Homework 10 **PSY4219/6219**
Due Tuesday December 13 at 11:59pm **Fall 2022**
30 points

Q1 (30 points). I want you to create a full experiment using PsychoPy and other tools and techniques we have talked about in this class. This is a fairly simple experiment, but a full experiment nonetheless.

To make clear the grading rubric, I have put in *italics* the points (e.g., *2 points*) for each components required to achieve full credit.

In a classic recognition memory experiment, subjects first study a list of items one at a time. After study, subjects are tested on old items and new items one at a time and simply need to decide whether every test item is "old" (studied) or "new" (not studied).

For this experiment, your study and test items will be images of common objects. The images I want you to use are available on Tim Brady's web site at UCSD: https://bradylab.ucsd.edu/stimuli.html

In particular, I want you use the **2400 Unique Objects** image set from his web site.

These images were used in this well-known paper: Brady, T. F., Konkle, T., Alvarez, G. A., & Oliva, A. (2008). Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences, USA*, 105 (38), 14325-14329.

In addition to demonstrating that you can implement a well-designed experiment, I want you to show me that you can code up your experiment efficiently and logically, with comments and proper structure, including the use of functions (*2 points*). Use proper Python and PsychoPy style as it relates to displaying images, carefully controlling image timing (*2 points*), obtaining responses and response times (*2 points*), using try-except to recover gracefully from errors, and the like (*1 point*). Make sure you seed the random number generator so that different runs of the experiment (by different subject in the experiment) produce different randomizations of images (*2 points*), but also make sure that the seed is saved so that the experimental sequence (the randomization) can be reproduced (*1 point*).

I want to see that your code is flexible enough to allow several aspects of the experiment to be parameterized by the experimenter (*2 points*). Before launching PsychoPy, your program should ask the experimenter to enter in (a) enter the subject number, (b) the length of the study list (between 5 and 50 images), and (c) the time each study image is to be presented (in seconds, allowing for times that are a fraction of a second to be entered). I want you to use the gui capabilities in PsychoPy, as discussed and demonstrated in class (*2 points*).

For the study phase, you should randomly select the specified number of study images needed from the folder of images you downloaded from Tim Brady's website. You should

use the technique demonstrated in class of reading a folder of files names, not hard-coding the image names into Python. (*2 points*)

Of course, these study images should be presented in random order to the subject.

Each study image should be presented one at a time, with the presentation time given by the time entered by the experimenter. Assume a 0.5 second blank interval between study items (but make sure your program is coded so that this is a variable that can easily be found and set to a different value if needed). (*1 point*)

We will test to make sure your actual study sequence timing matches precisely (within a few milliseconds) what it should be based on the length of the study list and the time each study image is to be presented. I want you to use the logging capabilities in PsychoPy to log the time of a particular flip, with the text accompanying each log informative of what is being logged. (*2 points*)

For the test phase, you should randomly select an equal number of new images (as old images) from the folder of images. Old items and New items should be randomly intermixed. Each test image should be presented until the subject makes an old/new recognition response. You should add some text on the screen telling the subject (and us) which keyboard key corresponds to "old" and which keyboard key corresponds to "new". (*1 point*)

Record both the choice and the response time using the methods we discussed in class. Since you are recording response times, make sure you catch cases where someone hits an invalid key first. (*1 point*)

The data from the experiment (as well as all of the details of each trial, including, for example, the image shown, the timing of the presentation, and the like) should be stored in a pandas dataframe and saved as a .csv file (as described and discussed in class). (*2 points*)

Also write a short program that reads your .cvs file into a pandas dataframe and summarizes the. For this assignment, I simply want you to calculate (and print to the Console) the proportion of hits (saying "old" to an old image) and proportion of false alarms (saying "old" to a new image) at test. (*3 points*)

For this assignment, you can assume that a subject only participates in one study-test sequence. But you should write your code so that it would be relatively easy for someone to take what you wrote and embed it in a larger program where, for example, each subject might study lists of three different lengths and study images for one of two different study intervals. So while I'm not asking you to write code for this more complex experiment, I am asking you to write your code so that it could be easily ported over and used in a more complex experiment. That means that you will need to break down your code into logical functions that can be reused. (*2 points*)

When you submit your assignment, DO NOT submit the Brady image set.

Unexcused late assignments will be penalized 10% for every 24 hours late, starting from the time class ends, for a maximum of two days, after which they will earn a 0.