Documentation

Legend	Meaning
Italisized text	Variable class/type (e.g. float, string, int etc.)
Bolded text	Function name
Blue text	Necessary variable
Red text	Optional variable (Default values shown)

File: **main.py**

Definition table explanation

Definition	Description	Inputs	Outputs
GetRefreshRateWindows()	Function to get the monitor refresh rate, assuming a windows operating system.	- None	- RefreshRate = *int*, Monitor refresh rate, in Hz
SetColorPalette(color, ManualAssign = [])	Utility function to set the color palette for the experiment. Colors which are set include: Window background color, Text color, Slider bar color, and Slider marker color (i.e. selection).	- color = *string*, Dominant color for the color palette. Known colors are: 'default', 'grey', 'slate', 'beige', 'white', 'red', 'blue'. - ManualAssign = *list* (length = 4), Option for the user to manually specify colors they want to use. Input should be a list with order [Background, Text, Slider bar, Marker] colors. More colors can be found here: https://www.w3schools.com/Colors/colors_names.asp	- Palette = *list*, *list of strings* (color names) to use for the experiment in order [Background, Text, Slider bar, Marker]
AssignGroups(Num_Participants, seed = 0)	Function to assign participants to the 'engaged' (label = 0) or 'disengaged' groups (label = 1). The function will split the participants evenly among the two groups, unless the number of participants is odd. For such cases, the last participant is randomly assigned to one of the groups	- Num_Participants = *float* or *int*, Total number of participants - seed = *int*, seed for random number generation. Default is 0.	- GroupAssign = *list*, list of 0's and 1's indicating which group a participant belongs to. The index + 1 corresponds to the participant number. (Note +1 is due to Python indexing starting at 0 and not 1)

GetParticipantInfo(Path2ListOfParticipants, GroupAssignment, Developer=False)	Function which retrieves the participant ID, group assignment, and prompts a dialog box through which participant data (such as age, height, gender etc.) can be recorded. Fields 'participant ID' and 'group assigment' are fixed.	- Path2ListOfParticipants = *string*, File path of a text file which contains the IDs of participants who have completed the experiment. This file is in ascending order so the first participant will be assigned the ID 1 and so on. - GroupAssignment = *list*, List of 0's and 1's indicating which group a participant belongs to. - Developer = *Boolean*, if True, then the script will be run in developer mode; which is used to test functionality. Here, the participant ID = 0. Default = False.	- Dlg_data = *list*, *list of strings* (entries) from the fields of the dialog box. - RunExp = *Boolean*, If true, then the experiment will be run, otherwise it will not. RunExp = False if dialog box is closed/cancelled - AllFields = *list*, *List pf strings* corresponding to Dlg_data which contains the field names of the dialog box (e.g. 'age')
GenSavePath(ParticipantID, DataFolder = 'ExpData')	Function to create a save folder for the participant data. This folder will be saved in one directory level above the program in a folder specified by 'DataFolder'. The participant data is then saved into a subfolder corresponding to their participant number	- ParticipantID = *int*, Participant number - DataFolder = *string*, Folder name in which data should be saved. Default = 'ExpData'	- ParticipantPath = *string*, path to the participant folder
Save2ColCSV(Filename, Fields, Data, ParticipantID, DataCautious = True)	Function to save data in a two column format (i.e. 1st column = variable names and 2nd column = data) as a .csv file.	- Filename = *string*, Name of data file. All files are prefixed with the participant ID - Fields = *list*, *list of strings* of variable names (e.g. age) - Data = *list*, *list of* data corresponding to Fields. - ParticipantID = *int*, Participant number - DataCautious = *Boolean*, If True, then the function will ensure that it does not overwrite existing files. If a file with the same name exists, then the function will instead save the current data appended with a unique ID. If False, then the function will overwrite existing files. Default = True	- None
RecordParticipantIDs(Path2ListOfParticipants, ParticipantID)	Function to record completed participant IDs, such that subsequent participant IDs are automatically created.	- Path2ListOfParticipants = *string*, File path of a text file which contains the IDs of participants who have completed the experiment. This file is in ascending order so the first participant will be assigned the ID 1 and so on. - ParticipantID = *int*, Participant number	- None

