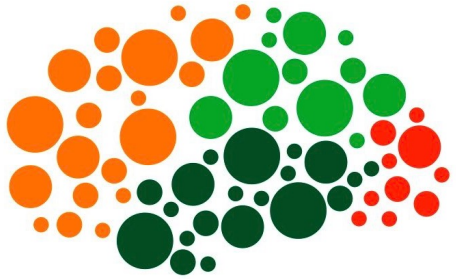
Comparaison outils d’expériences en ligne



jsPsych

lab.js

Résumé de la réunion précédente

- Choix du language 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

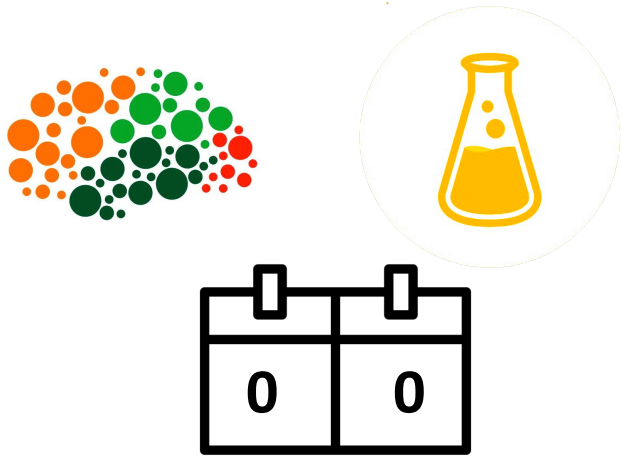
Table 5.1 Examples of questions answered with the methods described in this chapter.

Section: Method	Examples of answered questions
5.3: Psychophysical methods	– Can the user perceive each of the configurations of a sonic interaction? – Can the user differentiate between configurations?
5.4: Identification and categorization	– What naturalistic object is recognized in each of the configurations? – What emotional category is recognized in a sonic artefact?
5.5: Scaling and rating	– How does perceived effort vary between sonic feedbacks for robotic surgery applications? – How should the user-controlled gain for sound level vary so at to produce a linear increase in perceived loudness?
5.6: Dissimilarity estimation	– 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	– 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	– 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	– 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	– Which configuration of a sonic feedback system do users prefer the most? – Which configuration is the least annoying?
5.11: Continuous evaluation	– 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	– 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	– What sound properties should be manipulated to induce a target perceptual result (e.g., maximize preference)?
5.14: Motion capture	– 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.

