
pylinkwrapper Documentation

Release 1.0

Nick DiQuattro

October 23, 2015

1	Installation	3
2	Connector Class	5
3	Sample Experiment	7
4	Indices and tables	9
	Python Module Index	11

Contents:

Installation

1. [Download](#) and place the pylinkwrapper folder somewhere you can easily reach in the file system.
2. Add the folder to Psychopy's python path as detailed here: <http://www.psychopy.org/recipes/addCustomModules.html>

That's it! Here's a quick demo of getting connected to the Eyelink:

```
import pylinkwrapper

win = visual.window(monitor='nickMon', fullScr=True, allowGUI=False, color=-1)
tracker = pylinkwrapper.connect(win, '1_nd')
```

Check out the documentation for the [Connector Class](#) class to see what functions are available.

Connector Class

This class provides methods for interacting with the EyeLink from psychopy.

class `pylinkwrapper.connect` (*window*, *edfname*)

Provides functions for interacting with the EyeLink via Pylink.

Parameters

- **window** – Psychopy window object.
- **edfname** – Desired name of the EDF file.

calibrate (*cnum=13*, *paval=1000*)

Calibrates eye-tracker using psychopy stimuli.

Parameters

- **cnum** (*int*) – Number of points to use for calibration. Options are 3, 5, 9, 13.
- **paval** (*int*) – Pacing of calibration, i.e. how long you have to fixate each target.

drawIA (*x*, *y*, *size*, *index*, *color*, *name*)

Draws square interest area in EDF and a corresponding filled box on eye-tracker display.

Parameters

- **x** (*float or int*) – X coordinate in degrees visual angle for center of check area.
- **y** (*float or int*) – Y coordinate in degrees visual angle for center of check area.
- **size** (*float or int*) – length of one edge of square in degrees visual angle.
- **index** (*int*) – number to assign interest area in EDF
- **color** (*int*) – color of box drawn on eye-tracker display (0 - 15)
- **name** (*str*) – Name interest area in EDF

drawText (*msg*)

Draws text on eye-tracker screen.

Parameters **msg** (*str*) – Text to draw.

endExperiment (*spath*)

Closes and transfers the EDF file.

Parameters **spath** (*str*) – File path of where to save EDF file. Include trailing slash.

fixCheck (*size*, *fime*, *button*)

Checks that fixation is maintained for certain time.

Parameters

- **size** (*float or int*) – Length of one side of box in degrees visual angle.
- **ftime** (*int*) – Length of time to check for fixation in seconds.
- **button** (*char*) – Key to press to recalibrate eye-tracker.

recordOFF ()

Stops recording.

recordON (*sendlink=False*)

Starts recording. Waits 50ms to allow eyelink to prepare.

Parameters **sendlink** (*bool*) – Toggle for sending eye data over the link to the display computer during recording.

sendCommand (*cmd*)

Sends a command to the Eyelink.

Parameters **cmd** (*str*) – Command to send.

sendMessage (*txt*)

Sends a message to the tracker that is recorded in the EDF.

Parameters **txt** (*str*) – Message to send.

sendVar (*name, value*)

Sends a trial variable to the EDF file.

Parameters

- **name** (*str*) – Name of variable.
- **value** (*float, str, or int*) – Value of variable.

setStatus (*message*)

Sets status message to appear while recording.

Parameters **message** (*str*) – Text object to send, must be < 80 char

setTrialID (*idval=1*)

Sends message that indicates start of trial in EDF.

Parameters **idval** – Values to set TRIALID.

setTrialResult (*rval=0, scrcol=0*)

Sends trial result to indicate trial end in EDF and clears screen on EyeLink Display.

Parameters

- **rval** (*float, str, or int*) – Value to set for TRIAL_RESULT.
- **scrcol** – Color to clear screen to. Defaults to black.

Sample Experiment

This is a very simple experiment that demonstrates use of `pylinkwrapper`. It can be found in the `sample` folder.

```
'''
Pylink Wrapper test experiment
N. DiQuattro - January 2015

This is a simple experiment where a circle appears randomly on the screen. It's
purpose is to provide examples of how to use the pylink wrapper with a psychopy
experiment.

There's help documentaiton available for each of the functions that show the
available parameters.
'''

# Import modules
from psychopy import visual
from psychopy import core, event
import numpy as np

import pylinkwrapper# Here's the special one

# Window set-up
win = visual.Window(monitor = 'nickMon', units = 'deg', fullscr = True,
                    allowGUI = False, color = 0)

# Initiate eye-tracker link and open EDF
tracker = pylinkwrapper.connect(win, '1_test')

# Calibrate eye-tracker
tracker.calibrate()

# Stimulus
fix = visual.Circle(win, radius = 1, pos = (0, 0), fillColor = [1, 0, 0],
                    lineColor = [1, 0, 0])

cfix = visual.Circle(win, radius = .15, fillColor = -1, lineColor = -1)

# Display stimulus 5 times
for t in range(5):

    # Find random coordinates and set them
    fx = np.random.randint(-10, 10)
    fy = np.random.randint(-10, 10)
```

```
fix.setPos((fx, fy))

# Eye tracker trial set-up
stxt = 'Trial %d' % t
tracker.setStatus(stxt) # Define status message that appears on eye-link
                        # display

tracker.setTrialID() # Sends trial start message for EDF
tracker.sendMessage('Circle Trial')

# Draw IA
tracker.drawIA(fx, fy, 1, 1, 5, 'circle') # Draw interest area and box

# Start recording
tracker.recordON()

# Draw and display circle
cfix.draw()
fix.draw()
win.flip()

# Wait for response
keyp = event.waitKeys()

# Stop Recording
tracker.recordOFF()

# Send response key to EDF file
tracker.sendVar('response', keyp[0][0])

# End trial for EDF
tracker.setTrialResult()

# ISI with fixation check
cfix.draw()
win.flip()
tracker.fixCheck(2, 1, 'z')

# Retrieve EDF
tracker.endExperiment('C:\\\\edfs\\') # Closes and retrieves EDF file to
                                   # specified path
```

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`pylinkwrapper`, 5

C

calibrate() (pylinkwrapper.connect method), 5
connect (class in pylinkwrapper), 5

D

drawIA() (pylinkwrapper.connect method), 5
drawText() (pylinkwrapper.connect method), 5

E

endExperiment() (pylinkwrapper.connect method), 5

F

fixCheck() (pylinkwrapper.connect method), 5

P

pylinkwrapper (module), 5

R

recordOFF() (pylinkwrapper.connect method), 6
recordON() (pylinkwrapper.connect method), 6

S

sendCommand() (pylinkwrapper.connect method), 6
sendMessage() (pylinkwrapper.connect method), 6
sendVar() (pylinkwrapper.connect method), 6
setStatus() (pylinkwrapper.connect method), 6
setTrialID() (pylinkwrapper.connect method), 6
setTrialResult() (pylinkwrapper.connect method), 6