

An introduction to experiment building with OpenSesame

Sebastiaan Mathôt
Joshua Snell

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement n° 622738.

<http://www.cogsci.nl/smathot>

EPOS, Leiden, Sep 24, 2015
For resources, see <http://osdoc.cogsci.nl/epos2015>

Today

- A short introduction (20 min)
- Create an experiment (75 min) ...
- Coffee break
- ... and some more (75 min)

About OpenSesame

About OpenSesame

- A graphical experiment builder
 - Drag-and-drop, point-and-click
 - Complement with Python scripting
- Open source
 - Free of charge
 - Source code available
- Cross platform
 - All major platforms
 - Runtime support for Android

About OpenSesame

- A broad focus
 - Fits many types of research
- Psychophysics
 - Reaction time tasks, complex stimuli, etc.
- Neuroimaging
 - Parallel-port triggers, etc.
- Social psychology
 - Questionnaires, etc.
- Clinical applications
 - Test batteries, mobile (tablet-based) experiments, etc.

- Documentation
 - <http://osdoc.cogsci.nl>
- Community
 - <http://forum.cogsci.nl>
 - 1400+ members, daily activity, very responsive
- Outlook
 - Will OpenSesame still be there in [X] years?
 - Active development team
 - Large user base ($\pm 450,000$ yearly hits for cogsci.nl)

Developers

- A core team
 - Daniel Schreij (VU University Amsterdam)
 - Lotje van der Linden (CNRS / Aix-Marseille)
 - Edwin Dalmaijer (Oxford)
 - Eduard Ort (VU University Amsterdam)
 - Joshua Snell (CNRS / Aix-Marseille)
 - Sebastiaan Mathôt (CNRS / Aix-Marseille)
- Occasional contributors

Teaching

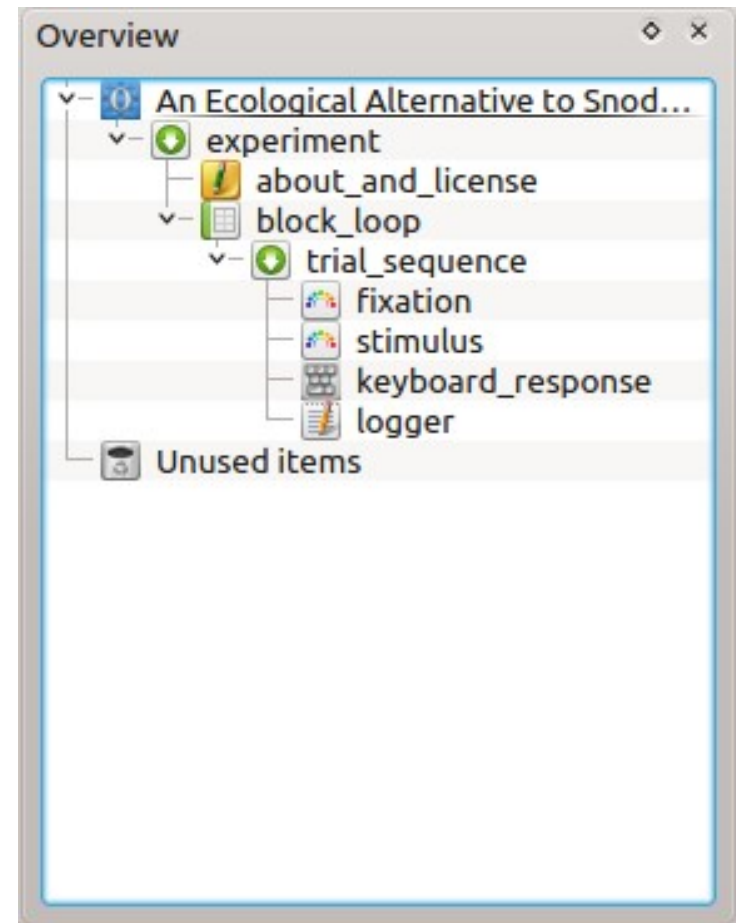
Teaching

- No licensing issues
- No steep learning curve
- Used for teaching at universities across the world
 - Used for many bachelor and master projects
 - Used as part of courses on programming/ research methods

Using OpenSesame

Items

- **Items** are building blocks
- **Ten core items** offer common functionality



Plug-ins

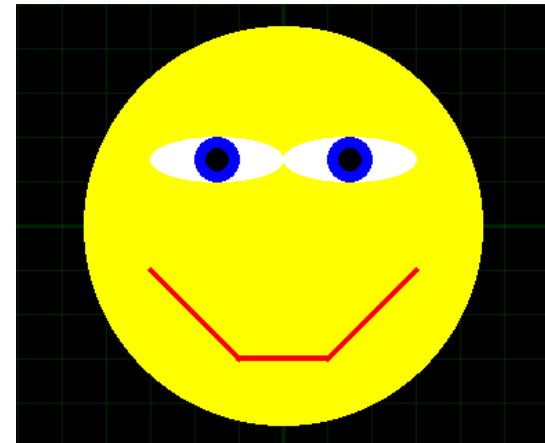


- **Plug-ins** are additional items
 - Eye trackers (PyGaze) (Dalmaijer, Mathôt, Van der Stigchel, 2014)
 - Video playback
 - Forms
 - Etc.
- Plug-ins also provide graphical controls
- New plug-ins can be written easily

