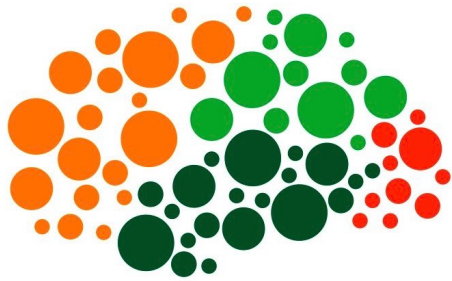
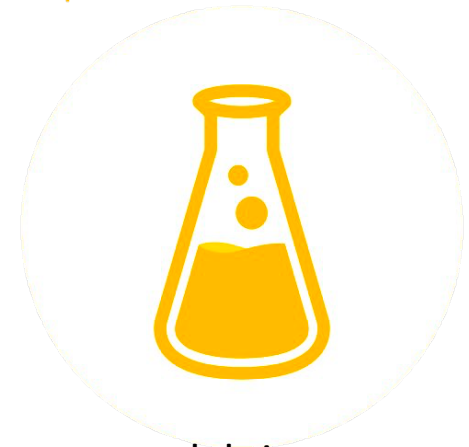


Comparaison outils d'expériences en ligne



jsPsych



lab.js

Résumé de la réunion précédente

- Choix du langage javascript
- Réduction de fait à deux outils : jsPsych / lab.js
- Comparer ces deux outils sur différents critères :
 - Facilité de prise en main
 - Modularité
 - Capacité à produire les principaux paradigmes expérimentaux qui intéressent l'équipe

Référence de paradigmes expérimentaux

- *Sonic interaction design (Franinovic, Serafin; 2012)*

<https://nubo.ircam.fr/index.php/s/aPorzEz9g546PyS>

- *Measurement with persons : Theory, Methods and Implementation Areas (Berglund, B. Rossi, T. Townsend, R. Pendrill; 2011)*

<https://nubo.ircam.fr/index.php/s/9eK2y3pFGYCJiXR>

Table 5.1 Examples of questions answered with the methods described in this chapter.

Section: Method	Examples of answered questions
5.3: Psychophysical methods	<ul style="list-style-type: none">– Can the user perceive each of the configurations of a sonic interaction?– Can the user differentiate between configurations?
5.4: Identification and categorization	<ul style="list-style-type: none">– What naturalistic object is recognized in each of the configurations?– What emotional category is recognized in a sonic artefact?
5.5: Scaling and rating	<ul style="list-style-type: none">– How does perceived effort vary between sonic feedbacks for robotic surgery applications?– How should the user-controlled gain for sound level vary so as to produce a linear increase in perceived loudness?
5.6: Dissimilarity estimation	<ul style="list-style-type: none">– Which properties of a complex sonic interaction are most relevant to the user?– Do different individuals focus on different attributes of the sensory events?
5.7: Sorting	<ul style="list-style-type: none">– How many categories of perceived materials can a sound synthesis algorithm reproduce?– What is the most typical configuration for each of the material categories?
5.8: Verbalization	<ul style="list-style-type: none">– What are the relevant semantic descriptors to describe the sound properties of a sonic interaction?– What are the individual interactive strategies? Are there problems in the prototype design?
5.9: Semantic differential	<ul style="list-style-type: none">– Which configuration has the highest aesthetic and functional value?– How do preference, perceived sound brightness and perceived efficiency covary for these particular sonic interactions?
5.10: Preference estimation	<ul style="list-style-type: none">– Which configuration of a sonic feedback system do users prefer the most?– Which configuration is the least annoying?
5.11: Continuous evaluation	<ul style="list-style-type: none">– Do users gesture map onto changes in the perceptual attributes of the sonic events?– How does the emotional response to a complex sound vary in time?
5.12: Multisensory contexts	<ul style="list-style-type: none">– What influences most strongly preference for cars? The sound of its doors closing or their felt weight?– Do sonic feedbacks significantly shorten the time required for parking a car?
5.13: Measurement of acoustical information	<ul style="list-style-type: none">– What sound properties should be manipulated to induce a target perceptual result (e.g., maximize preference)?
5.14: Motion capture	<ul style="list-style-type: none">– How do we use our body in interaction with a sonic artefact?– How do gestures and artefacts mutually influence a sonic interaction?

