

# Evaluation des outils d'experimentation en ligne et implémentation de jsPsych

Matthieu Fraticelli

5 septembre 2022

## Résumé

Ce document est un résumé de la mission effectuée au sein de l'équipe Perception et design sonores du laboratoire STMS de l'Ircam.

Son but est de dresser l'historique de la démarche, sa méthodologie, son débouché ainsi que les points à approfondir.

## Contact

Adresse mail : `fraticelli.matthieu@hotmail.fr`

tel : 06 37 91 55 07

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Un peu de contexte . . . . .	4
1.2	Organisation du document . . . . .	4
<b>2</b>	<b>Évaluation des outils d'expérimentation en ligne</b>	<b>4</b>
2.1	Les critères d'évaluation pour l'état des lieux . . . . .	4
2.2	Les différents outils évalués . . . . .	5
2.2.1	Fournisseur de service . . . . .	5
2.2.2	Builder . . . . .	6
2.2.3	Librairie open source . . . . .	6
2.2.4	Software . . . . .	6
2.2.5	Service d'hébergement spécialisé . . . . .	7
2.3	Résultats de l'évaluation . . . . .	7
<b>3</b>	<b>Sélection d'un seul outil</b>	<b>8</b>
3.1	lab.js . . . . .	8
3.2	jsPsych . . . . .	8
3.3	Sélection finale . . . . .	9
<b>4</b>	<b>Intégration au sein de l'équipe Perception et Design Sonores</b>	<b>9</b>
4.1	Solution d'hébergement . . . . .	9
4.1.1	Jatos . . . . .	9
4.1.2	Mindprobe . . . . .	9
4.1.3	Cognition.run . . . . .	9
4.1.4	Serveur Ircam . . . . .	10
4.2	Documentation, démonstrateur et mise à disposition des chercheurs . . . . .	10
4.2.1	Organisation . . . . .	10
4.2.2	Utilisation . . . . .	11
4.2.3	Contribution . . . . .	11
4.3	Problématiques et possibilités pour le futur . . . . .	11
4.3.1	Gestion d'un serveur . . . . .	11
4.3.2	Standardisation des tests d'écoute . . . . .	11

## Remerciements

Je tiens à remercier tout particulièrement l'équipe Perception et design sonores du laboratoire STMS de l'IRCAM, qui m'ont donné la chance de pouvoir collaborer avec eux sur le sujet passionnant de l'expérimentation en ligne, mais plus généralement qui m'ont permis de découvrir le monde de la recherche et de pouvoir m'y plonger pour la première fois.

Je leur exprime mon infinie gratitude pour tout ce que j'ai appris à leur côtés et tout ce que je pourrais continuer à apprendre grâce à cela.

Merci pour leur confiance, leur bienveillance et l'amicale exigence qui m'a stimulé tout au long de cette mission.

Matthieu.

# 1 Introduction

## 1.1 Un peu de contexte

La pratique de la méthode expérimentale se situe essentiellement dans le cadre du laboratoire. Les contraintes expérimentales, mais également techniques, exigeants un environnement particulier.

Mais les contraintes imposées par les passations des participants, ainsi qu'un besoin de panels importants pouvaient limiter la capacité des chercheurs à déployer certaines expériences.

Peu à peu, le développement d'internet et l'omniprésence des navigateurs sur les ordinateurs, ont attirés l'attention des chercheurs qui y ont vu une opportunité d'atteindre un nombre plus important de personnes. Cependant, l'expérimentation en ligne pose un certain nombre de questions à résoudre : l'hébergement de l'expérience, le langage employé, la compatibilité du mode de passation avec la rigueur scientifique etc.

L'épidémie de Covid-19 a accéléré ce processus et nombre d'expériences se sont vues déployées sur internet en raison des contraintes de déplacement.

L'objectif de ce document est de consigner la démarche de référencement qui a été l'objet de ma mission au sein de l'équipe Perception et Design Sonores du laboratoire STMS de l'Ircam.

