presented for specified time then stil=False  
   
 class Game:  
 this class intializing the game, if sets the window, the cue/  
 it calls the vblock class and takes a break between blocks (if required)  
 lastly it saves the user input (drown from block, from trials) and stacks all DataFrames into multiindex DataFrame  
 get\_final\_results function will return it.  
   
 part 3:  
 probe\_organizer function  
 Inputs:   
 - subject\_ID (in string form)  
 - main+path (in string form - the working directory which must contain the output and stimulus folders)  
 Output:  
 An array in which each row represents a probe comparison, with the following columns:  
 1. Go item index in list of go items  
 2. No go item index in list of no go items  
 3. Whether the go item should be displayed on the left ('0.0'/'1.0')  
 4. Go item name  
 5. No go item name  
   
 The stim\_folder variable should be changed to be the name of the stimulus folder, in the format "//foldername".  
 The function first imports the sorted BDM file which was created for the participant using their BDM results, and contains the identity of each item (Go item, no go item, etc.)  
 Next, the list of Go and No Go items is multiplied to allow multiple appearances of each item, and the list is randomly scrambled so the appearance is random, and the output array is created and saved to a txt file,  
   
 probe\_run function  
 Inputs:  
 - comparison\_array - the output created by the probe\_organizer function, containing all comparison info  
 - subject\_ID (in string form)  
 - main+path (in string form - the working directory which must contain the output and stimulus folders)  
 Output:  
 - A list of choices, containing a strings indicating the user's choice on each comparison.  
   
 The function pulls the names of go and no go items from comparison\_array.  
 Using a loop, the function uses psychopy to:  
 -Present the comparison on the screen so that one Go item and one No go item appear next to each other. Their position on screen is determined by the left\_go column in comparison\_array.  
 -After the comparison is shown, the function waits for a response from the user, which may be either a 'a' or 'l' key press, to indicate a left and right choice, respectively.  
 -Continue showing comparisons until the user has seen and responses to all of them  
 -Write responses into a list ("choice")  
   
 Main probe script   
 Runs both functions, and then uses their outputs to create a summarizing pandas Dataframe containing the results, which it saves to a csv file in the Output folder.  
 The output file also includes a column called "go\_chosen", which indicated whether the item chosen on that trial was the go item.*