Il a été choisi lors de la précédente réunion d’ajouter également la BWS, en soulevant qu’elle metait en jeu beaucoup d’éléments permettant de tester les capacités des deux outils évalués.

Facilité de prise en main

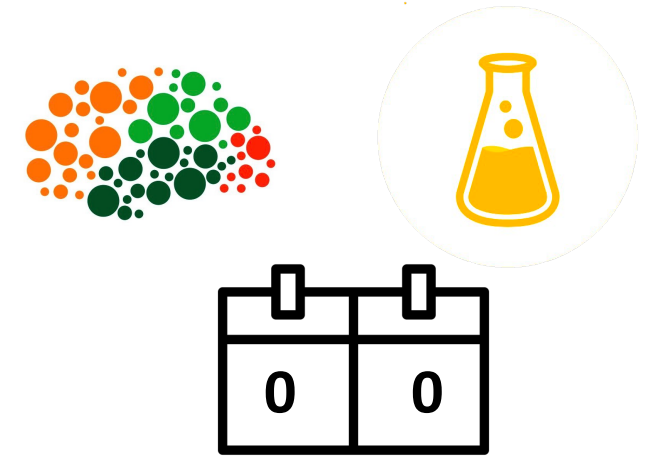
- Deux approches différentes

lab.js :

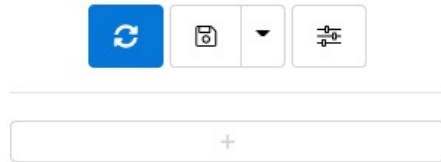
Builder

jsPsych :

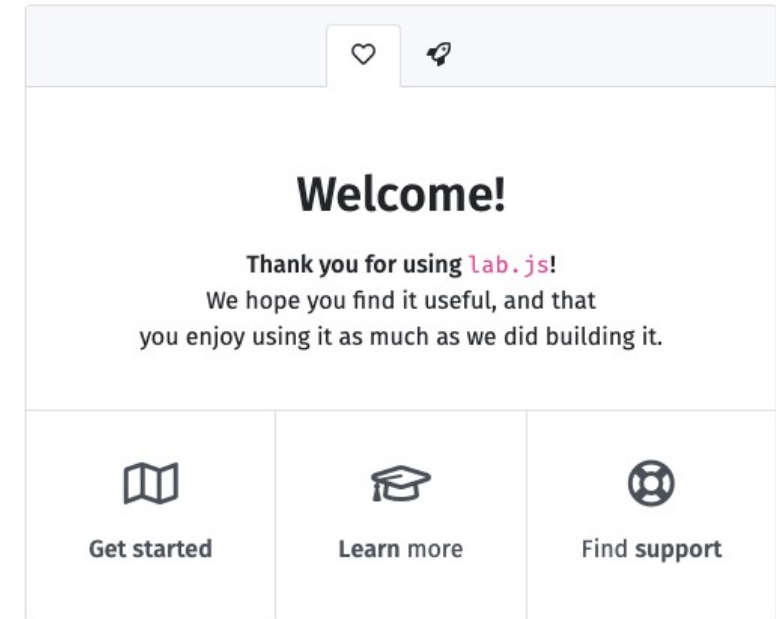
Librairie seule



lab.js







Interface graphique *user-friendly*




lab.js






Welcome

Instructions 

Loop


Iteration





Fixation cross

Task2 

Gap

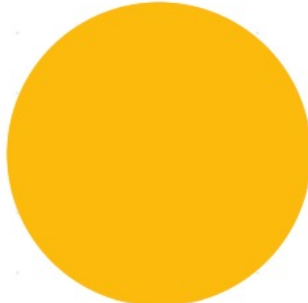
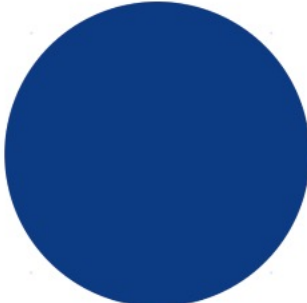
Bilan

Instructions 

Content Behavior Scripts 

Content



















Dans cette expérience, un cercle va apparaître au centre de l'écran.
Si le cercle est Bleu appuyez sur la lettre F sur votre clavier aussi vite que vous le pouvez.
Si le cercle est Orange appuyez sur la lettre J aussi vite que vous le pouvez.



Appuyez la lettre F

Appuyez la lettre J

Appuyez sur une touche pour commencer.