p.176 *Sonic interaction design*

Facilité de prise en main



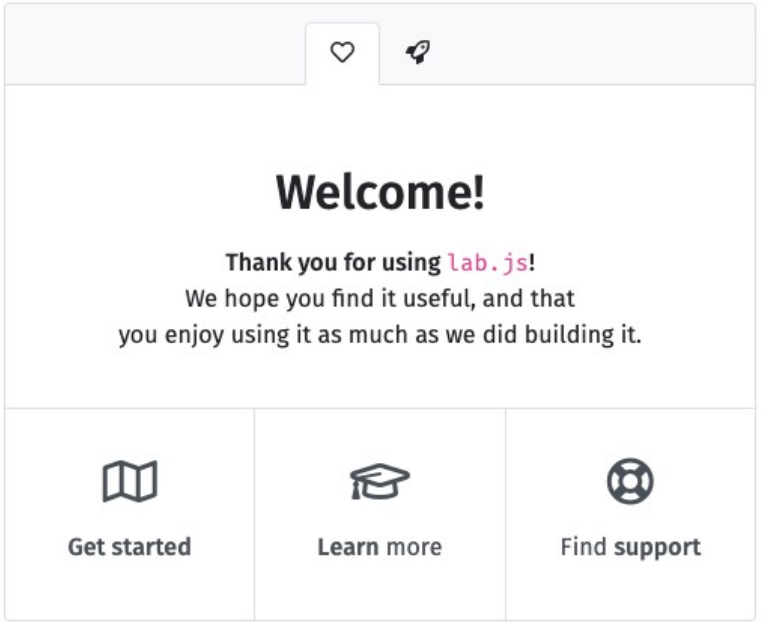
- Deux approches différentes

lab.js : **Builder**

