OpenSesame Tutorial

Creating a Gaze Cuing Experiment

Dec 2010

Sebastiaan Mathôt

Dept. Cognitive Psychology

Vrije Universiteit, Amsterdam

s.mathot@psy.vu.nl

In this tutorial you will learn how to create a simple, but complete psychological experiment using OpenSesame (Mathôt & Theeuwes, 2010). You will only use the OpenSesame graphical interface (i.e., no Python inline coding), although you will make small modifications to the OpenSesame script.

This document provides a highly detailed 'click-by-click' walk-through. Screenshots are provided as well as downloadable 'snapshots' of the experiment as it should be after each step of the tutorial.

About OpenSesame

OpenSesame is a cross-platform graphical experiment builder. It provides an exceedingly simple way to create psychological experiments using a point-and-click graphical interface. For more advanced tasks, you can make use of Python inline coding (not covered in this tutorial).

OpenSesame is freely available under the General Public License v3.

- **Note** — This tutorial assumes that you are running OpenSesame version 0.17 or newer. You can download the most recent version of OpenSesame here:

  http://wwww.cogsci.nl/opensesame

The experiment

The experiment that you will create is a gaze-cuing paradigm, as has been introduced by Friesen and Kingstone (1998). A face is presented at the center of the screen (Figure 1). This face looks either to the right or to the left. A target stimulus (a letter) is presented on the left or the right side of the face. A distractor stimulus (the letter 'X') is presented on the other side of the face. The task is to indicate as fast as possible whether the target letter was an 'F' or an 'H'. In the congruent condition, the face looks at the target. In the incongruent condition, the face looks at the distractor. As you may have guessed, the typical finding is that you are faster in the congruent condition, relative to the incongruent condition, even though the direction of gaze is not predictive of the target location. This shows that our attention is automatically guided by other people's gaze, even if this is useless. Of course, following other people's gaze is generally not useless at all, so this is actually a good strategy.

The experiment will consist of a practice and an experimental phase. We will present visual feedback after every block of trials and play a sound after every incorrect response.

Figure 1. The gaze cuing paradigm.

Step 1: Create the main sequence

When you start OpenSesame, you will see something like Figure 2.

By default there is a main sequence, simply called 'experiment'. Click on 'experiment' in the overview (by default on the left side) to open its tab. The 'experiment' sequence consists of a single item: a sketchpad called 'welcome'. We will keep this sketchpad and use it to present instructions (see step 12). But of course we need more than only a single sketchpad.

Append a new loop item, containing a new sequence item, for the practice phase — We need to append a new loop item to the main sequence. We will use this loop as the practice phase of the experiment. Select 'loop' in the menu after 'Append new item' and click on the button with the 'plus' icon. You will be asked to fill the new loop item with another item. This is because a loop by itself does nothing. A loop always needs another item to run. Choose a 'sequence' item in the menu under 'Create new item to use' and click on the 'Create' button.

- **What is a loop item?** — A loop is an item that adds structure to your experiment. It repeatedly runs another item, typically a sequence. You can also define variables (factors) in a loop.