Possibilité d'éditer du javascript sans utiliser une ligne de code

lab.js



Loop

Content

Behavior

Scripts

Loop

	position	A ▾	color	A ▾	fixation	⌚ ▾	touche	A ▾	+
≡	225		blue		250		f		🗑
≡	-225		orange		500		j		🗑
≡	225		blue		750				🗑
≡	-225		orange		1000				🗑
≡	225		blue		1250				🗑
≡	-225		orange		1500				🗑
≡	225		blue		1750				🗑
≡	-225		orange		2000				🗑
+									

Sample ⓘ

20

Sampled with replacement ▾

Further options

lab.js



Bilan

Content

Behavior

Scripts

Timeline Beta

0ms

100ms

200ms

300ms

400ms

500ms

600ms

700ms

800ms

900ms

1000ms

1100ms

Please add or select a timeline item

+

Timeout

Never

ms

Responses

label

action · event

target

filter · key/button

none (inactive)

window

any

+

Correct response

Undefined

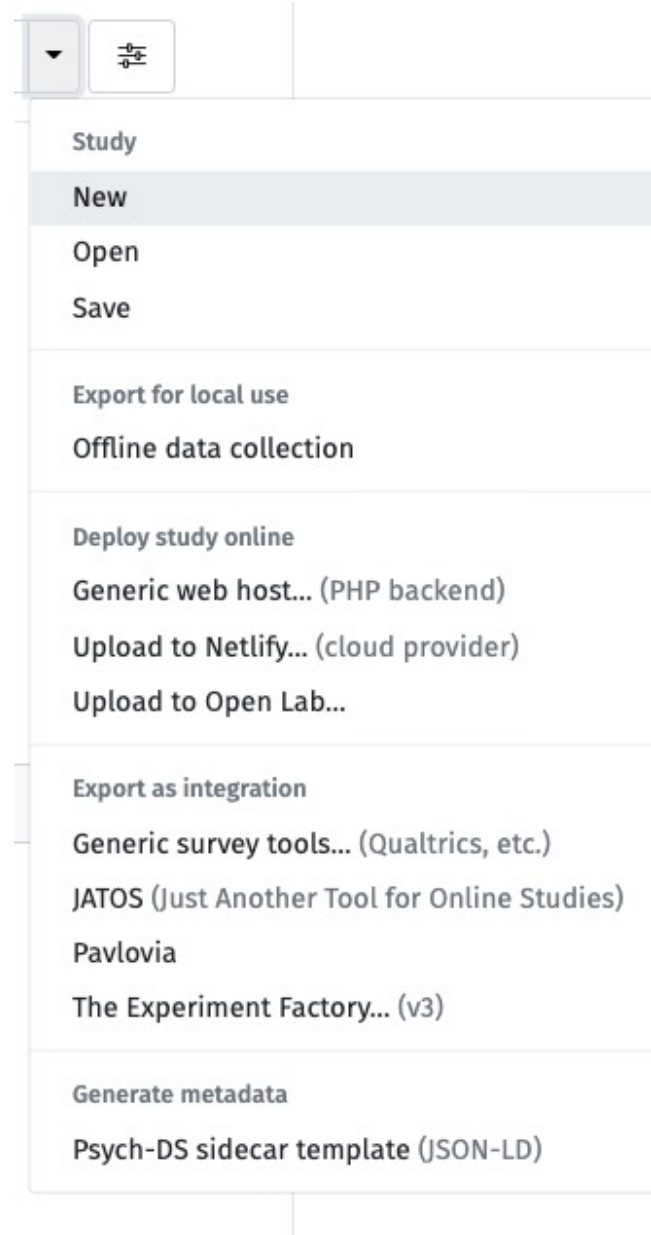
Meta

Skip

`${ optional condition }`

Tardy

lab.js



Mise en ligne simplifiée



jsPsych

- Absence d'interface utilisateur

```
experience3.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>My experiment</title>
5 <script src="https://unpkg.com/jspsych@7.0.0"></script>
6 <script src="https://unpkg.com/@jspsych/plugin-html-keyboard-response@1.0.0"></script>
7 <script src="https://unpkg.com/@jspsych/plugin-image-keyboard-response@1.0.0"></script>
8 <script src="https://unpkg.com/@jspsych/plugin-preload@1.0.0"></script>
9 <link href="https://unpkg.com/jspsych@7.0.0/css/jspsych.css" rel="stylesheet" type="text/css" />
10 </head>
11 <body></body>
12 <script>
13
14 /* Je lance jsPsych et je lui dit de me montrer les data à la fin */
15 var jsPsych = initJsPsych({
16   on_finish: function() {
17     jsPsych.data.displayData();
18   }
19 });
20
21 /* Je crée la timeline de l'expérience */
22 var timeline = [];
23
24 /* On va précharger les média pour éviter la latence dans le navigateur */
25 var preload = {
26   type: jsPsychPreload,
27   images: ['img/blue.png', 'img/orange.png']
28 };
29 timeline.push(preload);
30
31 /* Je définie le message d'accueil comme un trial */
32 var welcome = {
33   type: jsPsychHtmlKeyboardResponse,