## 1.2 Organisation du document

La démarche de ce document sera la suivante :

- Dans un premier temps, nous reviendrons sur les critères ayant guidés l'évaluation des outils.
- Ensuite, nous détaillerons les outils ayant été évalués et comment ils s'insèrent dans le paysage de l'expérimentation en ligne.
- Enfin dans une seconde partie, nous parlerons du choix qui a été fait de se concentrer sur jsPsych.

# 2 Évaluation des outils d'expérimentation en ligne

La pratique de l'expérimentation en ligne s'étant rapidement développée, les outils disponibles pour déployer ses expériences se sont rapidement multipliés.

Nous avons donc un nombre important d'outils, de bibliothèques, de techniques utilisées, qu'il nous faut inventorier et passer au crible d'une liste de critères particuliers.

## 2.1 Les critères d'évaluation pour l'état des lieux

Les critères utilisés pour évaluer les outils sont les suivants :

- La capacité à reproduire les paradigmes expérimentaux classiques utilisés au sein de l'équipe,
- La facilité de prise en main,
- La modularité permise,
- La longévité supposée de l'outil.

Concernant les principaux paradigmes expérimentaux utilisés dans l'équipe, nous nous sommes basés sur l'ouvrage *Sonic Interaction Design* (Franovic Serafin, 2012) en particulier le Chapitre 5 *Perceptual evaluation of sound-producing objects* (Giordano, Susini, Bresin).

Le tableau suivant est une reproduction de l'ouvrage :

Paradigmes expérimentaux classiques		
Nom famille de paradigme	Description	Code
Méthodes psychophysiques	e.g. jnd, détection de seuil, ...	1
Identifications et catégorisations	Identifier ou catégoriser des stimuli.	2
Notation et échelle	Noter un son sur une échelle, ou un ratio, par rapport à un attribut.	3
Estimation de dis-similarité	Evaluer à quel point deux sons sont semblables ou non.	4
Classement	Classer des sons, de façon libre ou selon des critères donnés.	5
Verbalisation	Recueillir des verbatims concernant des sons, librement ou par questions.	6
Différentiel sémantique	Noter un son selon un descripteur sémantique.	7
Estimation de la préférence	Juger de sa préférence entre deux stimuli par paire, ou classement.	8
Évaluation continue	Recueillir un avis du sujet de façon continue pendant la lecture du son.	9

Ils ne se veulent pas neutres, mais sont adaptés aux besoins de l'équipe, de la nécessité de produire les principaux paradigmes utilisés par les chercheurs.

## 2.2 Les différents outils évalués

Intéressons nous à présent aux différents outils qui ont été évalués. Ces outils peuvent être classés dans des catégories qui sont les suivantes :

- Fournisseur de service,
- Librairie open source,
- Software,
- Service d'hébergement spécialisé.

Entrons plus en détails dans leurs caractéristiques et listons les outils que nous avons rencontrés pour chacun de ses catégories.

### 2.2.1 Fournisseur de service

Par fournisseur de service nous désignons une entreprise proposant un produit permettant de réaliser des évaluations en ligne.

Généralement, ces fournisseurs de service propose un site internet où il est possible d'effectuer un bref design experimental en utilisant des templates mis à disposition ou bien de personnaliser directement son expérience dans une certaine mesure.

Voici ceux que nous avons répertoriés :

- **Qualtrics**,  
<https://www.qualtrics.com/>
- **Gorilla.sc**,  
<https://gorilla.sc/>
- **Testable**,  
<https://www.testable.org/>
- **Inquisit Web**,  
<https://www.millisecond.com/products/web>
- **Labvanced**,  
<https://www.labvanced.com/>

- **PsyTest**,  
<https://psytests.be/>

### 2.2.2 Builder

Les builders fonctionnent comme des fournisseurs de services, mais ne sont pas payants. Ils proposent un environnement graphique pour développer ses expériences sans manipuler beaucoup de code.

Cette solution est donc avantageuse pour le chercheur qui ne peut pas développer son expérience dans des principaux langages utilisés.