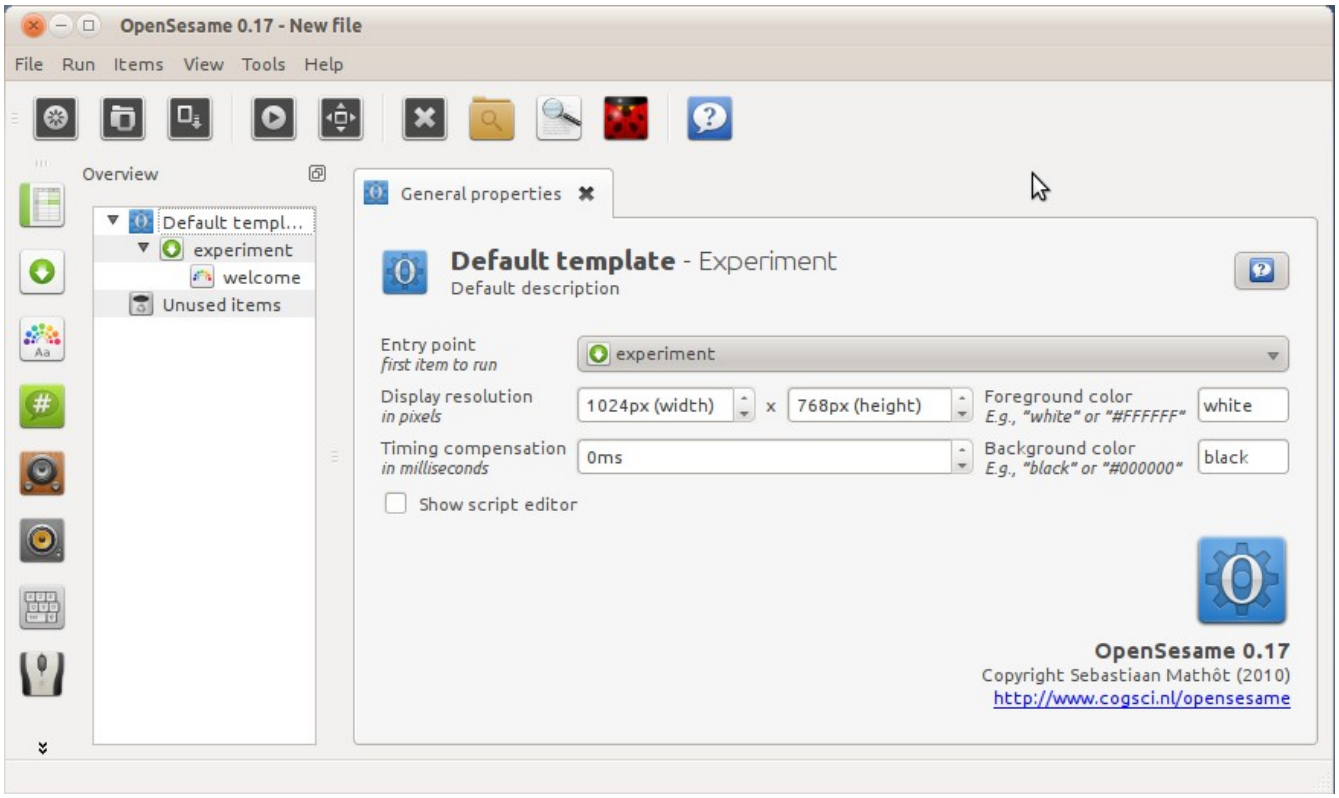


Figure 2. The OpenSesame window upon startup.

- **What is a sequence item?** — A sequence item also adds structure to your experiment. A sequence runs multiple other items in sequence.
- **The loop-sequence structure** — You will often want to repeat a sequence of events. To do this, you will need a loop item which contains a sequence item. By itself, a sequence does not repeat itself. It simply starts with the first item from the sequence and ends with the last item. By 'wrapping' a loop item around the sequence, you can repeat the sequence multiple times. For example, a single trial will usually be a single 'trial' sequence. A 'block' loop around this 'trial' sequence would then constitute a single block of trials. Similarly, but at another level of the experiment, a 'block' sequence may contain a single block of trials, followed by a participant feedback display. A 'practice_phase' loop around this 'block' sequence would then constitute the practice phase of the experiment. This may seem a bit abstract right now, but as you follow this tutorial, you will become familiar with the use of loops and sequences.

Append a new sketchpad item, for the end-of-practice message — After the practice phase, we want to inform the participant that the real experiment will begin. For this we need a sketchpad. Select 'sketchpad' in the menu marked 'Append new item' and click on the 'Plus' button.

- **What is a sketchpad item?** — As the name suggests, a sketchpad is used to present visual stimuli. This includes text, geometric shapes, fixation dots, Gabor patches, etc. You can draw on the sketchpad using the built-in drawing tools (see step 5).

Append a new loop item, containing the previously created sequence, for the experimental phase — We need a 'loop' item for the experimental phase, just like for the practice phase. Therefore, select 'loop' from the 'Append new item' menu and click on the 'plus' button. However, because the trials of the practice phase and the experimental phase are identical, they can use the same sequence. Therefore, in the dialog that appears after you have created a new loop item, select the previously created sequence (simply called 'sequence') from the menu below 'Select existing item to use' and click on the 'Select' button.