34   stimulus: "Bienvenue dans l'expérience test. Appuyez sur une touche pour commencer."
35 };
36 timeline.push(welcome);
37
38 /* On montre les instructions */
39 var instructions = {
40   type: jsPsychHtmlKeyboardResponse,
41   stimulus:
42     <p>Dans cette expérience, un cercle va apparaître au centre de l'écran.</p><p>Si le cercle est <strong>Bleu</strong>,<br>
43     <p>Si le cercle est <strong>Orange</strong>, appuyez sur la lettre J aussi vite que vous le pouvez.</p>
44     <div style='width: 700px;'>
45       <div style='float: left;'><img src='img/blue.png'></div>
46       <p class='small'><strong>Appuyez la lettre F</strong></p></div>
47       <div style='float: right;'><img src='img/orange.png'></div>
48       <p class='small'><strong>Appuyez la lettre J</strong></p></div>
49     </div>
50     <p>Appuyez sur une touche pour commencer.</p>
51   ,
52   post_trial_gap: 2000
53 };
54 timeline.push(instructions)
55
56 /* On défini une variable de timeline les stimulus */
57
58 var test_stimuli = [
59   { stimulus: "<div style='float: right;'><img src='img/blue.png'></div>", correct_response: 'f' },
60   { stimulus: "<div style='float: right;'><img src='img/orange.png'></div>", correct_response: 'j' }
61 ];
62
63 /* Maintenant on ajoute une croix de fixation entre les présentations */
64
65 var fixation = {
66   type: jsPsychHtmlKeyboardResponse,
67   stimulus: '<div style="font-size:60px;">+</div>',
68   choices: "NO_KEYS",
69   trial_duration: function(){
70     return jsPsych.randomization.sampleWithoutReplacement([250, 500, 750, 1000, 1250, 1500, 1750, 2000], 1)[0];
71   },
72   data: {
73     fixation_duration: function(){
74       return jsPsych.randomization.sampleWithoutReplacement([250, 500, 750, 1000, 1250, 1500, 1750, 2000], 1)[0];
75     }
76   }
77 };
78 timeline.push(fixation);
79
80 /* On ajoute les stimuli à la timeline */
81 timeline.push(test_stimuli);
82
83 /* On lance l'expérience */
84 jsPsych.run(timeline);
85 </script>
86 </html>
```



jsPsych

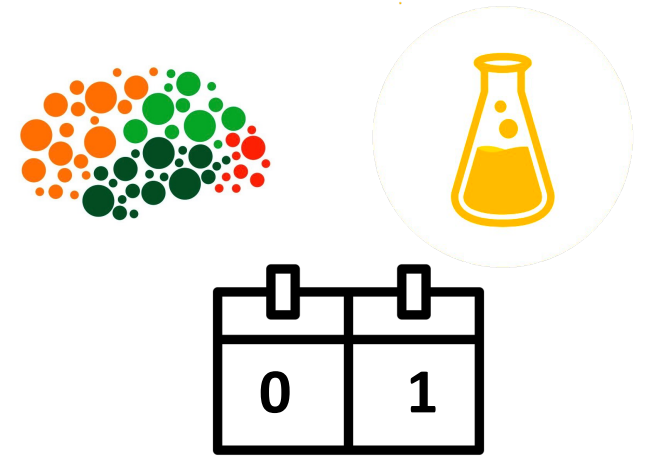
(Sauf via node.js qui semble permettre de s'en approcher)



