OpenSesame workshop 2019

Matt Green — mgreen@bournemouth.ac.uk

Nov 6th 5pm to 7pm & Nov 7th 9am to 11am in P105

This handout, and all the example experiments, can be downloaded from: `https://github.com/mjgreen/opensesame-workshop`. If you are reading this somewhere other than the workshop, where the program is available on the uni computers, you can download OpenSesame for your own machine from: `https://osdoc.cogsci.nl/3.2/download/`. Official video tutorials are available at: `https://www.youtube.com/sebastiaanmathot`. If you go on to use OpenSesame, please remember to cite it[1]. There is an active forum at: `https://forum.cogsci.nl/`.

Downloads; Links

[1] Sebastiaan Mathôt, Daniel Schreij, and Jan Theeuwes. Opensesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods*, 44(2):314–324, Jun 2012

Contents

Getting Started

1. **Download the workshop's supporting files:** these are all the examples, and a copy of this handout: go to: `https://github.com/mjgreen/opensesame-workshop`. Look for the green button that says "Clone or Download". When you click it, it will offer you "Download ZIP": click on that, and when the zip file has downloaded, extract it somewhere you will be able to find (maybe the Desktop, or a folder in your home drive). If you don't know how to extract a zip file, right-click on the zip file and choose "extract

all", then browse to the folder you want to use. It's important to get these files before we start, so if you need help, interrupt me and I will come and help you.

2. **Open up the OpenSesame program:** In P105, it is part of *AppsAnywhere*. Be patient and only click the link once, otherwise you will get multiple instances of the program open up, which can be confusing. Again, stop me and get help if you have any trouble doing that, so that everyone has the program open before we go any further.

The simplest case of an OpenSesame experiment

Every OpenSesame experiment is organised around a few core notions: **stimulus**, **instructions**, and **response**.

Problem specification

Let's imagine that we want to know whether the word *war* is associated with positive or negative emotions.

Solution in plain English

We need to present the word (in this example the word *war*) for some length of time (in this example it is 5 seconds); then instruct the participant what keys they can press (in this example it is *n* for negative, or *p* for positive); and then collect a response from the keyboard, only allowing *n* for negative, or *p* for positive).

Solution as flowchart

OpenSesame experiments can be visualised as flowcharts. Figure (1) is the flowchart for this particular example, the simplest case of a stimulus-response experiment.

Solution as OpenSesame experiment

Go to your OpenSesame window, and do `File -> Open`, then browse to the folder where you extracted the suporting materials, and choose the following file: `example-01-the-simplest-case.osexp`. Figure (2) shows what you will see when you open up the experiment, if you click on "experiment" in the list-view on the left-hand side. There are several different views you can get of an osexp[2]: notice that in this view, the experiment is represented in a very similar way to the flowchart representation in Figure (1).

[2] **.osexp** is the file extension for OpenSesame experiments, in the same way that **.jpeg** is the file extension for jpeg photos

Even in this very simple experiment, we can learn a lot about the way OpenSesame works by going through the exercises below.

1. The first thing for you to do is to run the experiment. In OpenSesame there are several different ways to run an experiment. In the top-left of the interface there are three buttons with arrows
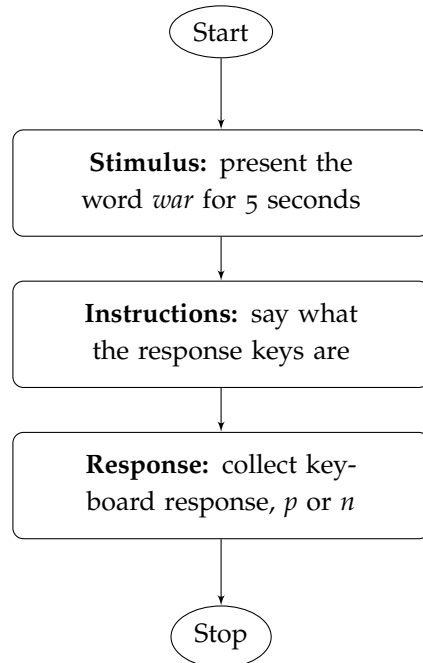
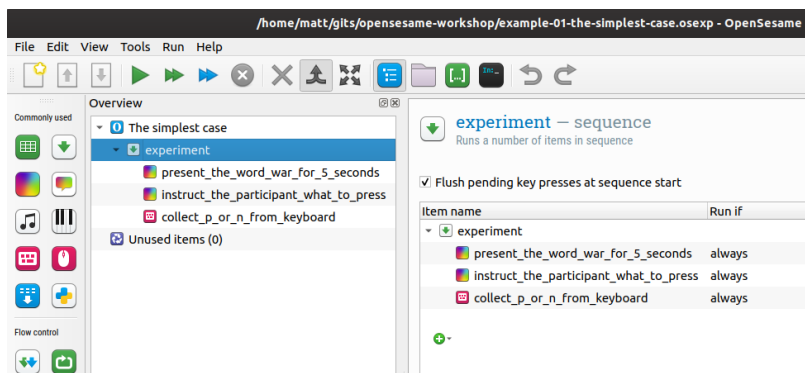Figure 1: Simplest stimulus-response experiment, presented as a flowchart

Figure 2: example 01 screenshot

facing right – each one runs the experiment in a different way, as described next:

(a) The big green single arrow is what you use to run an experiment while you are collecting real data with a real participant. In this mode, the experiment will ask you for a participant number, and the filename for the results file; it will run full-screen; and it will write out a results file at the end. While you are still developing the experiment, like we are now, you should use one of the double-arrows:

(b) The green double-arrow does the same as the single green arrow, except that instead of running full-screen, it runs in a window about half the size of the screen. This is handy because if there is an error, you will have a better chance of seeing why it went wrong.

(c) The blue double-arrow also runs in a smaller window, but it differs in that it doesn't prompt you for a participant number, and doesn't write a results file. The blue double-arrow is a sort of "quick-run" option.

Use the blue double-arrow to run the experiment now. You are aiming to verify that the experiment runs without throwing any errors; and to check that it does what you expect: i.e., it should show the word "war" for 5 seconds; then tell you what the response options are; and then wait for you to press either "p" or "n", but refuse to proceed if you press some other key.

Excercise 1: Test-run the experiment

2. The next thing is to explore the experiment to find out how each of the three parts of the flow works. When you click on an item in the *list-view* (or *flow-chart* if you prefer that term) on the left hand side of the interface, the right-hand-side will show a detailed view of what that item does, and what options you can set for that item. You are aiming to answer the following questions:

Excercise 2: Explore the experiment

(a) How do we specify that "war" is the text that gets displayed?

(b) How do we specify that "war" should be displayed for 5 seconds (rather than some other duration)?

(c) How do we make sure that the experiment will not accept any keyboard responses other than "p" or "n"?

(d) How do we make the experiment wait for as long as it takes the participant to respond, instead of timing out after some fixed interval of time?

More realistic cases: multiple trials using sequences and loops

sequencing

looping

Stimulus types available

Response types available

Feedback to participants

Eye-tracking (on any tracker)

Counterbalancing, and other uses of python)

Cross-platform support

Some comparisons of OpenSesame against alternative software