User interface vs script

Combining GUI and script

- The GUI generates a script
 - Custom language
 - Not Python!
- You can edit this script directly
- Afterwards you can continue using the GUI



Script (001, 015)

```
1 set duration "keypress"
2 set start_response_interval "no"
3 set description "Displays stimuli"
4 draw circle 0 0 288 fill=1 penwidth=1 color=yellow show_if="always"
5 draw line -96 32 -32 96 penwidth=4 color=red show_if="always"
6 draw line 32 96 96 32 penwidth=4 color=red show_if="always"
7 draw line -32 96 32 96 penwidth=4 color=red show_if="always"
8 draw ellipse -96 -64 96 32 fill=1 penwidth=1 color=white show_if="always"
9 draw ellipse 0 -64 96 32 fill=1 penwidth=1 color=white show_if="always"
10 draw ellipse -64 -64 32 32 fill=1 penwidth=1 color=blue show_if="always"
11 draw ellipse 32 -64 32 32 fill=1 penwidth=1 color=blue show_if="always"
12 draw fixdot -48 -48 color=black show_if="always"
13 draw fixdot 48 -48 color=black show_if="always"
```

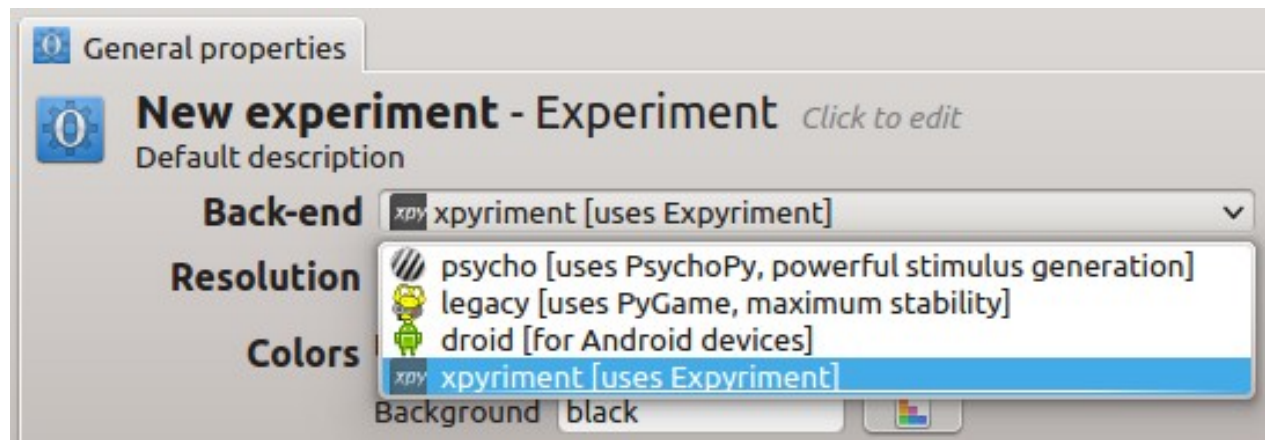
Combining GUI and script

- You can create a prototype display using the GUI, and add variables using scripting
- Prototype script:
 - draw image 0.0 0.0 "gaze_**left**.png" scale=1.0 center=1 show_if="always"
- Variable script:
 - draw image 0.0 0.0 "gaze_**[gaze_cue]**.png" scale=1.0 center=1 show_if="always"