Append a new sketchpad item, for the goodbye message — When the experiment is finished, we should inform the participant (see step 12). For this we need another sketchpad item. Select 'sketchpad' in the 'Append new item' menu and click on the 'plus' button.

Give the new items sensible names — By default, new items have names like 'sequence' and '_sketchpad'. It is good practice to give items sensible names. This makes it much easier to understand the structure of the experiment. If you want, you can also add a description to each item. Renaming items is very easy. Item names must consist of alphanumeric characters and underscores.

- Click on 'welcome' in the overview. Click on the 'welcome – Sketchpad' text at the top of the tab that you have just opened. Rename the item to 'instructions'.
- Click on 'loop' in the overview and rename it to 'practice_loop'.
- Click on 'sequence' (under 'practice_loop') in the overview and rename it to 'block_sequence'.
- Rename 'sketchpad' to 'end_of_practice'.
- Rename '_loop' to 'experimental_loop'.
- Rename '_sketchpad' to 'end_of_experiment'.

Give the whole experiment a sensible name. The experiment in its entirety also has a name (a title, in this context) and a description. Click on 'Default template' in the overview and rename the experiment to 'Tutorial 2: Gaze cuing'. Note that, unlike item names, the experiment title may contain spaces etc.

- **Tip** — If you don't like having many tabs open, you can close all tabs except the currently opened one by clicking on the 'Close other tabs' button in the toolbar or by pressing Control + T.

The overview of your experiment now looks like Figure 3. You can download the experiment up to this point here: step1.opensesame.tar.gz

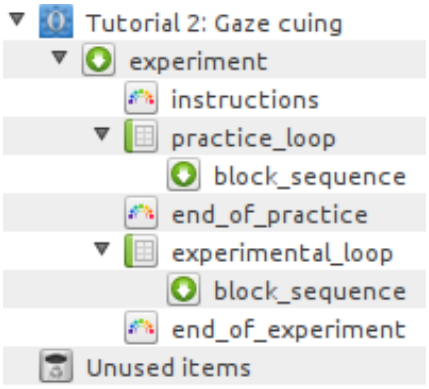


Figure 3. The overview after step 1

Step 2: Create the block sequence

Click on 'block_sequence' in the overview. At the moment the sequence is empty. We want the block sequence to consist of a block of trials, followed by a participant feedback display. For this we need to do the following:

Append a new loop, containing a new sequence, for a block of trials — For a single trial we need a sequence. For a block of trials, we need to repeat this sequence multiple times. Therefore, for a block of trials we need to wrap a loop around a sequence. Select 'loop' in the 'Append new item' menu and click on the 'plus' button. In the dialog that appears, select 'sequence' under 'Create new item to use' and click on the 'Create' button.

Append a feedback item — After every block of trials we want to give feedback to the participant, so that the participant knows how well he/ she is doing. For this we need a feedback item. Select 'feedback' in the 'Append new item' menu and click on the 'plus' button.

- **What is a feedback item?** — A feedback item is almost the same as a sketchpad item. The only difference is that a feedback item is not prepared in advance. This means that you can use it to present feedback, which requires up-to-date information about a participant's response. You should not use feedback items to present time critical displays, since the fact that it is prepared 'at runtime' means that its timing properties are not as good as that of the sketchpad item.

Give the new items sensible names — Rename 'loop' to 'block_loop' and 'sequence' to 'trial_sequence'. (See step 1 if you don't remember how to do this.) The name 'feedback' is fine, so you don't need to rename it.

The overview of your experiment now looks like Figure 4. You can download the experiment up to this point here: step2.opensesame.tar.gz

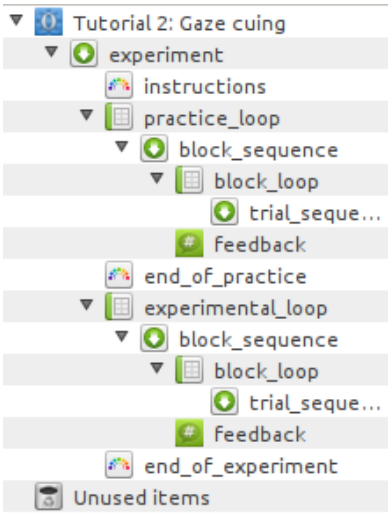


Figure 4. The overview after step 2

Step 3: Fill the block loop with independent variables

As the name suggests, the block loop corresponds to a single block of trials. In the previous step we created the block loop, but we still need to 'fill' it with independent variables, which will be varied within the block. Our experiment has three independent variables:

