Python Lecture 5 Notes: Intro to PsychoPy

Miles Martinez
Raphael Geddert

10/13/22

1 Introduction & Key Takeaways

a. Importing Psychopy

b. Opening windows

c. Drawing Stimuli

d. Documentation

2 Task Flows

Task flows are the logic of your experiment, expressed in code. For example, you might say: I would like to display a fixation cross for 1 second, then a blue rectangle stimulus on the screen for 3 seconds, and repeat all of this 5 times.

Expressed in python code, this might look something like this:

```
def displayFixation():
    print('This is where I'll display my fixation')

def displayRect():
    print('This is where I'll draw my rectangle')

nTrials = 5
for trial in range(nTrials):
    displayFixation()
    core.wait(1)

    displayRect()
    core.wait(3)
```

This is always the place you want to start when creating an experiment. It is so much easier to have a working task flow and then fill in the functions that for now just have print statements, than to try to create everything from scratch right away.

3 Importing psychopy

```
from psychopy import visual
from psychopy import core

#or, equivalently
```

```
from psychopy import visual, core
```

4 opening windows

```
win = visual.Window(size=(height,width),\
pos=(xcoord,ycoord),color=[r,g,b],other arguments)
```

The first step for any psychopy experiment is opening a display window, onto which we will draw stimuli, text, images, etc. The size input is in pixels, as is the position input. However, position input is centered at the center of your screen - so negative x coord values are to the left of the center and negative y coord values are below the center. Color is similar - it requires a list of three values from -1 to 1; the amount of red, amount of green, and amount of blue. -1 indicates no color, while 1 indicates the maximum amount of color in that color channel. For that reason, [1,1,1] is a white window, [-1,-1,-1] is a black window, and [0,0,0] is a gray window.

> !!!!!!
> Don't forget to close your windows once you're done with win.close() and/or core.quit()
> !!!!!!

5 Drawing Stimuli

we can use things like visual.Circle(), visual.square(), and other options here at this link. Once your initialize these, do something like the following:

```
my_stim = visual.Circle(window,...)
my_stim.draw()
win.flip()
```

The first line creates our stimulus, the second "draws" it on our canvas (kind of like the backside of our screen), and then the third line "flips" our screen, finally displaying our stimulus to our participants.

6 documentation

this link!!!!!