Exemple de réalisation de la même expérience sur les deux outils

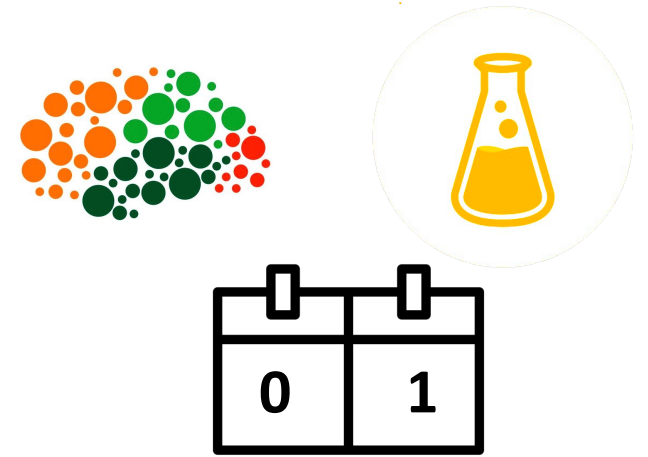
Facilité de prise en main

- lab.js est plus simple à prendre en main lorsque l'on ne maîtrise pas le javascript.
- Néanmoins si le projet de GUI utilisant jsPsych s'avère fonctionnel, le problème semble contournable au moins en partie.



Modularité

- Quel outil répond de la façon la plus adéquate au besoin de modularité, afin de permettre le partage au sein de l'équipe et entre chercheurs, des éléments communs aux différentes expériences ?





lab.js

- Facilité pour l'importation / l'exportation des expériences réalisées sur le builder.
 - Téléchargeables en *.json*
 - Expériences de références disponibles sur le site de l'outil.
- Négatif : peu de plugins sont disponibles de façon native, aucun d'eux n'est orienté audio.

jsPsych

- Librairie basée entièrement sur les plugins.
- Beaucoup de plugins disponibles, donc plusieurs dédiés à l'audio.
- Possibilité d'utiliser des extensions qui s'interfacent avec un plugin existant, pour élargir les possibilités.

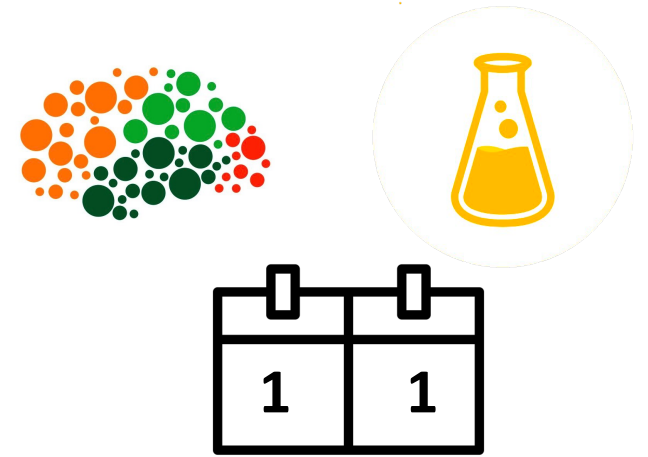
*Des templates sont disponibles pour coder des plugins et extensions.
Reste à évaluer la difficulté que cela représente.*