Voici ceux que nous avons répertoriés :

- **PsyToolkit**,  
<https://www.psychtoolbox.org/>
- **NodeGame**,  
<https://nodegame.org/>
- **Tatool**,  
<https://www.tatool-web.com/#!/>
- **LionessLab**,  
<https://lioness-lab.org/>
- **O-Tree**,  
<https://www.otree.org/>
- **labJs**,  
<https://lab.js.org/>
- **OpenSesame**,  
<https://osdoc.cogsci.nl/>
- **R-Shiny**,  
<https://designingexperiments.com/shiny-r-web-apps/>
- **Experiment Factory**,  
<https://www.expfactory.org/>
- **PsychStudio**,  
<https://www.psychstudio.com/>
- **Experimental Tribe**,  
<http://www.xtribe.eu/>

### 2.2.3 Librairie open source

Une librairie est une ressource écrite dans un certain langage de programmation, permettant d’avoir accès à des fonctions plus ”haut niveau” afin de ne pas devoir tout coder pour un usage précis.

Voici ceux que nous avons répertoriés :

- **PsychoPy**,  
<https://www.psychopy.org/>
- **PsychoJs**,  
<https://github.com/psychopy/psychojs>  
(Online counterpart of psychoPy)
- **jsPsych**,  
<https://www.jspsych.org/7.3/>

### 2.2.4 Software

Enfin, un logiciel est un produit complet, dont les fonctionnalités permettent de réaliser des expériences, dont certaines en ligne.

Voici ceux que nous avons répertoriés :

- **SuperLab**,  
<https://cedrus.com/superlab/index.htm>
- **PEBL**,  
<http://pebl.sourceforge.net/>

### 2.2.5 Service d'hébergement spécialisé

Un service d'hébergement spécialisé occupe une place particulière dans notre taxonomie. En effet, il ne propose pas de programmer une expérience en ligne *per se*, mais il permet d'héberger la dite expérience en ligne.

Le service met donc à disposition un serveur pour le déploiement de l'expérience, de façon payante ou non.

Voici ceux que nous avons répertoriés (certains étaient également présents dans les autres catégories) :

- **Utiliser un serveur interne**,
- **JATOS**,
- **MindProbe**,
- **Experiment Factory**,
- **Cognition.run**,

Il serait également pertinent de classer les outils selon le langage qu'ils utilisent. Cette classification est moins pertinente me semble-t-il que celle proposée précédemment, en raison de la grande différence des types d'outils : certains n'étant pas à proprement parlé basé sur un langage de programmation en raison d'une interface graphique poussée, ou autre.

Nous nous contenterons de lister dans le tableau suivant, les langages utilisés par chacun des outils (hormis ceux proposant une interface graphique se passant de langage de programmation).

## 2.3 Résultats de l'évaluation

Voici le résultat de l'évaluation. Ce tableau permet de lire la capacité des outils à produire les différents paradigmes énoncés plus haut, en utilisant le code qui leur avait été attribué.

Lorsque l'information n'était pas disponible notamment quand l'outil était payant, un " ? " est utilisé.

Résultats de l'évaluation selon les critères choisis												
Nom de l'outil	Type	Langage	Host	1	2	3	4	5	6	7	8	9
Qualtrics	Service provider	/	Provided		X	X			X	X	X	
Gorilla.sc	Service provider	/	Provided		X	X			X	X	X	
Testable	Service provider	/	Provided	?	?	?	?	?	?	?	?	?
Inquisit Web	Service provider	/	Provided	?	?	?	?	?	?	?	?	?
Labvanced	Service provider	/	Provided		X	X	X	X	X	X	X	
PsychoPy	Librairie	Py	Pavlovica	X	X	X	X	X	X	X	X	X
PsychoJS	Librairie	Py/js	Pavlovica	X	X	X	X	X	X	X	X	X
jsPsych	Librairie	js	Choosable		X	X	X		X	X	X	
Psytoolkit	Builder	/	Provided			X	X		X	X	X	
NodeGame	Builder	js	Provided									
Tatool	Builder	js	Provided									
LionessLab	Builder	/	Provided									
O-Tree	Builder	Py	Provided									
LabJs	Builder	js	Choosable		X	X		X	X	X	X	
OpenSesame	Builder	Py/js	Provided		X	X	X	X	X	X	X	
R-Shiny	Librairie	R	Choosable									
Experiment Factory	Builder	/	Provided									
PsychStudio	Builder	/	Provided		X	X	X	X	X	X	X	
SuperLab	Software	/	Remote	X	X	X	X	X	X	X	X	
Experimental Tribe	Builder	Py/js	Provided									
PEBL	Software	/	Choosable									

