Programming Skills
Lecture 7

PsychoPy

Input & (more) complex designs

Recap

- Presenting stimuli
- Getting response and latency
- Saving data

The experiment loop

```
from psychopy import visual, core, event

win = visual.Window()
msg = visual.TextStim(win)

trials = [‘red’, ‘blue’, ‘green’]

for trial in trials :

    msg.setText(trial)
    msg.draw()
    win.flip()
    clock = core.Clock()
    respond= event.waitKeys(maxWait=6.0, timeStamped=clock)

    if respond:
        response, latency = response[0]
    else:
        response, latency = -1, -1

    outputFile = open(‘data.txt’, ‘a’)
    outputFile.write(“{}\t{}\n”.format(response, latency)
    outputFile.close()

win.close()
```

Other stimulus objects

- ImageStim
- SimpleImageStim
- ShapeStim
- MovieStim

`visual.ImageStim(win)`
parameters

`win = windowObject`

Required: the stimulus must know in which window to draw itself

`image = “image name.jpg”`

Image file to be rendered

`size = (1, 0.5)`

size of the image

`position = (-0.75, 0.75)`

position of the image

`units = “norm”`

Controls what the position and size values mean

visual.ImageStim(win)

some methods

.setImage(“image.jpg”)

Change the image

.setSize() or .size

Change the size, this works too:

```
stim.size = stim.size * 2
```

.setPos() or .pos

Change position, this works too:

```
stim.pos = stim.pos + 1
```

.draw() or .setAutoDraw(True)

Draw the stimulus to the video buffer

More on: http://www.psychopy.org/api/visual/imagestim.html

Tip

Use ImageStim for instruction!

Blocks

```
# open your window
# prepare the stimulus objects

# for block in blocks:
     # update trials list / response options
     # for trial in trials:
          # update stimuli
          # draw stimuli
          # flip your window
          # get response
          # write to data file

# close your window
```

Input

- PsychoPy GUI module
  - Dlg
  - DlgFromDict
- Create your own input functions
- (PsychoPy Question Library)

Conditions

- Between subjects
  - Often easiest to decide based on subject nr to keep cell counts equal:

    ```
    condition = subjectNr % nrConditions
    ```

- Within subject
  - Within block manipulation
  - Blocked manipulations

Within-block manipulations

- Different trial types, for different levels of within subject variables
- Keep track of trial types to make sure every trial type is presented (equally often)
- This asks for more structure, and for more advanced randomization than a simple `shuffle`

data.TrialHandler

- Uses Excel file (.xlsx) to identify:
  - Conditions
  - Corresponding manipulations
  - Corresponding stimuli
- Takes care of randomization
- Takes care of storing data

TrialHandler Excel file

- Each row is a trial type
- First row contains variable name
- The rows below contain the levels of the variable (including stimulus)

TrialHandler example

```
from psychopy import data

# import conditions
conditions = data.importConditions(‘names.xlsx’)

#create trial handler
trials = data.TrialHandler(conditions, 2)

for trial in trials :
    print trial[‘stimulus’], trial[‘group’]
```

`data.TrialHandler()` parameters

`Conditions`

Required: conditions object, created using
`data.importConditions(‘conditions.xlsx’)`

`nReps`

Number of repeats for all conditions

`Method`

‘random’, ‘sequential’, ‘fullRandom’

`extraInfo`

Dictionary with additional info to save
(like subject, age, gender)

Attributes after creation

`.nTotal`

total number of trials

`.nRemaining`

total number of trials remaining

`.thisN`

total number of trials completed

`.finished`

Boolean expression whether it is done

Methods

`.next()`

Advances to next trial and returns it

`.getFutureTrial(n=1)`

Returns the condition for n trials in the future without advancing it

.**`getEarlierTrial(n=-1)`**

Return the condition for n trials previously

Data storage TrialHandler methods

```
.addData(‘variablelabel’, value)
```

add data for the current trial

```
.saveAsText(‘textfile’)
```

saves all data to a text file

```
.saveAsExcel(‘excelfile’)
```

saves all data to an excel file

Display error feedback

Easy with TrialHandler: check whether response is correct for the current trial’s condition

```
if trial[‘condition’] == ‘male’:
    if response == ‘a’:
        trials.addData(‘correct’,1)
    else:
        trials.addData(‘correct’,0)
```

Blocked manipulations

- Just use separate TrialHandler for each block if necessary
- Manipulate order of blocks as between-subjects manipulation
  - using shuffle
  - based on subject number for counter-balancing

Recap

- Presenting images
- Blocks
- Conditions
- TrialHandler
- Counterbalancing

This week’s homework

Program a Stroop task using PsychoPy

- 32 trials: 4 ink colors x 4 color names x 2 presentations
- Task: name the ink the stimulus is written in, possible responses are the 4 color names (so use four different keys)
- Use TrialHandler to load conditions, randomize and run the trials, and save the data
- Display instructions using ImageStims
- Give error feedback when subject makes a mistake
- Save trial number, ink color of the stimulus, color name of the stimulus, response, latency, whether the response is correct (note: some of these will be saved automatically with TrialHandler!) to a text or excel file
- Also save subject number, subject age, and subject gender

Practice exam

- Online now
- Solution will be online Friday 3th at 10:00 AM
- Q&A in next lecture