



Erasmus+



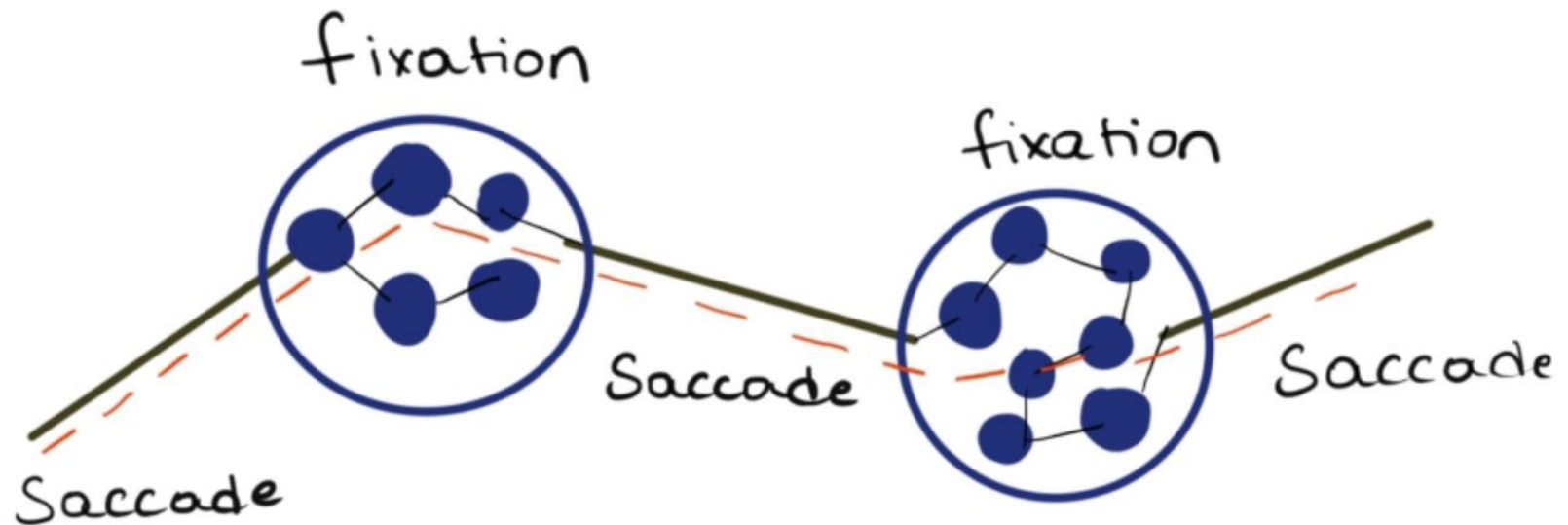
Project no: 2023-1-TR01-KA210-VET-000165855

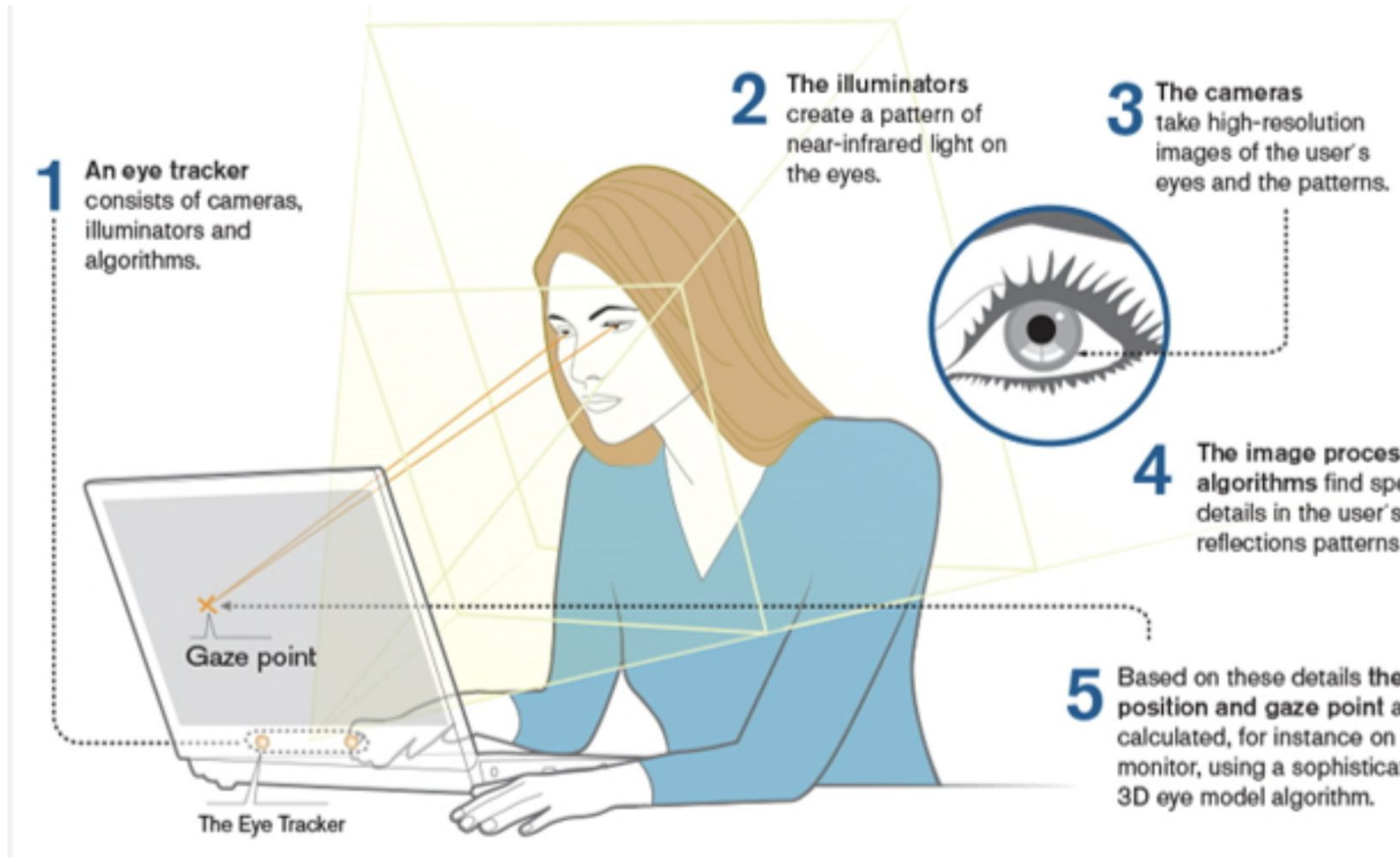
Introduction to eye-movement research with PsychoPy