List of Plugins	
These are the plugins that are included in the jsPsych release. Additional plugins may be available in the community repository .	
For an overview of what plugins are and how they work, see our plugins overview .	
Plugin	Description
animate	Shows a sequence of images at a specified frame rate. Displays key presses (including timing information) made by the subject while they are viewing the animation.
audio-button-response	Plays an audio file and allows the subject to respond by clicking a button to click. The button can be customized extensively (e.g., using images in place of standard buttons).
audio-keyboard-response	Plays an audio file and allows the subject to respond by pressing a key.
audio-slider-response	Plays an audio file and allows the subject to respond by moving a slider to indicate a value.
call-function	Executes an arbitrary function call. Eventual display anything to the subject, and the subject is usually unaware that this plugin has even executed. It's useful for performing tasks as specified for each experiment, such as setting data.
dynamic-button-response	Shows a stimulus on a HTML canvas element, and records a button click response. Useful for displaying dynamic, parametrically defined graphics, and for controlling the positioning of multiple graphical elements (images, text, images).
dynamic-keyboard-response	Shows a stimulus on a HTML canvas element, and records a key press response. Useful for displaying dynamic, parametrically defined graphics, and for controlling the positioning of multiple graphical elements (images, text, images).
dynamic-slider-response	Shows a stimulus on a HTML canvas element, and asks the subject to respond by moving a slider to indicate a value. Useful for displaying dynamic, parametrically defined graphics, and for controlling the positioning of multiple graphical elements (images, text, images).
integration-animation	The subject responds to an animation and can be given feedback about their response.
integration-text	The subject responds to an HTML-formatted stimulus using the keyboard and can be given feedback about the contents of their response.
integration-image	The subject responds to an image using the keyboard and can be given feedback about the contents of their response.
choice	Plays for displaying a choice task and recording participants' answers against a correct solution.
external-html	Displays an external HTML page (such as a consent form) and lets the subject respond by clicking a button or pressing a key. Plugins can capture their response, which is useful for making sure that a subject has given consent before starting the experiment.
eye-tracking	Displays a set of images on the screen in random locations. Subjects can click and drag the images to move them around the screen. Records all the moves made by the subject, so the sequence of moves can be recovered from the data.
full-screen	Toggles the experiment to and out of full-screen mode.
forced-button-response	Display an HTML-formatted stimulus and allows the subject to respond by clicking a button to click. The button can be customized extensively (e.g., using images in place of standard buttons).
forced-keyboard-response	Display an HTML-formatted stimulus and allows the subject to respond by pressing a key.
forced-slider-response	Display an HTML-formatted stimulus and allows the subject to respond by moving a slider to indicate a value.
hit-test	The implicit association task, using HTML-formatted stimuli.
hit-image	The implicit association task, using images as stimuli.
image-button-response	Displays an image and allows the subject to respond by clicking a button to click. The button can be customized extensively (e.g., using images in place of standard buttons).
image-keyboard-response	Displays an image and allows the subject to respond by pressing a key.
image-slider-response	Displays an image and allows the subject to respond by moving a slider to indicate a value.
link-buttons	For displaying link buttons to the subject. Allows the subject to navigate between pages of individual subpages on buttons.
mouse02	Displays rows of alternatives (such as stimuli) for free mutually exclusive categorization, typically as 'true' or 'false' on a particular column (e.g., importance, preference, similarity). The participant responds by entering one radio button corresponding to an alternative in both the left and right response columns.
pseudostim	This plugin loads images, audio, and video files into the browser's memory before they are needed in the experiment in order to improve stimulus and response timing, and to avoid disrupting the flow of the experiment.
randomize-stimulus	The subject interacts with a stimulus by modifying a parameter of the stimulus and observing if a change to the stimulus is visible.
reaction	Calibrates the display so that stimuli displayed with a known physical size.
same-different-text	A same-different judgment task. An HTML-formatted stimulus is shown, followed by a text gap, and then another stimulus is shown. The subject indicates whether the stimuli are the same or different.
same-different-image	A same-different judgment task. An image is shown, followed by a text gap, and then another stimulus is shown. The subject indicates whether the stimuli are the same or different.
serial-reaction-time	A set of tones are displayed on the screen and one of them changes color. The subject presses a key that corresponds to the different color tone as fast as possible.
serial-reaction-time-visual	A set of tones are displayed on the screen and one of them changes color. The subject clicks the box it is changed color as fast as possible.
several-choice-items	Presents a screen of HTML items, allows for mixing multiple kinds of feedback.
several-choice	Displays several-choice questions.
several-multi-choice	Displays multiple-choice questions with one answer allowed per question.
several-multi-select	Displays multiple-choice questions with multiple answers allowed per question.
several-text	Shows a prompt with a text box. The subject provides a response and then submits by clicking a button.
video-button-response	Displays a video file with many options for customizing playback. Subject responds to the video by one or more actions.



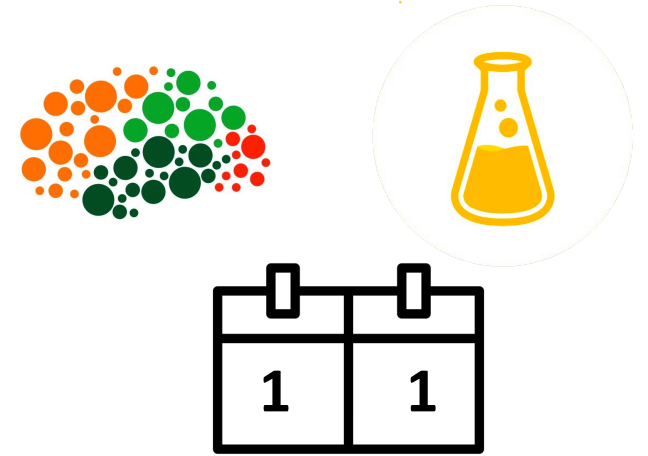
Modularité

- lab.js est modulaire, mais préférentiellement dans les limites de son builder.
- jsPsych est d'approche moins évidente, mais sa conception orientée plugins rend l'outil beaucoup plus modulaire.