ShowVAS(Window, Question, VASLabels, RefreshRate, TickLims = [-15, 0, 15], MarkerColor = 'DarkSlateGrey', TextColor = 'White', SliderColor = 'LightGrey')	Function to display a VAS with accompanying question.	- Window = *window object*, PsychoPy window on which the VAS will be shown. - Question = *string*, question to be asked - VASLabels = *list*, *List of strings* representing the labels for the VAS from left to right (e.g. ['Disagree', 'Agree']) - RefreshRate = *int*, Monitor refresh rate, in Hz - TickLims = *list*, *list of ints* (limits) for the VAS. The length of this list dictates how many 'ticks' (lines) are shown on the scale. By default, there are 3 lines indicating the extremes and the middle. - MarkerColor = *string*, color of markers - TextColor = *string*, color of text - SliderColor = *string*, color of slider bar	- Response = *float*, normalized user answer of the VAS (i.e. from 0 to 1 if leftmost extreme = 0, or from -1 to 1 if the VAS is symmetrical about 0). Normalization occurs w.r.t the rightmost extreme.
ShowSlider(Window, Question, Labels, Ticks, RefreshRate, Style = 'rating', size = (1.2, 0.1), MarkerColor = 'DarkSlateGrey', TextColor = 'White', SliderColor = 'LightGrey')	Function to display a question with a (discrete interval) response slider. The number of response options is equal to the number of Ticks.	- Window = *window object*, PsychoPy window on which the discrete slider will be shown. - Question = *string*, question to be asked - Labels = *list*, *List of strings* representing the labels for the slider from left to right (e.g. ['Disagree', 'Neutral', 'Agree']) - Ticks = *list*, *List of numbers (float or int)* which indicate how many response options the user has for the question. E.g. for a 5-point scale `Ticks = [1, 2, 3, 4, 5]` or another collection of 5 unique numbers (e.g. [-2. -1. 0, 1, 2]). - RefreshRate = *int*, Monitor refresh rate, in Hz - Style = *string*, PsychoPy style of slider. Default is 'rating' which shows a bar. See PsychoPy documentation for more. - Size = *tuple*, pair of *floats* which indicate the size of the slider w.r.t the window size in normalized units (here, 2 = full size). The first element scales the width while the second element scales the height of the bar. - MarkerColor = *string*, color of markers - TextColor = *string*, color of text - SliderColor = *string*, color of slider bar	- Response = *float*, the user answer from the slider. This value corresponds to the value of Ticks at that point. E.g. For `Ticks = [-1, 0, 1]` if the user clicks the middle then `Response = 0`.

AskFoodNeophobia(Window, RefreshRate, MarkerColor = 'DarkSlateGrey', TextColor = 'White', SliderColor = 'LightGrey')	Function to present the Food Neophobia Scale (FNS) to participants	- Window = *window object*, PsychoPy window on which the FNS questions will be asked - RefreshRate = *int*, monitor refresh rate, in Hz - MarkerColor = *string*, color of markers - TextColor = *string*, color of text - SliderColor = *string*, color of slider bar	- Questions = *list, list of strings* containing the Neophobia questions - Answers = *list, list of integers* (from 1 to 7) indicating the user's responses to the questions. Any reversals, for relevant questions, are already applied in these questions.
GetImages(FolderPath)	Function to import all images of specified type from specified folder	- FolderPath = *string*, path to the folder where the images are located. This path should be appended with the desired file type. For example, to import JPEG images use FolderPath = 'Path/to/Images/*.jpg'	- Imgs_path = *list, list of strings* which indicate the path to each of the images. Images are only imported when they are needed.
RandomIzeImageOrder(ImageList, seed = 0)	Function to randomize the order of presentation for each image, within a category. The order of categories is also randomized, with the condition that an image from each category is shown at least once before images from an already-shown category are presented.	- ImageList = *list, list of list of strings* which indicate the path to each of the images. Shape = Number Categories x Number Images. Assumption: The number of images in each category is the same. - Seed = *int*, seed for randomization. Default = 0. For randomization across participants, the participant ID can be used as the seed.	- RandImageList = *numpy array*, randomized list of strings which indicate the path to each of the images. Images are only imported when they are needed. Shape = Number Categories x Number Images - RandomizedImageOrder = *numpy array*, bookkeeping of the randomization. It holds the indices of the randomization such that, when applied to the original image list, the result would be RandImageList above. Shape = Number Categories x Number Images. - CatOrder = numpy array, list of the order of presentation of the categories. Shape = Number Images x Number Categories