Fredrik Allenmark

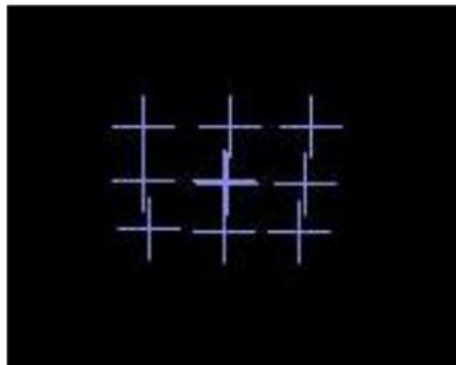


- Saccades are very fast eye-movements between gaze points
- Fixations are periods of relatively slow and small eye-movements (microsaccades)

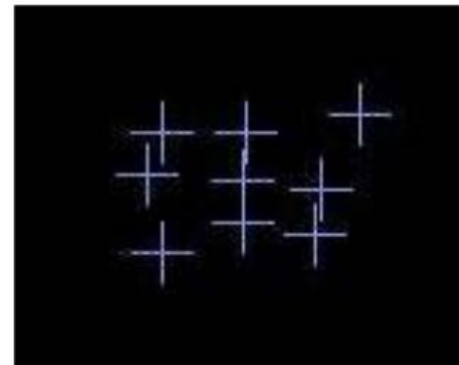




- The algorithms need to be fine-tuned to a particular person and viewing situation
- For this reason the eye-tracker requires calibration at the start of an experiment
- Repeated calibration may be needed (for example after breaks)

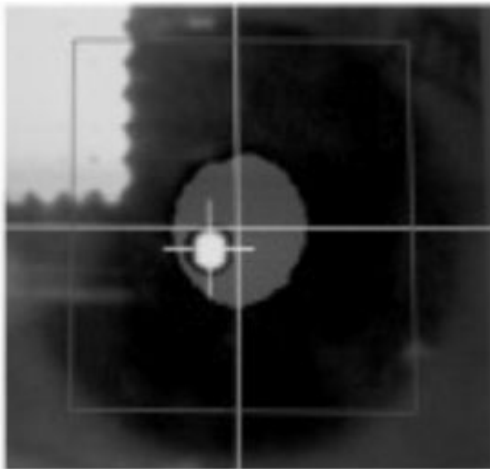


Good Calibration

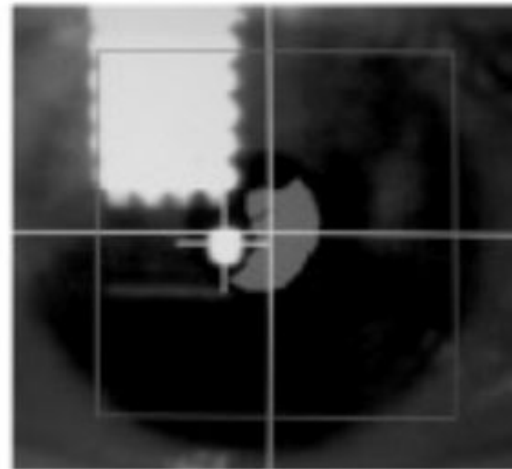


Poor Calibration

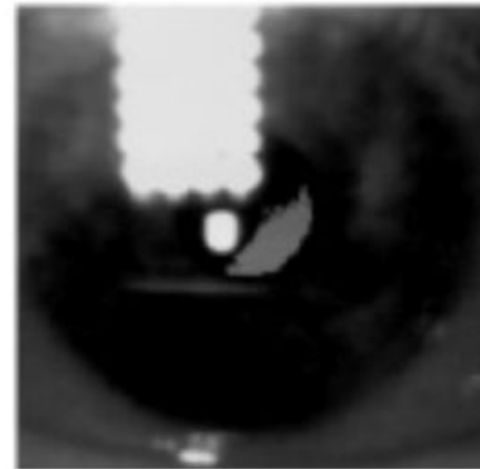
- Eye-glasses make eye-tracking more difficult if reflexes in the glasses cover the pupil
 - Adjusting the angle of the camera can help



(a) Tracking works

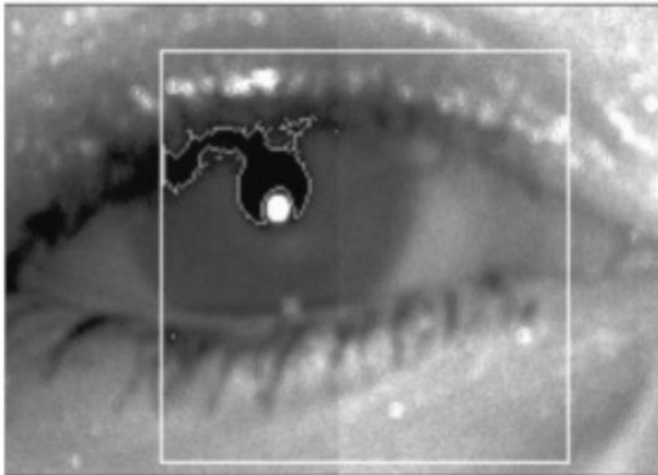


(b) Tracking with off-set

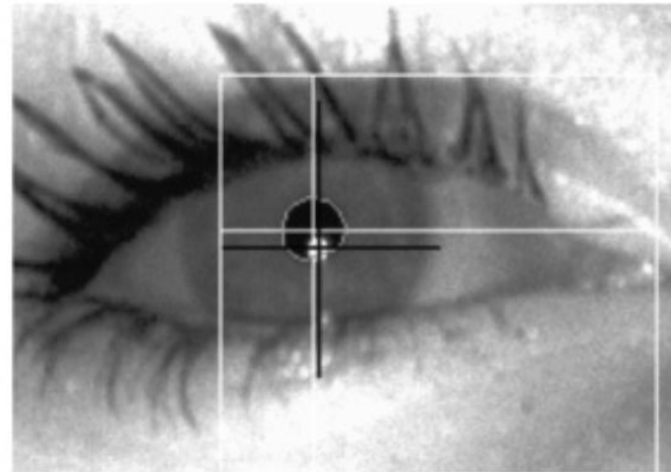


(c) Tracking lost

- Mascara can interfere with detection of the pupil
 - Consider informing participants in advance not to wear mascara on the day of the experiment



(a) Participant with drooping eyelid and downward eyelashes. A thick, dark brush of lashes melds with the pupil and makes it impossible for the recording software to identify the pupil.