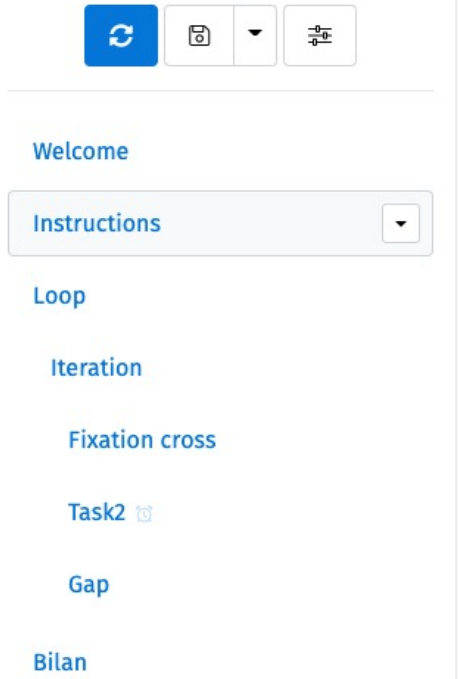
jsPsych : **Librairie seule**

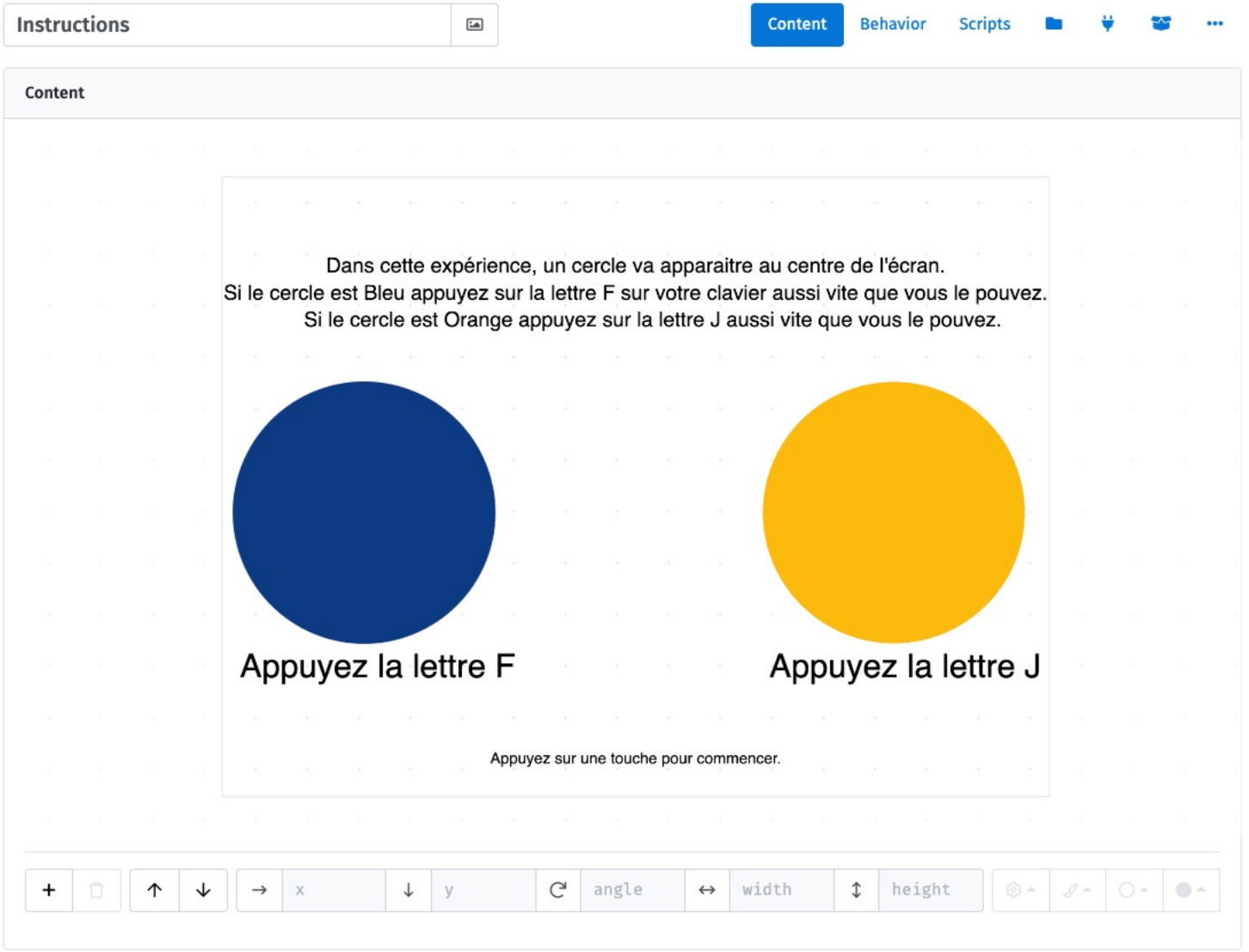
lab.js

Interface graphique *user-friendly*