ShowImage(Window, ImagePath, RefreshRate, Duration, Scale = 1)	Function to show image stimuli	- Window = *window object*, PsychoPy window on which the image will be shown - ImagePath = *string*, path to the image to be shown - RefreshRate = *int*, monitor refresh rate, in Hz - Duration = *float*, Duration of image presentation, in seconds - Scale = *float*, scale of the image w.r.t (smallest dimension in) window size. Scale = 1 means image is as large as possible (while maintaining aspect ratio).	- None
ShowText(Window, Text, RefreshRate, Duration, Position = (0,0), Height = 0.15, TextColor = 'White')	Function to show text stimuli	- Window = *window object*, PsychoPy window on which the image will be shown - Text = *string*, text to display - RefreshRate = *int*, monitor refresh rate, in Hz - Duration = *float*, Duration of text presentation, in seconds - Position = *tuple, tuple of floats* indicating the position of the text on the window. (0, 0) is at the center of the screen and (1, 1) is the top right. - Height = *float*, height of the text scaled w.r.t monitor size. - TextColor = *string*, color of text	- None
ShowMovie(Window, MoviePath, Scale = 1)	Function to show video stimuli	- Window = *window object*, PsychoPy window on which the movie will be shown - MoviePath = *string*, path to the movie file - Scale = *float*, scale of the movie w.r.t (smallest dimension in) window size. Scale = 1 means movie is as large as possible (while maintaining aspect ratio).	- None
ShowEmojiGrid(Window, RefreshRate, Scale = 1, Position = (0, 0))	Function to show (interactive) EmojiGrid response tool	- Window = *window object*, PsychoPy window on which the EmojiGrid will be shown - RefreshRate = *int*, monitor refresh rate, in Hz - Scale = *float*, scale of the EmojiGrid w.r.t (smallest dimension in) window size. Scale = 1 means EmojiGrid is as large as possible (while maintaining aspect ratio). - Position = *tuple, tuple of floats* indicating the position of the text on the window. (0, 0) is at the center of the screen and (1, 1) is the top right.	- PosOnGrid = *list*, pair of coordinates indicating the response location on the grid. Here, [0, 0] indicates the origin, [1, 1] gives the top right, and [-1, -1] gives the bottom left of the EmojiGrid.

SaveImageResponseData(Filename, ImgList, Data, ParticipantID, ColNames = [], DataCautious = True)	Function to save the self-report (e.g. EmojiGrid) responses to the image stimuli.	- Filename = *string*, Name of data file. All files are prefixed with the participant ID - ImgList = *list*, list of strings containing the file names of the images (e.g. IMG_01.jpg), prefixed by their category (e.g. Asian, Dutch, Molded) - Data = *numpy array*, 2-D array of response data (e.g. coordinates on EmojiGrid). Shape = Number IMGS x Number data points per image - ParticipantID = *int*, participant number - ColNames = *list, list of strings* containing the names of each column. The number of columns should correspond to the number of columns in 'Data'. - DataCautious = *Boolean*, If True, then the function will ensure that it does not overwrite existing files. If a file with the same name exists, then the function will instead save the current data appended with a unique ID. If False, then the function will overwrite existing files. Default = True	- None
ShowEmoGrInstruction(Window, Instructions, RefreshRate, Scale = 1.5, TextColor = 'White')	Function to display text (Instructions) next to an interactive EmojiGrid. To proceed with the experiment, users must click somewhere on the EmojiGrid (demonstrating it's use). Text is displayed to the left of the EmojiGrid.	- Window = *window object*, PsychoPy window on which the EmojiGrid and text will be shown - Instructions = *list, list of strings* which represent the accompanying text. The first line is considered a 'title' and will thus be larger than the other text. Each string in the list corresponds to a new line. E.g. ['Title', 'Line 1', 'Line 2', …, 'Line N'] - RefreshRate = *int*, monitor refresh rate, in Hz - Scale = *float*, scale of the EmojiGrid w.r.t half the (smallest dimension in) window size. Scale = 1 means image is as large as possible (while maintaining aspect ratio). - TextColor = *string*, color of the instruction text.	- PosOnGrid = *list*, pair of coorindates indicating the response location on the grid. Here, [0, 0] indicates the origin, [1, 1] gives the top right, and [-1, -1] gives the bottom left of the EmojiGrid.

ShowImInstruction(Window, Instructions, ImagePath, RefreshRate, Scale = 1, TextColor = 'White')	Function to display text (Instructions) next to an image. To proceed with the experiment, users must click on 'Next'. Text is displayed to the left of the Image.	- Window = *window object*, PsychoPy window on which the Image and text will be shown - Instructions = *list, list of strings* which represent the accompanying text. The first line is considered a 'title' and will thus be larger than the other text. Each string in the list corresponds to a new line. E.g. ['Title', 'Line 1', 'Line 2', …, 'Line N'] - ImagePath = *string*, path to the image. - RefreshRate = *int*, monitor refresh rate, in Hz - Scale = *float*, scale of the image w.r.t half the (smallest dimension in) window size. Scale = 1 means image is as large as possible (while maintaining aspect ratio). - TextColor = *string*, color of the instruction text.	- None
FrameWait(Window, RefreshRate, Duration)	Function to pause the experiment for a fixed duration.	- Window = *window object*, PsychoPy window on which the pause will occur - RefreshRate = *int*, monitor refresh rate, in Hz - Duration = *float*, duration of the pause, in seconds	- None
CheckQuitWindow(Window)	Function to check if 'esc' has been pressed. If so, then it will close the window.	- Window = *window object*, PsychoPy window for which the check will occur	- None