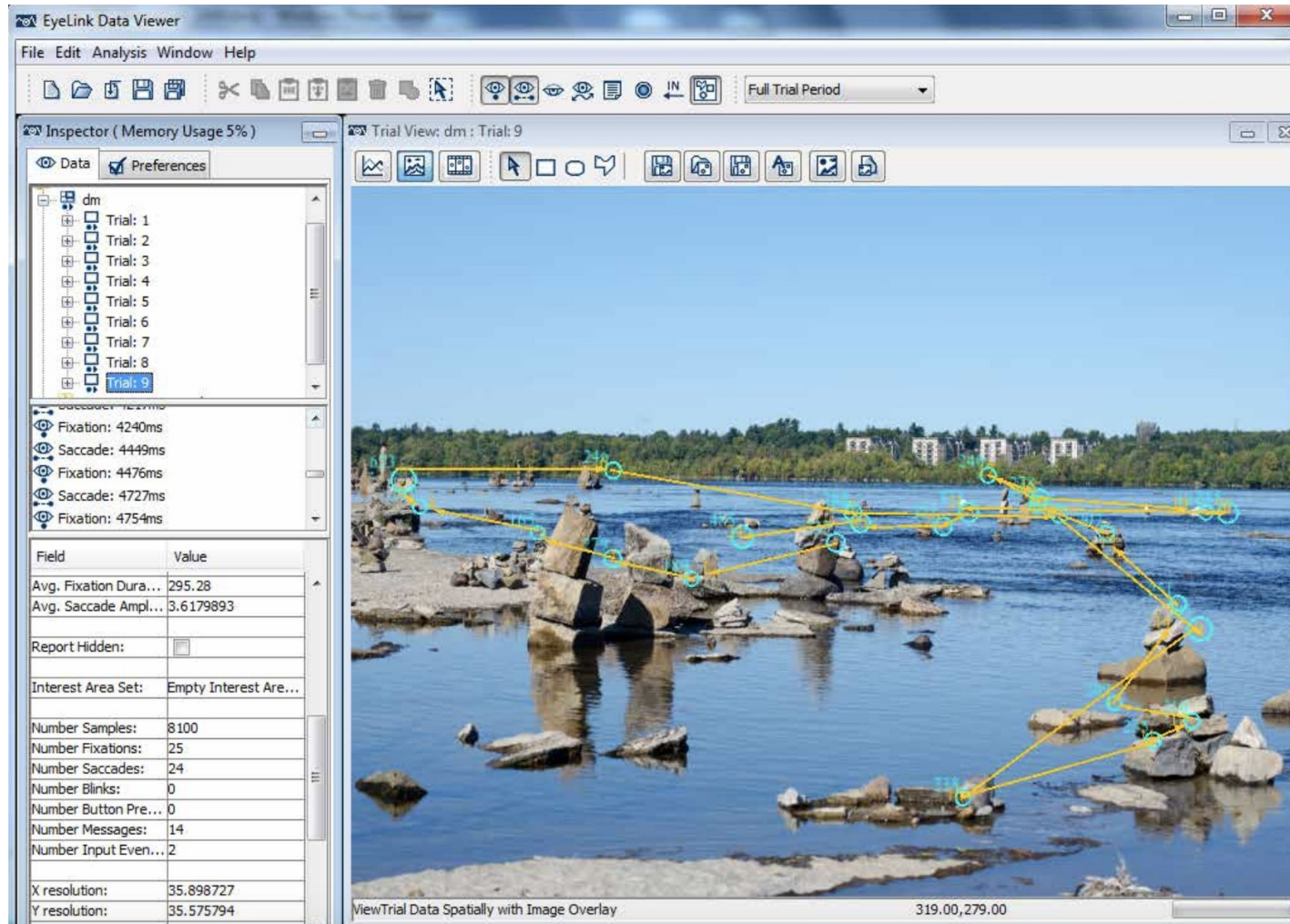
(b) Another participant with mascara, but with upward-pointing lashes; in this case the mascara is easy to exclude, and lashes like these seldom occlude the pupil.

Eyelink tower mount



Eyelink desktop mount

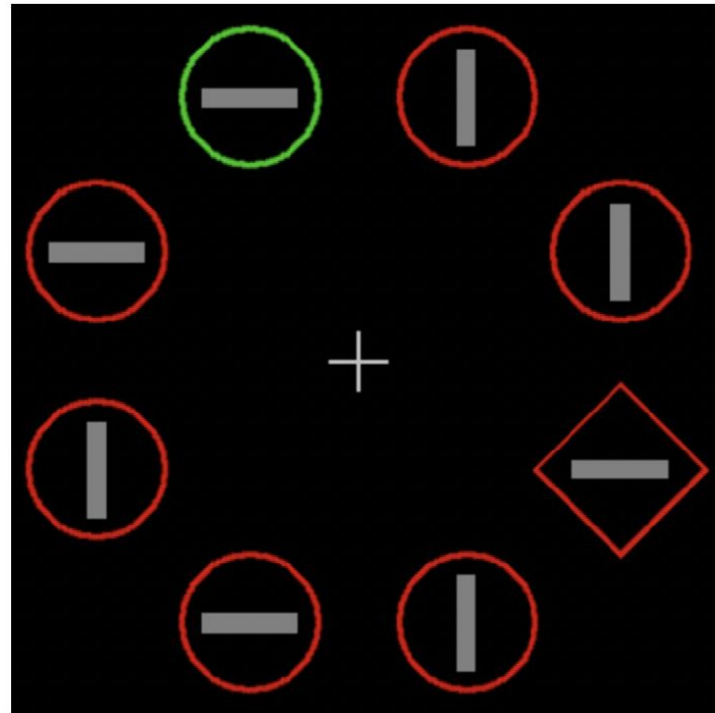




- Scanpaths
- Number/frequency of fixations in an area of interest
- Fixation durations
- Saccadic onset latency
- Pupil size