- **gaze_cue** can be 'gaze_left.png' or 'gaze_right.png'. Note that we use the full file names of the images that we are going to use, rather than simply 'left' and 'right'. This is convenient when we are going to create the cue display (see step 5).
- **target_pos** (the position of the target) can be '-300' or '300'. These values reflect the X-coordinate of the target in pixels (0 = center). Again, using the coordinates directly, rather than 'left' and 'right', is convenient when we are going to create the target displays (see step 5).
- **target_letter** (the target letter) can be 'F' or 'H'.

Our experiment has 2 x 2 x 2 = 8 levels. Although 8 is not that many (most experiments will have more levels), we don't need to type in all the possible combinations of variables by hand. Click on 'block_loop' in the overview to open its tab. Now click on the 'Variable wizard' button. In the variable wizard, you simply define all variables by typing the name in the first row and the levels in the rows below the name (see Figure 5). If you select 'Ok', you will see that the block loop has been filled with all 8 possible combinations. That's convenient!

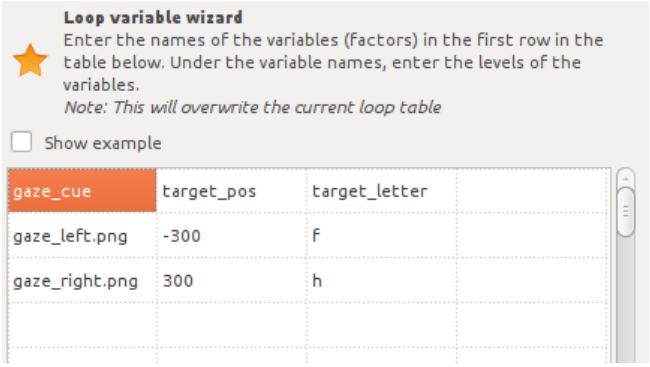


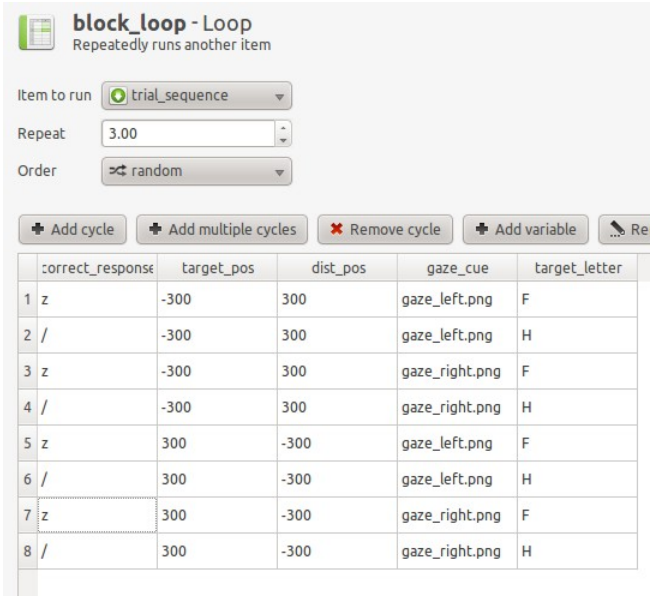
Figure 5. The loop variable wizard in step 3.

However, we are not done yet. There are two more variables that we need to add: the location of the distractor and the correct response.

- **dist_pos** Click on 'Add variable' and type 'dist_pos 300' in the dialog that appears. This will create a variable called 'dist_pos', with the value '300'. Change the value of 'dist_pos' to '-300' in every row where 'target_pos' is '300'.
- **correct_response** Click on 'Add variable' and type 'correct_response z' in the dialog that appears. Change 'correct_response' to '/' where 'target_letter' is 'H'. This means that the participant should press the 'z' key if he/ she sees an 'F' and the '/' key if he/ she sees an 'H'.