Back-ends

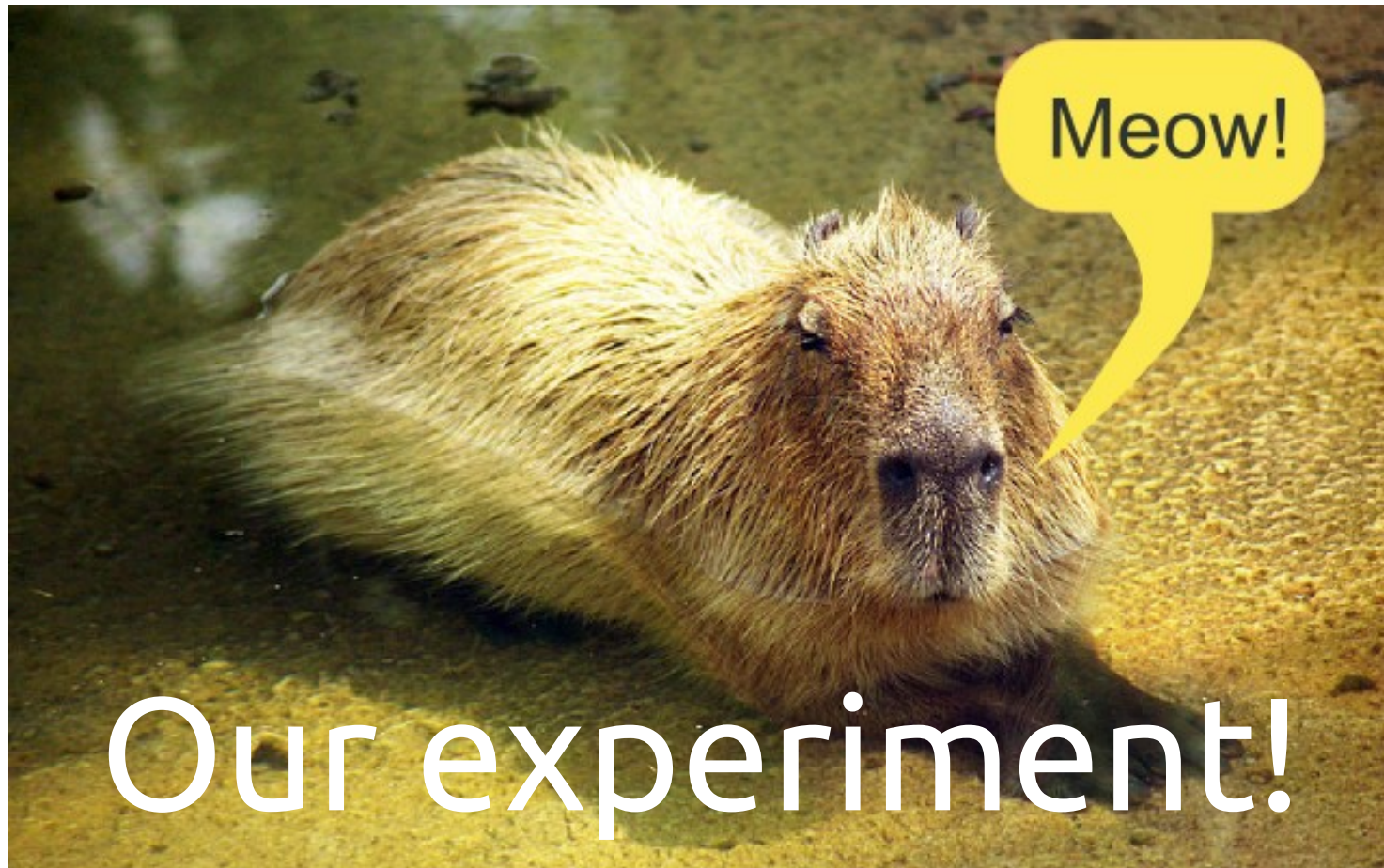
Back-ends

- There are many ways to control the display, input, etc.
- OpenSesame is not tied to one method
- **Back-ends** can be flexibly added, like plug-ins



Back-ends

- Each back-end has its own benefits
 - Temporal precision
 - Stability
 - Extra functionality
 - Cross-platform support
- **Xpyriment** → Simple with good temporal precision. Expyriment-based (Krause & Lindeman, 2013)
- **Legacy** → Fallback, modest temporal precision
- **Psycho** → PsychoPy based, good temporal precision (Peirce, 2007)
- **Droid** → For Android devices



- Multisensory cueing task
- Participants see:
 - cat
 - dog
 - capybara
- ... and hear:
 - meow
 - bark

- Task:
 - dog → tap left
 - cat → tap right
 - capybara → no response
- Hypotheses:
 - Congruent sound and image (cat + meow, or dog + bark) → fast response
 - Incongruent sound and image (cat + bark, or dog + meow) → slow response
 - When there is a capybara, most false responses are congruent with the sound

- Formally: $S_N \times V_3 \times A_2$
- What kind of design is this?
 - Within-subjects or between-subjects?
 - Fully crossed?
 - How many factors with how many levels?

And now for the tutorial!

References

- Dalmaijer, E., Mathôt, S., & Van der Stigchel, S. (2014). PyGaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments. *Behavior Research Methods*, 46(4), 913–921. [doi:10.3758/s13428-013-0422-2](https://doi.org/10.3758/s13428-013-0422-2)
- Krause, F., & Lindemann, O. (2013). Expyriment: A Python library for cognitive and neuroscientific experiments. *Behavior Research Methods*. [doi:10.3758/s13428-013-0390-6](https://doi.org/10.3758/s13428-013-0390-6)
- Mathôt, S., Schreij, D., & Theeuwes, J. (2012). OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods*, 44(2), 314–324. [doi:10.3758/s13428-011-0168-7](https://doi.org/10.3758/s13428-011-0168-7)
- Peirce, J. W. (2007). PsychoPy: Psychophysics software in Python. *Journal of Neuroscience Methods*, 162(1-2), 8–13. [doi:10.1016/j.jneumeth.2006.11.017](https://doi.org/10.1016/j.jneumeth.2006.11.017)