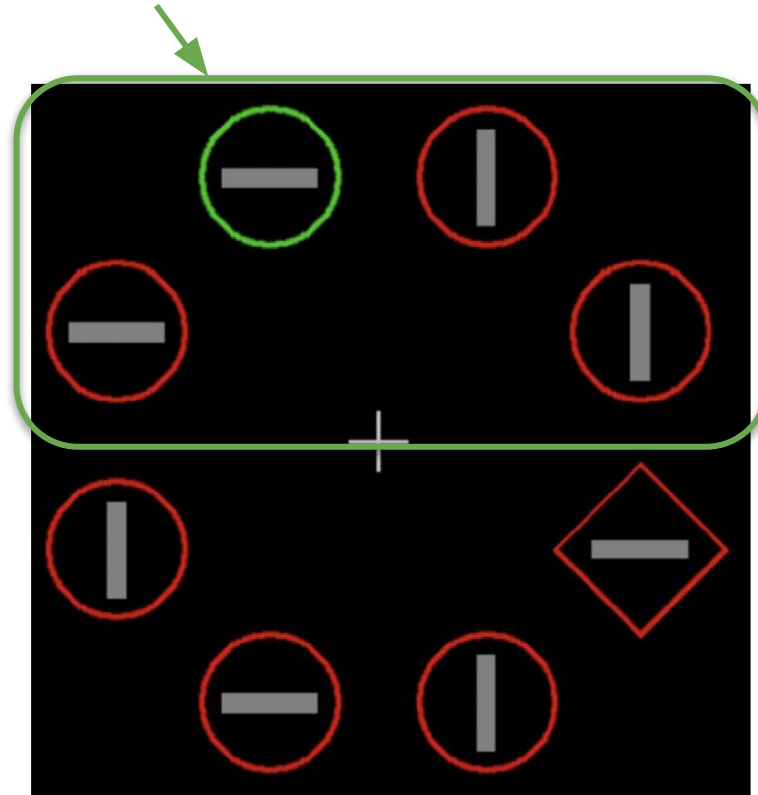
Example: proportion of first fixations

Salient distractor

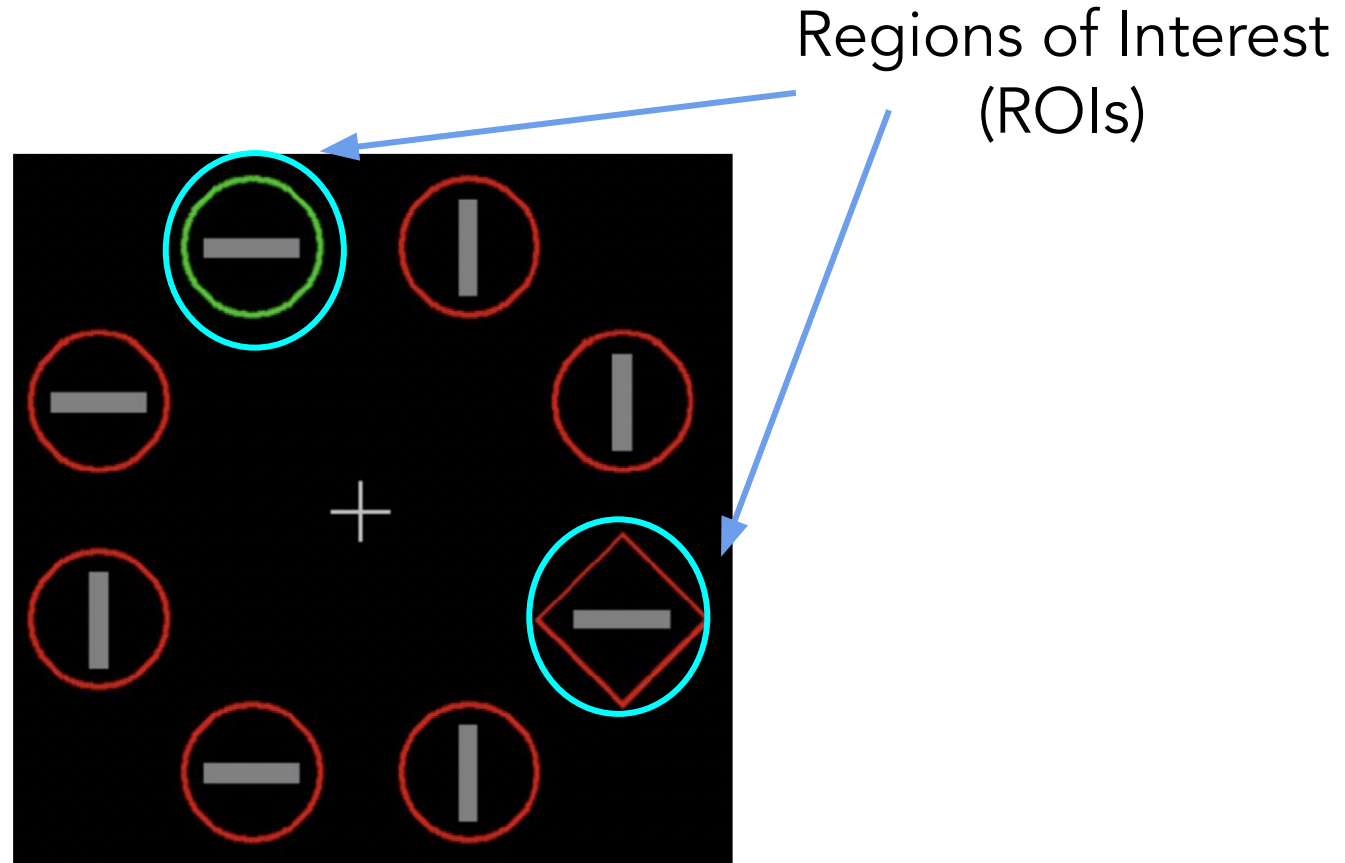


Search target

Distractor occurs more often in one region

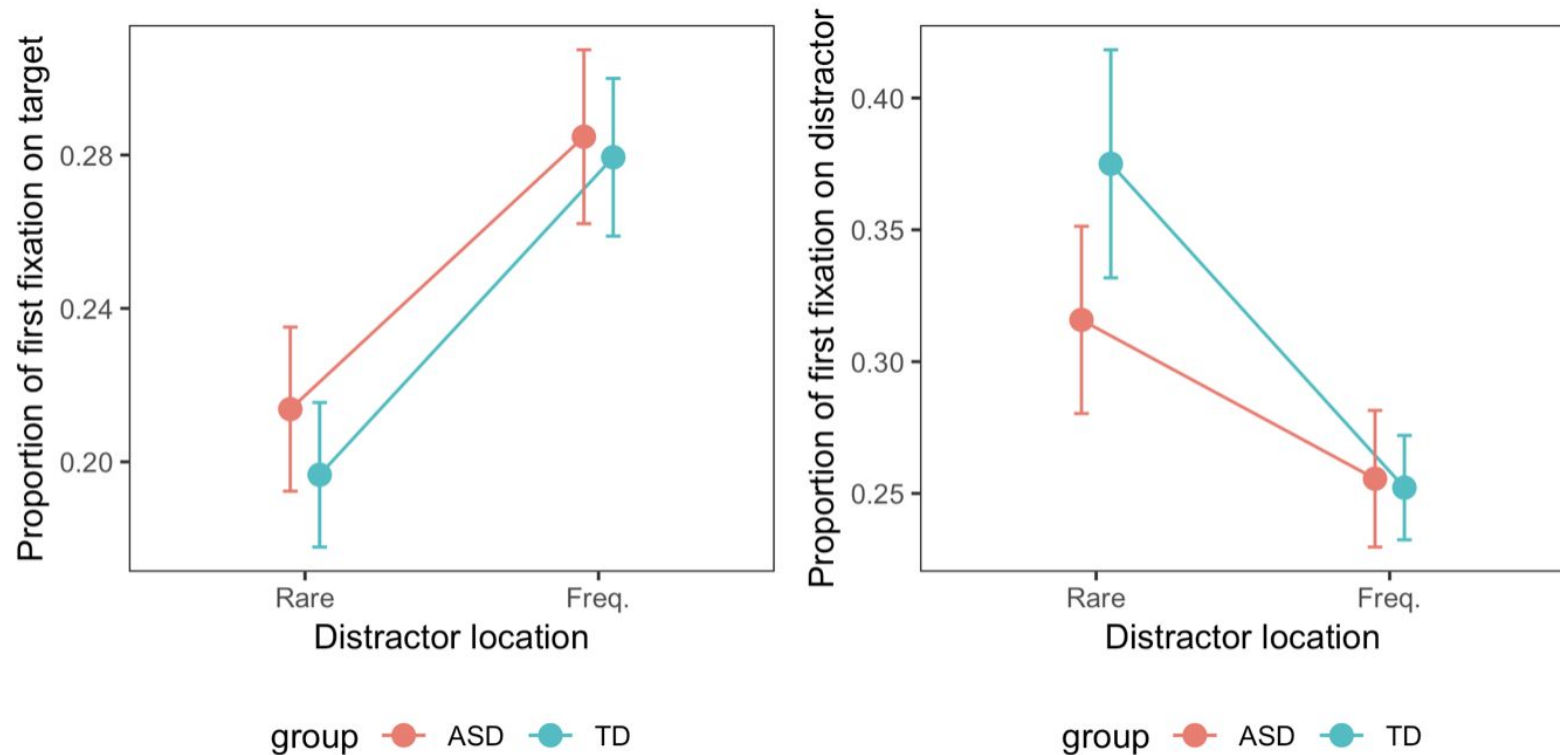


Example: proportion of first fixations



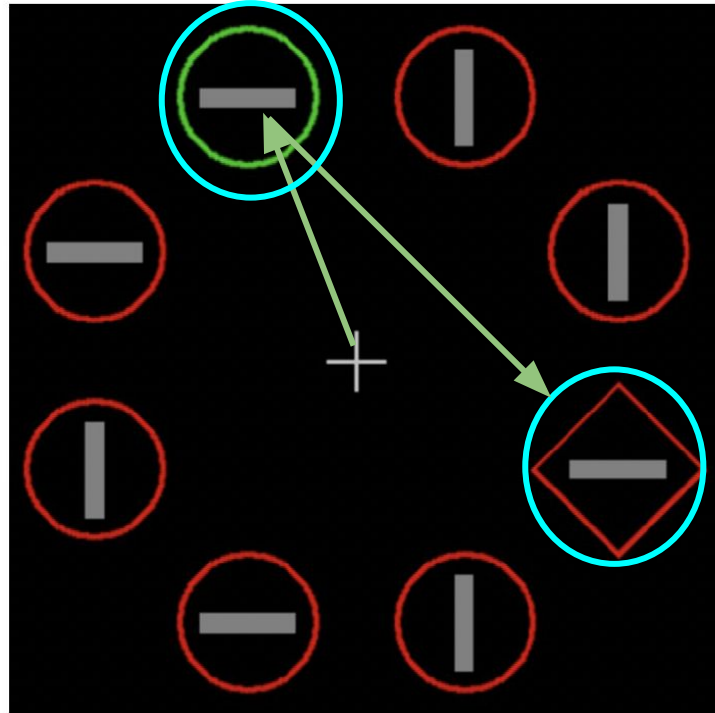
