

Problem Set 3

November 2016

1 First experiment

In this first experiment, we will propose a very simple experiment, so that you can get used to Psychopy.

In this experiment, you will run 50 dot detection trials and store the data collected from each trial in a matrix. Each trial starts with a cross shape at the center of the screen, on which the subject must fix their eyes for 1s. After a period of time selected randomly between 0.5s and 1.5s, either a black dot or nothing will show up on the monitor for some time selected randomly between 50ms to 500ms. After this, a green square will show up in the center of the monitor to inform the subject to make a decision.

The subject needs to press 'Left' button when he or she detects the dot, and press 'Right' button if no dot appeared before the square showed up.

1.1 General code setup

First of all, we have to configure some settings so that Psychopy can start running. Make sure the window size matches the size of your own window, so that we can run this experiment in a full screen.

```
from psychopy import core, visual, event
from numpy import random
import json      # Import Psychopy libraries to your code
win = visual.Window(size=(1600,900), color='black')      # Open a black window

numTrials = 50
fixationTime = 1
responseVector = []
dotTimeVector=[]
isDotTrial = ?
```

1.2 Trial loop

Then we designed the trial based on what we have discussed. Specifically, we use `.draw()` to draw the stimulus in the buffer and use `.flip()` to show it on

the screen. `.flip()` again will make it disappear. You can use `core.wait(time)` for waitint for some time. Use `event.waitKeys()` for getting information about pressing the button.

```
for i in range(numTrials):
    # Random time between cross and dot
    jitterTime = ?
    # Random dot time
    dotTime = ?
    dotTimeVector.append(?)

    # Define and draw fixation cross
    fixationCross = visual.TextStim(win, color='#FFFFFF', text = "+", units='norm', height=0.1)
    fixationCross.draw()
    # Show fixation cross
    win.flip()
    # Wait for fixation time
    core.wait(?)
    # Remove fixation cross
    win.flip()
    # Wait for random jittered time until dot
    core.wait(?)
    # Plot white dot IF IT IS A DOT TRIAL:
    dot = visual.Circle(win, radius=10, edges=32, fillColor=(10,10,10), fillColorSpace='rgb')
    # Check if this is a trial with a dot
    if i in isDotTrial:
        dot.draw()
    win.flip()
    # Wait for the randomized dot time
    core.wait(?)
    # Remove dot from the screen
    win.flip()
    # Wait a little bit more without the dot
    core.wait(1-?)
    # Draw and display green square for response
    square = visual.Rect(win, width=50, height=50, fillColor=(0,255,0), fillColorSpace='rgb')
    square.draw()
    win.flip()
    # Wait for response
    k = ''
    while k[0] not in ['left', 'right']:
        k = event.waitKeys()
        responseVector.append(k)
# END OF FOR LOOP
```

1.3 Save data

Finally, you need to save the data in a file for analysis in the future. Don't forget to close the window and quit the core.

```
# Saving results to a file
f = open('datafile.txt', 'w')
# JSON allows you to save the full vector to the file by using dump (yes, dump!)
json.dump(responseVector, f) # Saving responses to file
json.dump(isDotTrial.tolist(), f) # Saving trial identifiers to file
json.dump(dotTimeVector, f) # Saving randomized dot time to file

win.close()
core.quit()
```