Simplicité à produire les paradigmes classiques

- Les outils sont ils à même de fournir un cadre permettant de produire les paradigmes expérimentaux classiques nous intéressant ?





lab.js

La principale limitation est la contrepartie à la grande simplicité de l'outil, qui ne permet pas de façon directe de réaliser certaines expériences.

Cela reste en partie faisable, en rusant un peu sur l'outil.

✗
✓
≈
≈
✗
✓
≈
✓
✗
?

Table 5.1 Examples of questions answered with the methods described in this chapter.

Section: Method	Examples of answered questions
5.3: Psychophysical methods	<ul style="list-style-type: none">– Can the user perceive each of the configurations of a sonic interaction?– Can the user differentiate between configurations?
5.4: Identification and categorization	<ul style="list-style-type: none">– What naturalistic object is recognized in each of the configurations?– What emotional category is recognized in a sonic artefact?
5.5: Scaling and rating	<ul style="list-style-type: none">– How does perceived effort vary between sonic feedbacks for robotic surgery applications?– How should the user-controlled gain for sound level vary so as to produce a linear increase in perceived loudness?
5.6: Dissimilarity estimation	<ul style="list-style-type: none">– Which properties of a complex sonic interaction are most relevant to the user?– Do different individuals focus on different attributes of the sensory events?
5.7: Sorting	<ul style="list-style-type: none">– How many categories of perceived materials can a sound synthesis algorithm reproduce?– What is the most typical configuration for each of the material categories?
5.8: Verbalization	<ul style="list-style-type: none">– What are the relevant semantic descriptors to describe the sound properties of a sonic interaction?– What are the individual interactive strategies? Are there problems in the prototype design?
5.9: Semantic differential	<ul style="list-style-type: none">– Which configuration has the highest aesthetic and functional value?– How do preference, perceived sound brightness and perceived efficiency covary for these particular sonic interactions?
5.10: Preference estimation	<ul style="list-style-type: none">– Which configuration of a sonic feedback system do users prefer the most?– Which configuration is the least annoying?
5.11: Continuous evaluation	<ul style="list-style-type: none">– Do users gesture map onto changes in the perceptual attributes of the sonic events?– How does the emotional response to a complex sound vary in time?
5.12: Multisensory contexts	<ul style="list-style-type: none">– What influences most strongly preference for cars? The sound of its doors closing or their felt weight?– Do sonic feedbacks significantly shorten the time required for parking a car?



jsPsych

Plus de possibilités au prix d'une plus grande difficulté de prise en main.

Certains plugins déjà existants permettent de faciliter la création de certains paradigmes.

Table 5.1 Examples of questions answered with the methods described in this chapter.

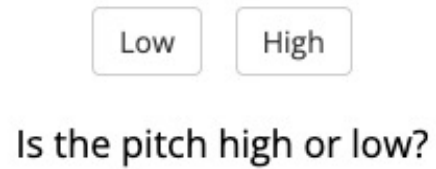


Section: Method	Examples of answered questions
5.3: Psychophysical methods	<ul style="list-style-type: none">– Can the user perceive each of the configurations of a sonic interaction?– Can the user differentiate between configurations?
5.4: Identification and categorization	<ul style="list-style-type: none">– What naturalistic object is recognized in each of the configurations?– What emotional category is recognized in a sonic artefact?
5.5: Scaling and rating	<ul style="list-style-type: none">– How does perceived effort vary between sonic feedbacks for robotic surgery applications?– How should the user-controlled gain for sound level vary so as to produce a linear increase in perceived loudness?
5.6: Dissimilarity estimation	<ul style="list-style-type: none">– Which properties of a complex sonic interaction are most relevant to the user?– Do different individuals focus on different attributes of the sensory events?
5.7: Sorting	<ul style="list-style-type: none">– How many categories of perceived materials can a sound synthesis algorithm reproduce?– What is the most typical configuration for each of the material categories?
5.8: Verbalization	<ul style="list-style-type: none">– What are the relevant semantic descriptors to describe the sound properties of a sonic interaction?– What are the individual interactive strategies? Are there problems in the prototype design?
5.9: Semantic differential	<ul style="list-style-type: none">– Which configuration has the highest aesthetic and functional value?– How do preference, perceived sound brightness and perceived efficiency covary for these particular sonic interactions?
5.10: Preference estimation	<ul style="list-style-type: none">– Which configuration of a sonic feedback system do users prefer the most?– Which configuration is the least annoying?
5.11: Continuous evaluation	<ul style="list-style-type: none">– Do users gesture map onto changes in the perceptual attributes of the sonic events?– How does the emotional response to a complex sound vary in time?
5.12: Multisensory contexts	<ul style="list-style-type: none">– What influences most strongly preference for cars? The sound of its doors closing or their felt weight?– Do sonic feedbacks significantly shorten the time required for parking a car?