Example: proportion of first fixations

First saccades less frequently landed in the distractor ROI for distractors in the frequent region: evidence for proactive distractor suppression



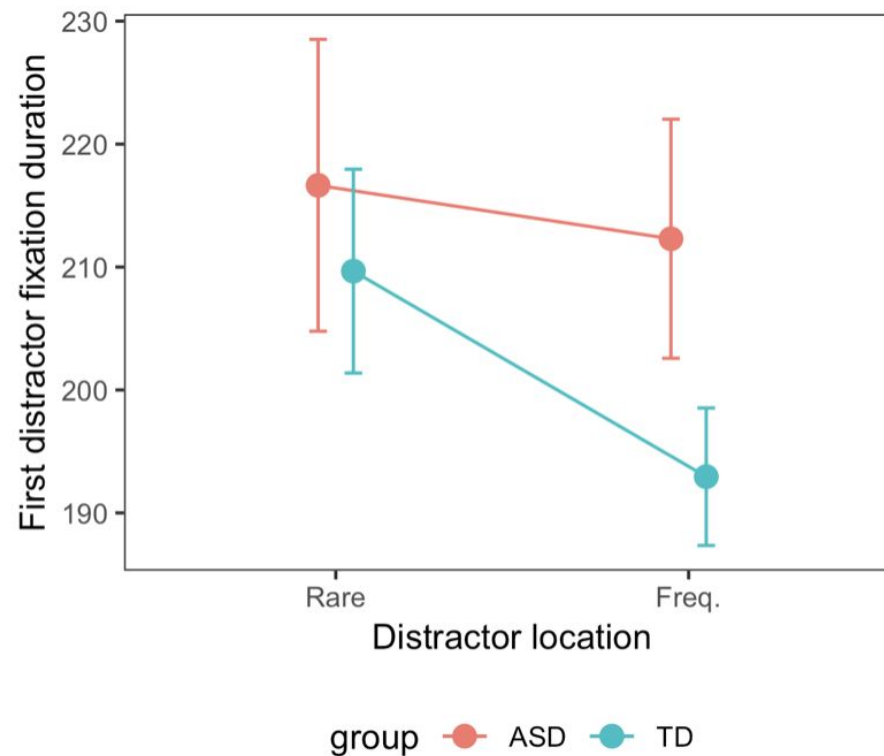
Example: fixation durations

How long does the fixation
in distractor ROI last?