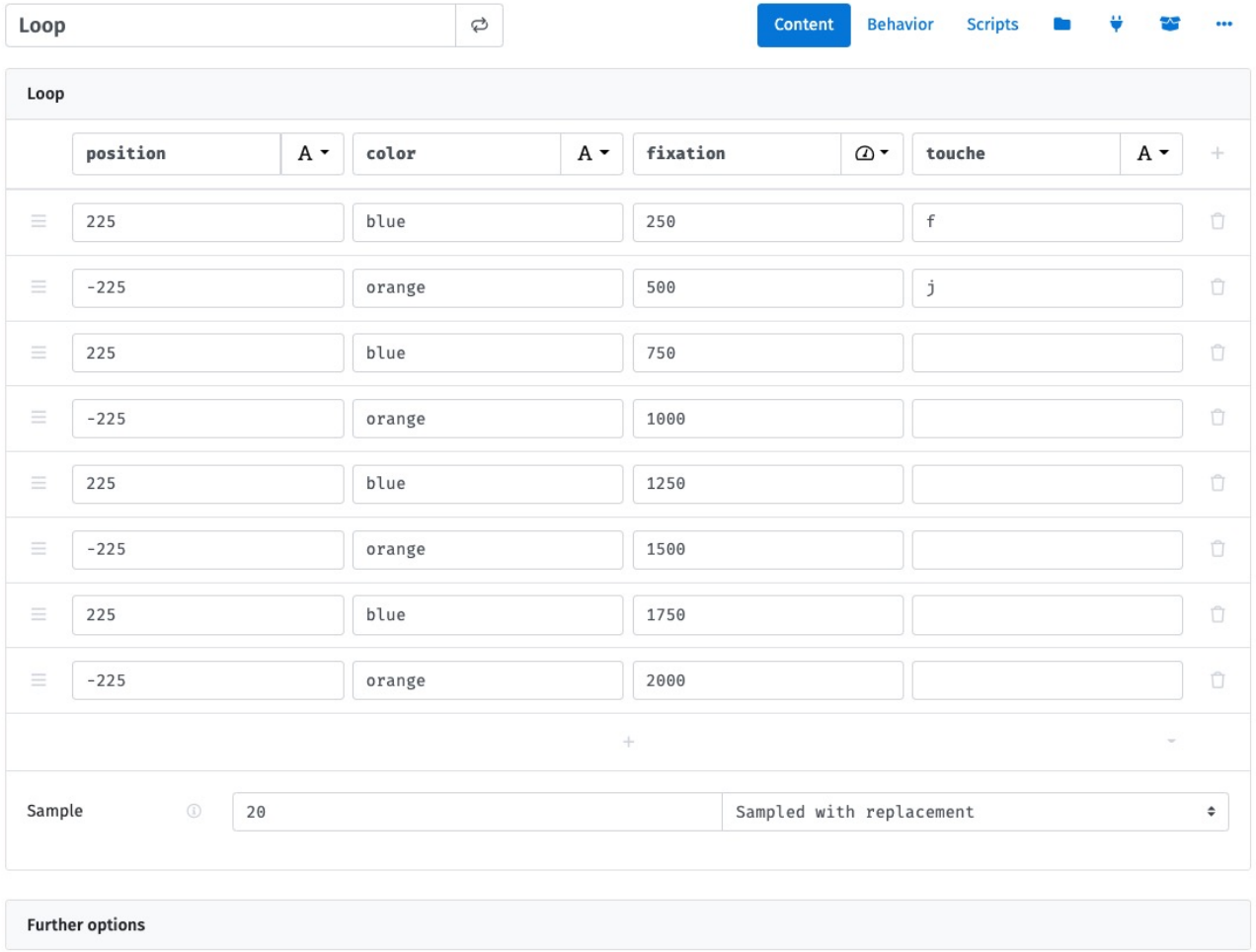
lab.js



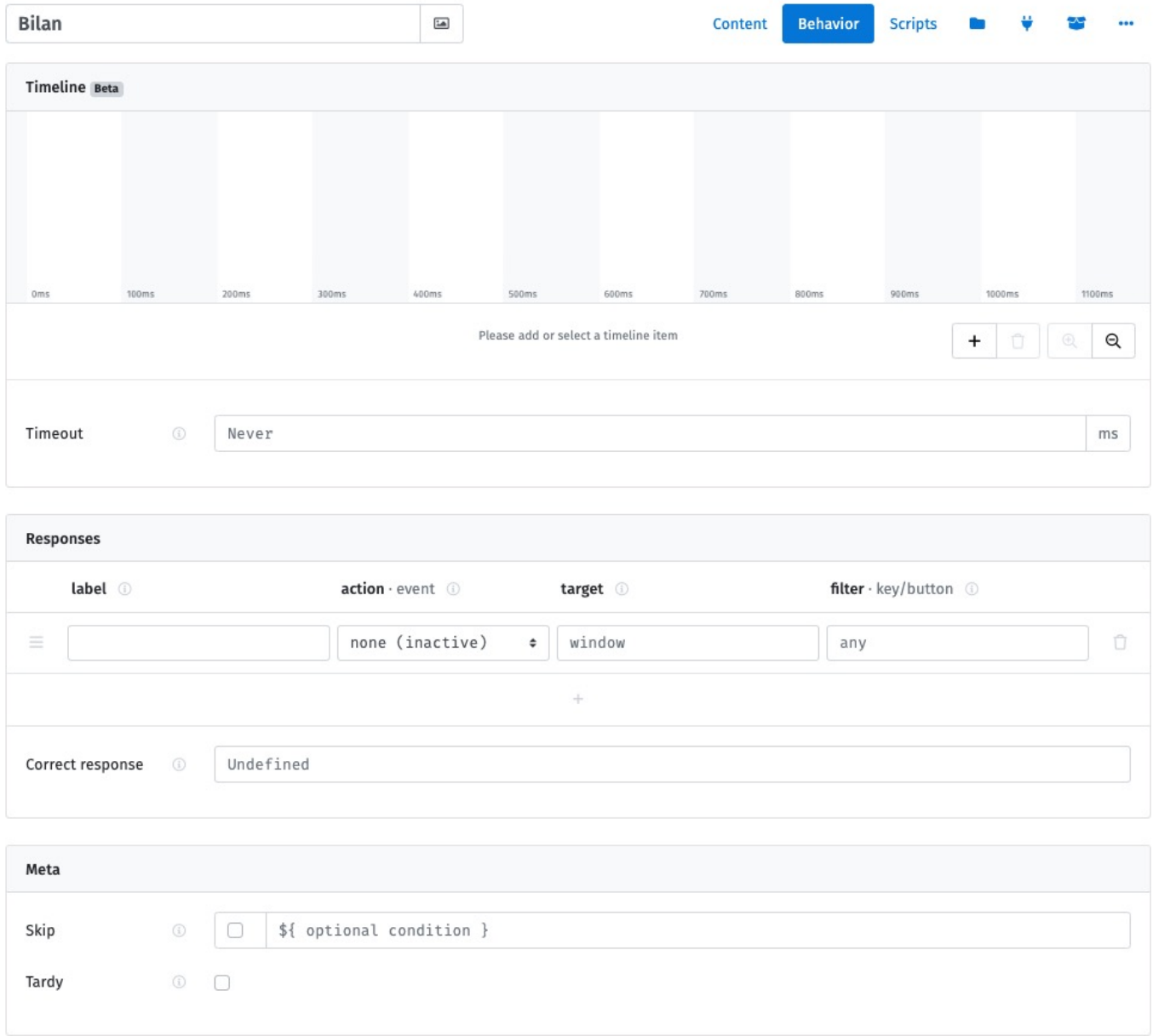


Possibilité d'éditer du javascript sans utiliser une ligne de code