There is one last thing to be done. 'Repeat' is currently set to '1'. This means that each level will be executed once. So the block now consists of 8 trials, which is a bit short. A reasonable length for a block of trials is 24, so set 'Repeat' to 3 (3 repeats x 8 levels = 24 trials). You don't need to change 'Order', because 'random' is exactly what we want.

The block_loop now looks like Figure 6. You can download the experiment up to this point here: step3.opensesame.tar.gz



	correct_response	target_pos	dist_pos	gaze_cue	target_letter
1	z	-300	300	gaze_left.png	F
2	/	-300	300	gaze_left.png	H
3	z	-300	300	gaze_right.png	F
4	/	-300	300	gaze_right.png	H
5	z	300	-300	gaze_left.png	F
6	/	300	-300	gaze_left.png	H
7	z	300	-300	gaze_right.png	F
8	/	300	-300	gaze_right.png	H

Figure 6. The block loop after step 3,

Step 4: Add images and sound files to the file pool

We will not actually draw the stimuli in this experiment, but use images from file. In addition, we will play a sound if participants make an error. For this we need a sound file. You can download the required files here:

- gaze_neutral.png
- gaze_left.png
- gaze_right.png
- incorrect.ogg

After you have downloaded these files (to your desktop, say), click on the 'Show file pool' button in the OpenSesame toolbar (or: Menu → View → Show file pool). This will show the file pool, by default on the right side of the OpenSesame window. The easiest way to add the four files to the file pool is to drag them from the desktop (or wherever you have downloaded the files to) into the file pool. Alternatively, you can click on the 'plus' button in the file pool and add files using the file select dialog that appears. The file pool will be automatically saved with your experiment if you save your experiment in the .opensesame.tar.gz format (which is the default format).

Your file pool now looks like Figure 7.

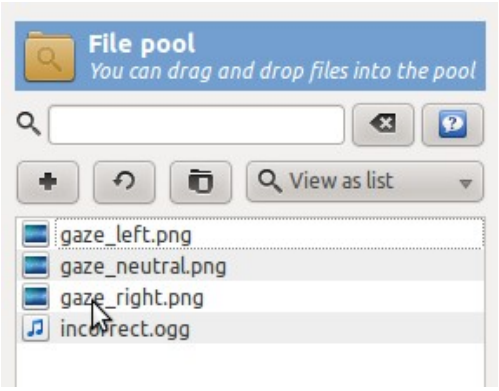


Figure 7. The file pool after step 4.

Step 5: Fill the trial sequence with items

A trial in our experiment is fairly straightforward.

1. **Fixation dot** — 750 ms, sketchpad item
2. **Neutral gaze** — 750ms, sketchpad item
3. **Gaze cue** — 500ms, sketchpad item
4. **Target** — 0ms, sketchpad item
5. **Response collection** — keyboard_response item
6. **Play a sound if response was incorrect** — sampler item
7. **Log response to file** — logger item

Click on 'trial_sequence' in the overview to open the trial_sequence tab. Select 'sketchpad' in the 'Append new item' menu and click four times on the 'plus' button. Similarly, select and append a keyboard_response item, a sampler item, and a logger item.

- **What is a keyboard_response item?** — As the name suggests, a keyboard_response item collects a participant's response from the keyboard.
- **What is a sampler item?** — A sampler item plays a sound from a sound file.
- **What is a logger item?** — A logger item writes data to the log file. This is very important, because if you forget to include a logger item, no data will be logged during the experiment!

Again, we will rename the new items, to make sure that the trial_sequence is easy to understand. Rename 'sketchpad' to 'fixation_dot', '_sketchpad' to 'neutral_gaze', '__sketchpad' to 'gaze_cue', '___sketchpad' to 'target', and 'sampler' to

'incorrect_sound'. (See step 1 if you don't remember how to do this.)

The incorrect_sound item should only be executed if an error was made. To do this, we need to change the conditional statement (in the 'Run if …' field) to 'correct = 0'. This works, because the keyboard_response item automatically creates a 'correct' variable, which is set to '1' (correct), '0' (incorrect) or 'undefined' (if no correct_response variable was defined, see step 3).