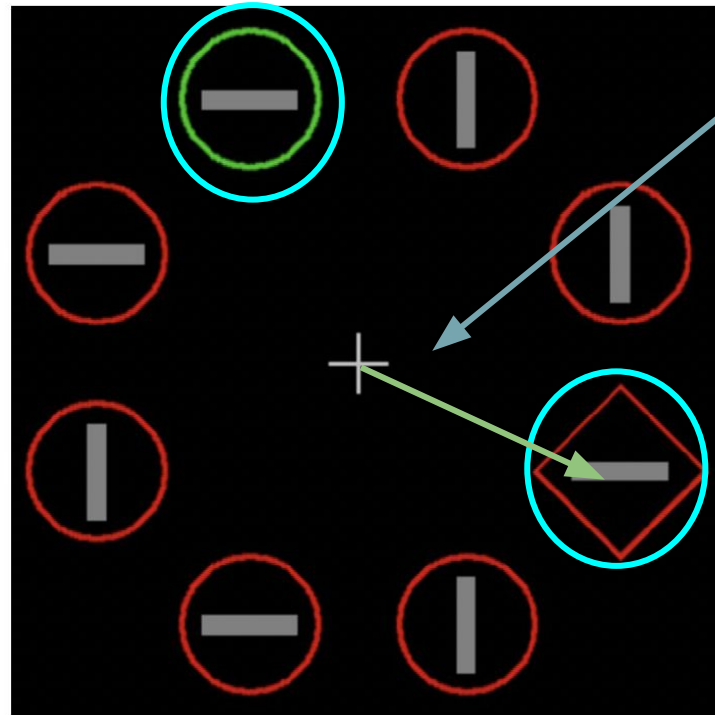


Example: fixation durations

Fixations in the distractor AOI had shorter duration for distractors in the frequent region: evidence for reactive distractor suppression



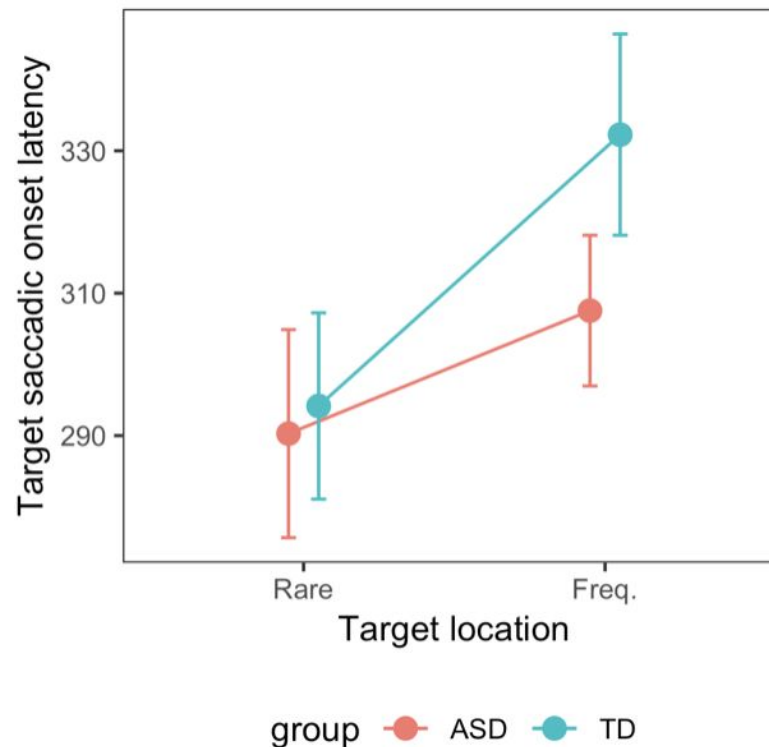
Example: saccadic onset latency



How long time after stimulus appearance does the first saccade start?

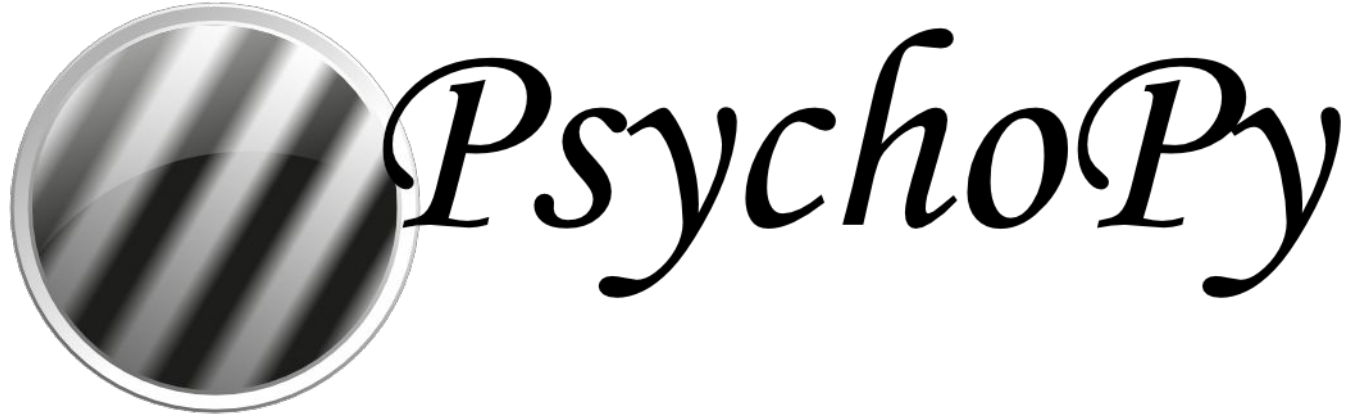
Example: saccadic onset latency

Slower saccades to target when it appeared in the frequent (distractor) region: distractor suppression also affects the target



Most important extra steps needed for an eye-tracker experiment:

- Start and stop recording
- Calibration at start of exp.
(repeat as needed)
- Send messages to eye-tracker at critical time points
(e.g. appearance of stimulus, disappearance of stimulus)
- Optionally define ROIs
(this can also be done later during analysis)



<https://www.psychopy.org/download.html>

- PsychoPy was originally developed in 2002-2003 by visual neuroscientist Jonathan Peirce at University of Nottingham for use in his lab