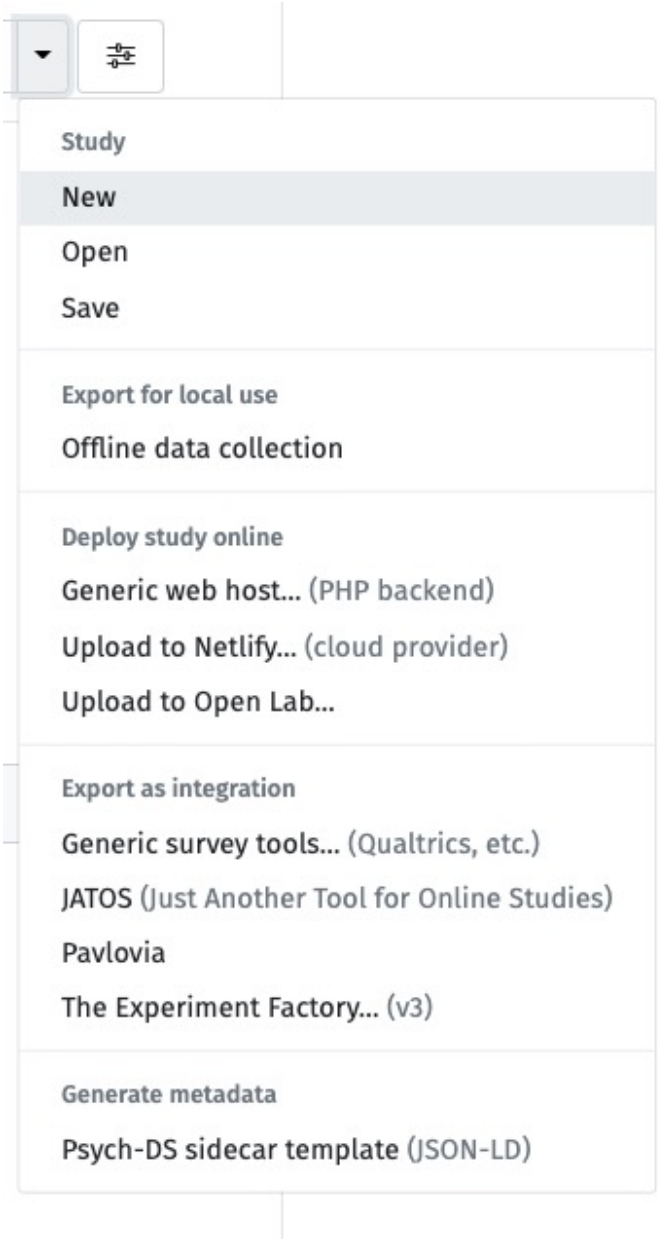
lab.js


Loop
Content
Behavior
Scripts
Loop
position
color
fixation
touche
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