The trial_sequence now looks like Figure 8. You can download the experiment up to this point here: step5.opensesame.tar.gz

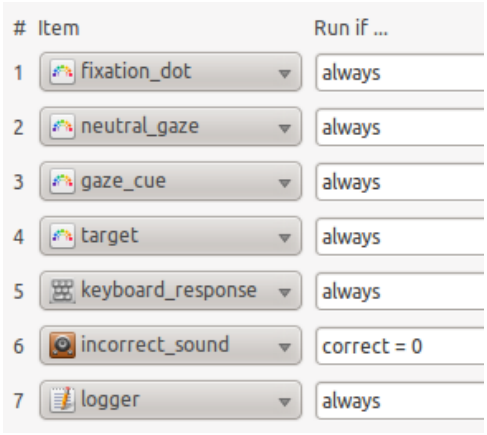


Figure 8. The trial sequence after step 5.

Step 6: Draw the sketchpad items

The sketchpad items that we have created in step 5 are still blank. It's time to do some drawing!

Set the background color to black — Click on 'fixation_dot' in the overview to open its tab. You can see that the sketchpad is black, while the images that we have downloaded have a white background. Oops, we forgot to set the background color of the experiment to white (it is black by default)! Click on 'Tutorial 2: Gaze cuing' in the overview. Change 'Foreground color' to 'black' and 'Background color' to 'white'.

Draw the fixation dot — Go back to the fixation_dot by clicking on 'fixation_dot' in the overview. You can draw on the sketchpad right away, but (unless your screen is really large) it looks a bit cramped. Therefore, click on 'Open editor in new window' to open the sketchpad editor in a separate window. Now select the fixation dot tool by clicking on the button with the small gray dot. If you move your cursor over the sketchpad, you can see the screen coordinates in the top-right. Click on the center of the screen (0, 0) to draw a central fixation dot. Click on 'Close' to close the editor. Finally, change the 'Duration' from 'keypress' to '750', since we want the fixation dot to be presented for 750 ms.

- **Tip:** Make sure that the (foreground) color is set to black. Otherwise you will draw white on white and won't see anything!

Draw the neutral gaze — Open the neutral_gaze sketchpad and, again, open the sketchpad editor in a new window. Now select the image tool by clicking on the button with the aquarium-like icon. Click on the center of the screen (0, 0). The 'Select file from pool' dialog will appear. Select 'gaze_neutral.png' and click on the 'Select' button. The neutral gaze image will now stare at you from the center of the screen! Click on 'Close' to close the editor. Finally, like before, change the 'Duration' from 'keypress' to '750'.

Draw the gaze cue. Open the gaze_cue sketchpad, open the sketchpad editor in a new window, and select the image tool. Click on the center of the screen (0, 0) and select 'gaze_left.png'. Click on 'Close' to close the editor.

Obviously, we are not done yet, because the gaze cue should not always be 'left', but should depend on the variable 'gaze_cue', which we have defined in step 3. However, by drawing the 'gaze_left.png' image to the sketchpad, we have generated a script which needs only a tiny modification to make sure that the proper image is shown. Click on the 'Edit script' button at the top-right of the tab. You will see the script that corresponds to the sketchpad that we have just created:

```
set duration "keypress"

set description "Displays stimuli"

set start_response_interval "no"

draw image 0.0 0.0 "gaze_left.png"
scale=1.0 center=1
```

The only thing that we have to do is replace 'gaze_left.png' with [gaze_cue]. This means that OpenSesame uses the variable 'gaze_cue' to determine which image should be shown. While we are at it, we might as well change the duration to '500'. Click on the 'Apply' button to apply the changes to the script, which now likes like this:

```
set duration 500

set description "Displays stimuli"

set start_response_interval "no"

draw image 0.0 0.0 [gaze_cue] scale=1.0
center=1
```

If you now go back to the gaze_cue tab by clicking on 'gaze_cue' in the overview, you will see a message saying that one object is not shown, because it is

defined using variables. Don't worry, it will be shown during the experiment!

Draw the target — We want three objects to be part of the target display: the target letter, the distractor letter, and the gaze cue. Like before, we will start by creating a static display using the sketchpad editor. After this, we will only need to make minor changes to the script so that the exact display depends on the variables.