Jonathan Peirce

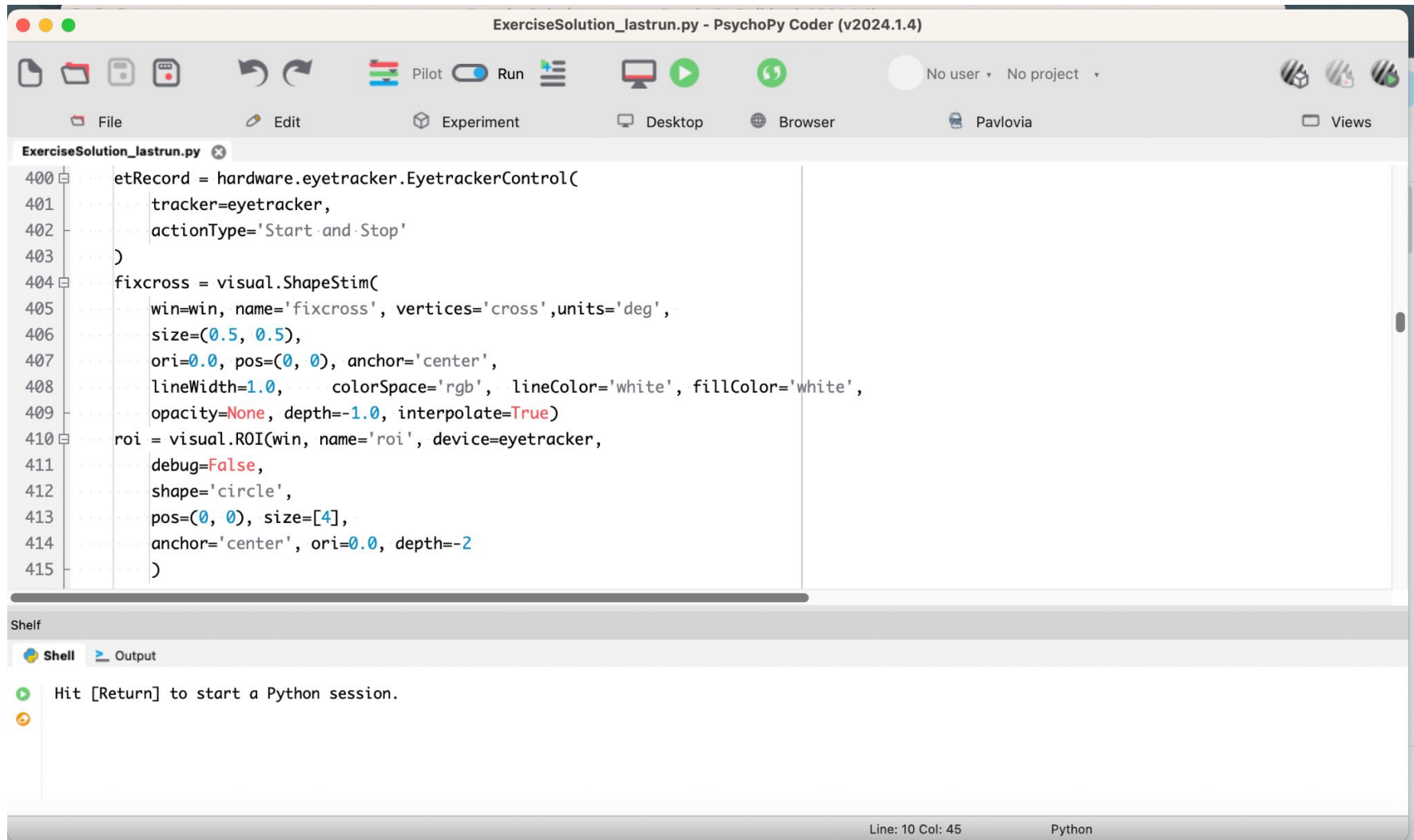
- PsychoPy was originally developed in 2002-2003 by visual neuroscientist Jonathan Peirce at University of Nottingham for use in his lab
- Originally, PsychoPy was a Python library, not an application with a graphical user interface (GUI)

```
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

ExerciseSolution_lastrun.py 9+ x

Users > fredrik > Documents > Documents > Cog science methods course 2023-2024 > PsychoPy exercise > ExerciseSolution > ExerciseSolution_lastrun.py > showExpInfoDlg

341 def run(expInfo, thisExp, win, globalClock=None, thisSession=None):
399     # --- Initialize components for Routine "Fixation" ---
400     etRecord = hardware.eyetracker.EyetrackerControl(
401         tracker=eyetracker,
402         actionType='Start and Stop'
403     )
404     fixcross = visual.ShapeStim(
405         win=win, name='fixcross', vertices='cross', units='deg',
406         size=(0.5, 0.5),
407         ori=0.0, pos=(0, 0), anchor='center',
408         lineWidth=1.0, colorSpace='rgb', lineColor='white', fillColor='white',
409         opacity=None, depth=-1.0, interpolate=True)
410     roi = visual.ROI(win, name='roi', device=eyetracker,
411         debug=False,
412         shape='circle',
413         pos=(0, 0), size=[4],
414         anchor='center', ori=0.0, depth=-2
415     )
416
417     # --- Initialize components for Routine "trial" ---
418     strooptext = visual.TextStim(win=win, name='strooptext',
419         text='',
420         font='Open Sans',
421         pos=(0, 0), height=1.0, wrapWidth=None, ori=0.0,
422         color='white', colorSpace='rgb', opacity=None,
423         languageStyle='LTR',
424         depth=-1.0);
425     key_resp = keyboard.Keyboard(deviceName='key_resp')
426
427     # --- Initialize components for Routine "End" ---
428     text_4 = visual.TextStim(win=win, name='text_4',
429         text='The experiment is finished.\nThank you for participating!',
430         font='Open Sans',
431         pos=(0, 0), height=1.0, wrapWidth=None, ori=0.0,
432         color='white', colorSpace='rgb', opacity=None,
433         languageStyle='LTR',
434         depth=0.0);
435     key_resp_4 = keyboard.Keyboard(deviceName='key_resp_4')
```

The screenshot displays the PsychoPy Code Editor (v2024.1.4) with a file named 'ExerciseSolution_lastrun.py'. The editor features a menu bar (File, Edit, Experiment, Desktop, Browser, Pavlov, Views) and a toolbar with icons for file operations, running, and debugging. The main code area contains the following Python code:

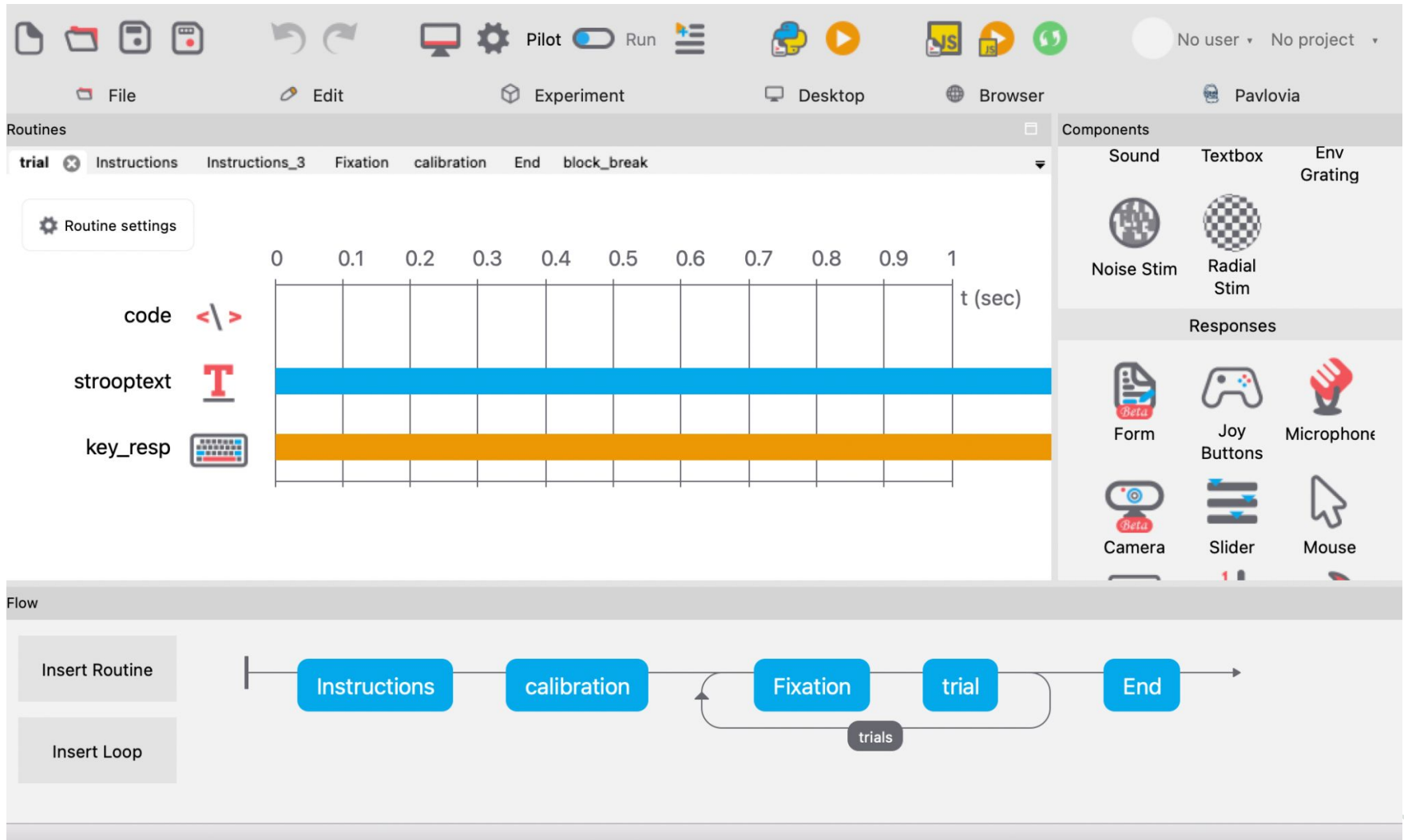
```
400 etRecord = hardware.eyetracker.EyetrackerControl(
401     tracker=eyetracker,
402     actionType='Start and Stop'
403 )
404 fixcross = visual.ShapeStim(
405     win=win, name='fixcross', vertices='cross', units='deg',
406     size=(0.5, 0.5),
407     ori=0.0, pos=(0, 0), anchor='center',
408     lineWidth=1.0, colorSpace='rgb', lineColor='white', fillColor='white',
409     opacity=None, depth=-1.0, interpolate=True)
410 roi = visual.ROI(win, name='roi', device=eyetracker,
411     debug=False,
412     shape='circle',
413     pos=(0, 0), size=[4],
414     anchor='center', ori=0.0, depth=-2
415 )
```

Below the code editor is a 'Shelf' section with tabs for 'Shell' and 'Output'. The 'Shell' tab is active, showing a prompt to start a Python session:

```
Hit [Return] to start a Python session.
```

The status bar at the bottom indicates 'Line: 10 Col: 45' and 'Python'.

- PsychoPy was originally developed in 2002-2003 by visual neuroscientist Jonathan Peirce at University of Nottingham for use in his lab
- Originally, PsychoPy was a Python library, not an application with a graphical user interface (GUI)
- Since then, PsychoPy has become a widely used tool for creating psychological and cognitive neuroscience experiments, which can still be used as a Python library but can also be used with a GUI (“builder view”) with little or no programming



The screenshot displays the Psychopy Builder interface, which is used for creating experimental routines. The interface is divided into several sections:

- Top Bar:** Contains icons for File, Edit, Experiment, Desktop, Browser, and Pavlovian. It also shows the user status (No user) and project status (No project).
- Routines:** A list of routines including trial, Instructions, Instructions_3, Fixation, calibration, End, and block_break. The 'trial' routine is currently selected.
- Components:** A panel on the right side showing various components that can be added to the routine, such as Sound, Textbox, Env Grating, Noise Stim, Radial Stim, Responses, Form, Joy Buttons, Microphone, Camera, Slider, and Mouse.
- Flow:** A diagram at the bottom showing the sequence of routines. The flow starts with 'Instructions', followed by 'calibration', then a loop of 'Fixation' and 'trial' routines, and finally 'End'. The 'trial' routine is highlighted in blue.
- Timeline:** A horizontal timeline at the top of the routine editor, ranging from 0 to 1 second. It shows the duration of each routine: 'Instructions' (0 to 0.1s), 'calibration' (0.1 to 0.2s), 'Fixation' (0.2 to 0.3s), 'trial' (0.3 to 0.9s), and 'block_break' (0.9 to 1.0s).

- We will now build an experiment using PsychoPy builder view
- Please install the latest PsychoPy version (if you did not already)

<https://www.psychopy.org/download.html>