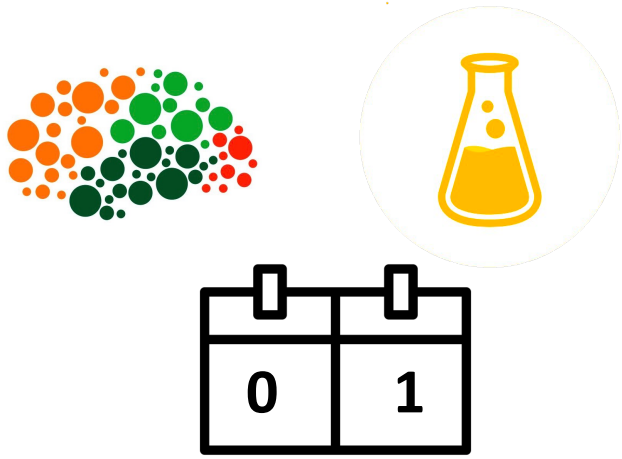
```
<!DOCTYPE html>
<html>
  <head>
    <title>My experiment</title>
    <script src="https://unpkg.com/jspsych@7.0.0"></script>
    <script src="https://unpkg.com/@jspsych/plugin-html-keyboard-response@1.0.0"></script>
    <script src="https://unpkg.com/@jspsych/plugin-image-keyboard-response@1.0.0"></script>
    <script src="https://unpkg.com/@jspsych/plugin-preload@1.0.0"></script>
    <link href="https://unpkg.com/jspsych@7.0.0/css/jspsych.css" rel="stylesheet" type="text/css" />
  </head>
  <body></body>
  <script>

    /* Je lance jsPsych et je lui dit de me montrer les data à la fin */
    var jsPsych = initJsPsych({
      on_finish: function() {
        jsPsych.data.displayData();
      }
    });

    /* Je crée la timeline de l'experience */
    var timeline = [];

    /* On va précharger les média pour éviter la latence dans le navigateur */
    var preload = {
      type: jsPsychPreload,
      images: ['img/blue.png', 'img/orange.png']
    };
    timeline.push(preload);

    /*Je définie le message d'accueuil comme un trial */
    var welcome = {
      type: jsPsychHtmlKeyboardResponse,
      stimulus: "Bienvenue dans l'experience test. Appuyez sur une touche pour commencer."
    };
    timeline.push(welcome);

    /* On montre les instructions */
    var instructions = {
      type: jsPsychHtmlKeyboardResponse,
      stimulus: `
      <p>Dans cette expérience, un cercle va apparaitre au centre de l'écran.</><p>Si le cercle est <strong>Bleu</strong>,
      <p>Si le cercle est <strong>Orange</strong>, appuyez sur la lettre J aussi vite que vous le pouvez.</p>
      <div style='width: 700px;'>
      <div style='float: left;'><img src='img/blue.png'></img>
      <p class='small'><strong>Appuyez la lettre F</strong></p></div>
      <div style='float: right;'><img src='img/orange.png'></img>
      <p class='small'><strong>Appuyez la lettre J</strong></p></div>
      </div>
      <p>Appuyez sur une touche pour commencer.</p>
      `,
     post_trial_gap: 2000

    };
    timeline.push(instructions)

    /* On défini une variable de timeline les stimulus */

    var test_stimuli = [
      { stimulus: "<div style='float: right;'><img src='img/blue.png'></img>", correct_response:'f'},
      { stimulus: "<div style='float: right;'><img src='img/orange.png'></img>", correct_response: 'j'}
    ];

    /* Maintenant on ajoute une croix de fixation entre les présentation*/

    var fixation = {
      type: jsPsychHtmlKeyboardResponse,
      stimulus: '<div style="font-size:60px;">+</div>',
      choixes: "NO_KEYS",
      trial_duration: function(){
        return jsPsych.randomization.sampleWithoutReplacement([250, 500, 750, 1000, 1250, 1500, 1750, 2000], 1)[0];
      },
```