Click on 'target' in the overview to open the target tab and open the editor in a new window. Like before, draw the 'gaze_left.png' image at the center of the screen. Now select the draw text tool by clicking on the button with the red 'Ab' icon. The default font size is 18pt, which is a bit small for our purpose, so change the font size to 32pt. Now click on (-320, 0) in the sketchpad (the X-coordinate does not need to be exactly 300, since we will change this to a variable anyway). Enter '[target_letter]' in the dialog that appears, to draw the target letter (when drawing text, you can use variables directly). Similarly, click on (320, 0) and draw an 'X' (the distractor is always an 'X'). Click on the 'Close' button.

- **Tip:** Make sure that the (foreground) color is set to black. Otherwise you will draw white on white and won't see anything!

Now open the script editor by clicking on the 'Edit script' button at the top-right of the tab. The script looks like this.

```
set duration "keypress"

set description "Displays stimuli"

set start_response_interval "no"

draw image 0.0 0.0 "gaze_left.png"
scale=1.0 center=1

draw textline -320.0 0.0 "[target_letter]"
center=1 color=black font_family=mono
font_size=32

draw textline 320.0 0.0 "X" center=1
color=black font_family=mono font_size=32
```

Like before, change 'gaze_left.png' to [gaze_cue]. We also need to make the position of the target and the distractor depend on 'target_pos' and 'dist_pos' respectively. To do this, simply change '-320' to '[target_pos]' and '320' to '[dist_pos]'. Click 'Apply' to apply the changes to the script, which now looks like this:

```
set duration "keypress"

set description "Displays stimuli"

set start_response_interval "no"

draw image 0.0 0.0 [gaze_cue] scale=1.0
center=1

draw textline [target_pos] 0.0
"[target_letter]" center=1 color=black
font_family=mono font_size=32

draw textline [dist_pos] 0.0 "X" center=1
color=black font_family=mono font_size=32
```

Go back to the target tab by clicking on 'target' in the overview. We still need to change two final things.

First, set 'Duration' to '0'. This does not mean that the target is presented for only 0ms, but that the experiment will advance to the next item (the keyboard_response) right away. Since the keyboard_response waits for a response, but doesn't change the screen, the target will remain visible until a response has been given.

Second, check the box marked 'start_response_interval'. This will make the keyboard_response use the onset of the target item as the start of the response interval when determining the response_time. If you don't explicitly set the start of the response interval, the onset of the response item will be used. In this case, this boils down to the same thing, but it is good practice to set the start of the response interval anyway.

You can download the experiment up to this point here: step6.opensesame.tar.gz

Step 7: Configure the keyboard response item

Click on 'keyboard_response' in the overview to open its tab. You will see three options: correct response, allowed responses, and timeout.

We have already set the 'correct_response' variable in step 3, so we don't need to set it here. If we do, we will simply override the previously set correct response, which is definitely not what we want.

We do need to set the allowed responses. Enter 'z;/' in the allowed responses field. The semicolon is used to separate responses. So the keyboard_response now only accepts 'z' and '/' keys. All other key presses are ignored, with the exception of 'escape' which aborts the experiment.

We also want to set a timeout, which is the maximum interval that the keyboard_response waits before deciding that the response was incorrect and setting the 'response' variable to 'timeout'. '2000' (ms) is a good value.

The keyboard_response now looks like Figure 9. You can download the experiment up to this point here: step8.opensesame.tar.gz

Figure 9. The keyboard response after step 8.

Step 8: Configure the incorrect (sampler) item

The incorrect item doesn't need that much work. Essentially we only need to select the sound that should be played. Click on 'incorrect' in the overview to open its tab. Click on the 'Browse' button and select 'incorrect.ogg' from the file pool.

The sampler now looks like Figure 10. You can download the experiment up to this point here: step8.opensesame.tar.gz

Figure 10: The incorrect sampler after step 8.

Step 9: Configure the variable logger

We need to determine which variables we want to log to the data file. Click on 'logger' in the overview to open its tab. You will see many variables, most of which are not very relevant. Luckily, OpenSesame can help us select the appropriate variables. Click on the 'Smart select' button. This will automatically select:

- all variables that are defined in the loops;
- all variables offered by the response items (the keyboard_response in this case);
- the count_[item name] variables of all sequence items, which keep track of how often a particular sequence has been executed (essentially these trial and block counters).