jsPsych

- Quelques exemples visuels :



```
var trial = {  
  type: jsPsychAudioButtonResponse,  
  stimulus: 'sound/tone.mp3',  
  choices: ['Low', 'High'],  
  prompt: "<p>Is the pitch high or low?</p>"  
};
```



jsPsych

- Quelques exemples visuels :

Is the pitch high or low? Press 'e' for low and 'i' for high.

```
var trial = {  
  type: jsPsychAudioKeyboardResponse,  
  stimulus: 'sound/tone.mp3',  
  choices: ['e', 'i'],  
  prompt: "<p>Is the pitch high or low? Press 'e' for low and 'i' for high.</p>",  
  response_ends_trial: true  
};
```



jsPsych

- Quelques exemples visuels :



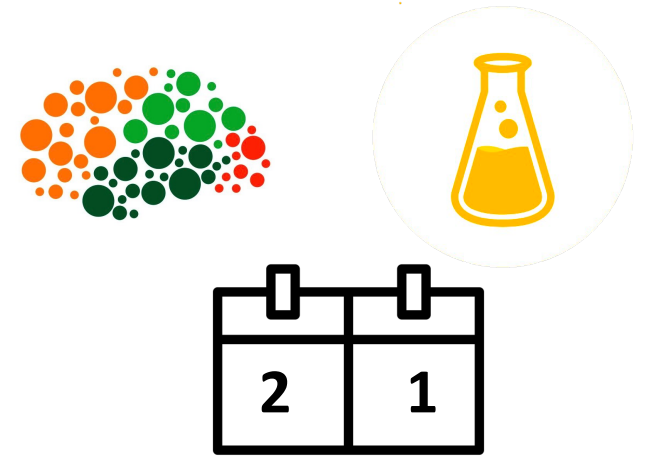
How funny is the joke?

Continue

```
var trial = {  
  type: jsPsychAudioSliderResponse,  
  stimulus: 'sound/speech_joke.mp3',  
  labels: ['Not Funny', 'Funny'],  
  prompt: '<p>How funny is the joke?</p>',  
}
```


Simplicité à produire les paradigmes classiques

- jsPsych est plus complet pour mettre en place la plupart des paradigmes en première approximation.
- lab.js pourrait permettre de le faire, au prix de quelques détournements pour conserver la facilité de son interface.

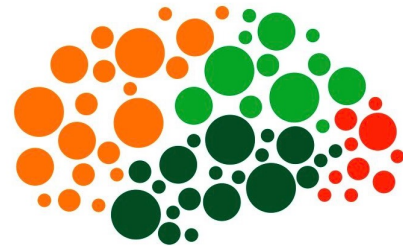


Premières conclusions

- lab.js est un outil simple de prise en main, qui permet de réaliser très rapidement des expériences basiques.
- jsPsych est plus complexe de prime abord, mais permet des manipulations plus précises lorsque l'on souhaite.

Proposition de sélection d'un seul outil

En vue d'un approfondissement et de la poursuite de la mission, proposition de ne conserver qu'un seul des deux outils.



Appronfondissement – Suite

Différentes questions doivent encore être étudiées et approfondies :

- jsPsych peut fonctionner avec HTML5 Audio et WebAudio API
- Quelles sont les limitations de formats audio ?
- Quel poids peut prendre l'ensemble des fichiers audio ?
- Question sur la génération d'audio ?
- Question de l'adaptabilité pour certains paradigmes ?