jsPsych

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

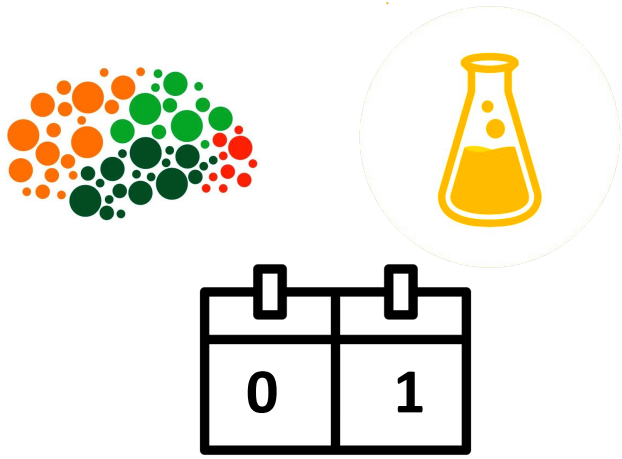
Exemple de réalisation de la même expérience sur les deux outils

Facilité de prise en main



- lab.js est plus simple à prendre en main lorsque l'on ne maitrise 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.

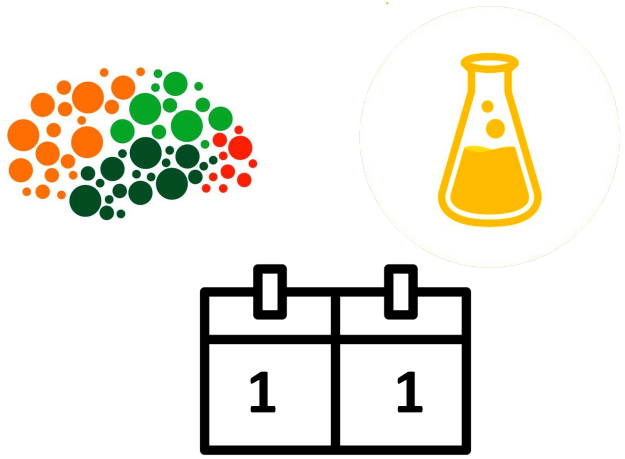

List of Plugins

These are the plugins that are included in the jsPsych release.

Additional plugins may be available in the

For an overview of what plugins are and how they work, see our

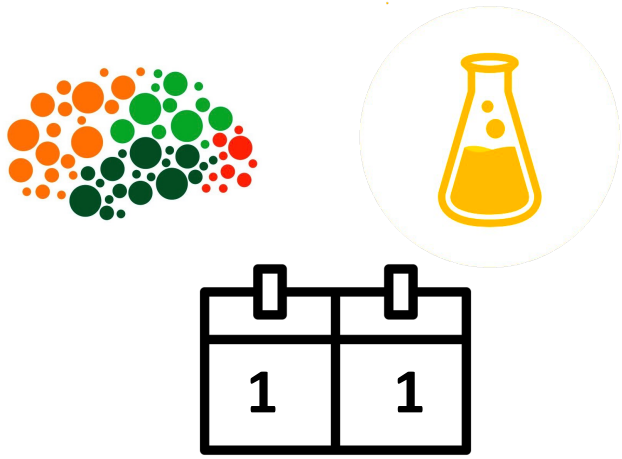
Modularité



- 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.

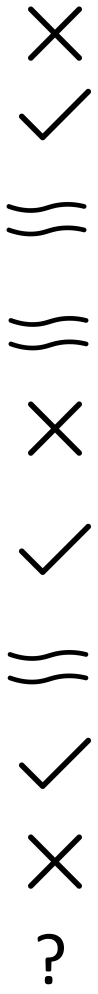


Table 5.1 Examples of questions answered with the methods described in this chapter.