We don't need anything else, so we're done here.

The logger now looks like Figure 11. You can download the experiment up to this point here: step9.opensesame.tar.gz

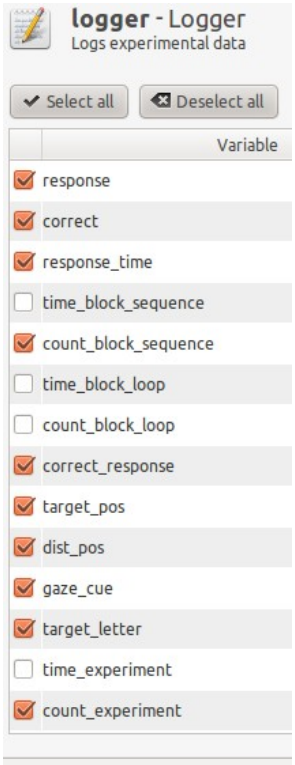


Figure 11. The logger after step 9.

Step 10: Draw the feedback item

After every block of trials, we want to present feedback to the participant to let him/ her know how well he/ she is doing. Click on 'feedback' in the overview to open its tab and open the editor in a new window. Select the draw text tool and click at (0, -128). Enter 'Your average response time was [avg_rt]ms'. Simarly, draw 'Your accuracy was [acc]%' at (0, -64) and 'Press any key to continue …' at (0, 64). Click on the 'Close' button to close the editor.

- **Feedback and variables** — Response items automatically keep track of the accuracy and average response time of the participant in the

variables 'acc' (synonym: 'accuracy') and 'avg_rt' (synonym: 'average_response_time') respectively. Feedback items automatically resets these variables.

- **Tip** — Make sure that the (foreground) color is set to black. Otherwise you will draw white on white and won't see anything!

The feedback item now looks like Figure 12 You can download the experiment up to this point here: step10.opensesame.tar.gz

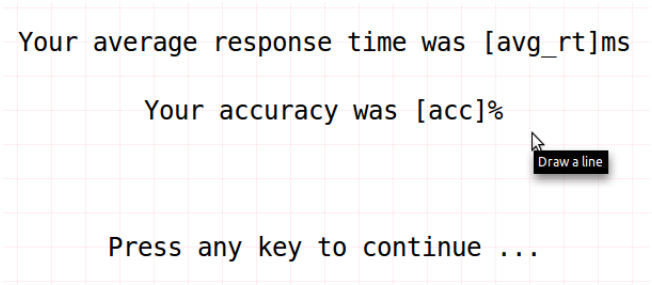


Figure 12. The feedback item after step 10.

Step 11: Set the length of the practice phase and experimental phase

We have previously created the practice_loop and the experiment_loop items, which both call the block_sequence (i.e., a block of trials). However, right now they call the block_sequence only once, which means that both the practice and the experimental phase consist of only a single block of trials.

Click on 'practice_loop' to open its tab and set 'Repeat' to '2'. This means that the practice phase consists of two blocks.

Click on 'experimental_loop' to open its tab and set 'Repeat' to '8'. This means that the experimental phase consists of eight blocks.

Step 12: Draw the instructions, end_of_practice and end_of_experiment sketchpads

I think you can handle this step your own! Simply open the appropriate tabs and draw some text on the sketchpads to present instructions, an end-of-practice message and an end-of-experiment message.

- **Tip:** Make sure that the (foreground) color is set to black. Otherwise you will draw white on white and won't see anything!

Step 13: Run the experiment!

You're done! Click on the 'Run in window' or 'Run fullscreen' buttons in the toolbar to run your experiment.

- **Tip** — If you want to give your experiment a test run without having to press any keys, you can activate the auto response option (Menu → Run → Enable auto response). In this mode, OpenSesame will simulate the participants responses.

You can download the finished experiment here: tutorial2.opensesame.tar.gz

References

Friesen, C. K., & Kingstone, A. (1998). The eyes have it! Reflexive orienting is triggered by nonpredictive gaze. *Psychonomic Bulletin & Review, 5,* 490–495.

Mathôt, S., & Theeuwes, J. (2010). OpenSesame (Version 0.17) [Computer software and manual]. Retrieved December 18, 2010, from http://www.cogsci.nl/opensesame