Section: Method	Examples of answered questions
5.3: Psychophysical methods	– Can the user perceive each of the configurations of a sonic interaction? – Can the user differentiate between configurations?
5.4: Identification and categorization	– What naturalistic object is recognized in each of the configurations? – What emotional category is recognized in a sonic artefact?
5.5: Scaling and rating	– How does perceived effort vary between sonic feedbacks for robotic surgery applications? – How should the user-controlled gain for sound level vary so at to produce a linear increase in perceived loudness?
5.6: Dissimilarity estimation	– 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	– 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	– 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	– 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	– Which configuration of a sonic feedback system do users prefer the most? – Which configuration is the least annoying?
5.11: Continuous evaluation	– 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	– 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.

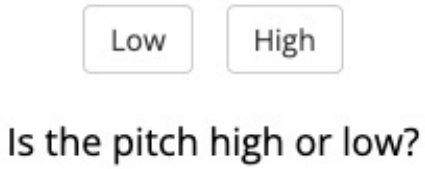
Certains plugins déjà existants permettent de faciliter la création de certains paradigmes.

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

Section: Method	Examples of answered questions	
5.3: Psychophysical methods	– Can the user perceive each of the configurations of a sonic interaction? – Can the user differentiate between configurations?	?
5.4: Identification and categorization	– What naturalistic object is recognized in each of the configurations? – What emotional category is recognized in a sonic artefact?	✓
5.5: Scaling and rating	– How does perceived effort vary between sonic feedbacks for robotic surgery applications? – How should the user-controlled gain for sound level vary so at to produce a linear increase in perceived loudness?	✓
5.6: Dissimilarity estimation	– 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	– 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	– 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	– 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	– Which configuration of a sonic feedback system do users prefer the most? – Which configuration is the least annoying?	✓
5.11: Continuous evaluation	– 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	– 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.

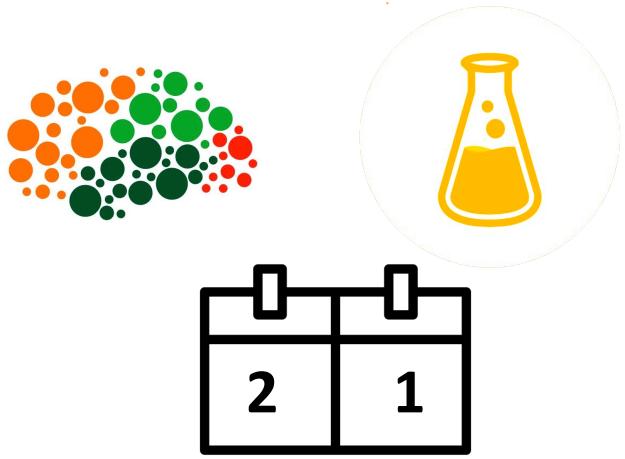
```
var trial = {
    type: jsPsychAudioKeyboardResponse,
    stimulus: 'sound/tone.mp3',
    choices: ['e', 'i'],
    prompt: "<p>Is the pitch high or low? Press 'e' for low and 'i' for high.</p>",
    response_ends_trial: true
};
```

jsPsych

- Quelques exemples visuels :


```
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



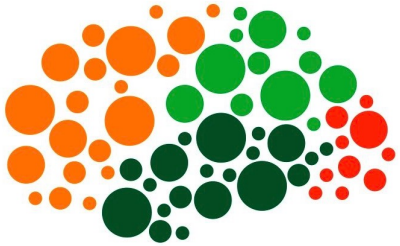
- jsPsych est plus complet pour mettre en place la plupart des paradigmes en première approximation.
- lab.js pourrait permettre de le faire, au prix de quelques détournements pour conserver la facilité de son interface.

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 HTLML 5 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 ?