Certaines informations étaient également indisponibles lorsque les outils n'étaient pas directement destinés à l'audio.

### 3 Sélection d'un seul outil

Basé sur ces résultats et différentes réunion d'équipe, nous avons choisi de sélectionner deux outils. Cette sélection a permis de retenir **lab.js** et **jsPsych**.

Ce choix a été fait en raison de l'utilisation du javascript comme langage de développement, qui était naturellement celui le plus adapté pour une utilisation en ligne sur le web et aussi en raison de la capacité de ces deux outils à produire beaucoup des différents paradigmes utilisés.

#### 3.1 lab.js

Un lien vers lab.js :

<https://lab.js.org/>

lab.js est un un des outils classés dans la catégorie des **builder**, c'est à dire qu'il propose une interface graphique en ligne pour venir designer les expériences.

#### 3.2 jsPsych

Un lien vers jsPsych :

<https://www.jspsych.org/7.2/>

jsPsych est quant à lui un des outils classés dans la catégories des **librairies**, c'est à dire qu'il s'agit d'une sur-couche javascript (plus précisément en typescript) qui permet d'avoir accès à un niveau de contrôle plus macroscopique, dédié à la création d'expériences en ligne.



### 3.3 Sélection finale

Après une comparaison poussée au niveau des deux outils sélectionnés, il a été décidé de choisir **jsPsych** comme outil principal pour l'expérimentation en ligne au sein de l'équipe.

Ce choix est dû à une plus grande souplesse de jsPsych pour la modularité et un fonctionnement orienté directement vers l'utilisation de plugins. Même si sa prise en main peut paraître plus difficile au départ en raison d'une connaissance requise du javascript, la comparaison entre les deux a mis en évidence une plus grande complexité expérimentale possible, avec une personnalisation plus grande. Il était également possible de modifier certains paradigmes sur lab.js, mais au prix de retouches de code importante, qui rendaient alors nul l'avantage face à jsPsych.

Mais, n'étant qu'une librairie sans solution d'hébergement en elle-même, nous discuterons plus loin des choix qui ont été faits à ce niveau là, ainsi que des différentes possibilités offertes pour un déploiement futur.

## 4 Intégration au sein de l'équipe Perception et Design Sonores

L'intégration de jsPsych au sein de l'équipe s'est déroulée en deux étapes. Dans un premier temps, il a été mené un travail de développement sur l'outil, afin d'appliquer la librairie à un besoin immédiat des doctorants et chercheurs. Cette phase a été marquée par de nombreux aller-retours au sein de l'équipe, pour commencer à mettre en place une méthodologie de portage en ligne. Ces échanges et les connaissances qui en ont résultées ont permis de définir un guide des bonnes pratiques qui sera détaillé plus loin, ainsi que de déterminer le cadre qui sera le plus utile aux chercheurs pour bénéficier des connaissances acquises durant cette mission.

Dans cette partie nous allons détailler les aspects pratiques du déploiement de jsPsych au sein de l'équipe PDS.

### 4.1 Solution d'hébergement

jsPsych est une librairie JavaScript. En elle-même, il est possible de la télécharger pour s'en servir en local, ou bien de l'héberger sur un serveur pour s'en servir directement en ligne.

Contrairement à certains outils que nous avons détaillés plus haut, jsPsych ne propose pas de service d'hébergement. Un rapide tour d'horizon des solutions qui s'offraient à nous s'est imposé et a débouché sur différentes possibilités.

#### 4.1.1 Jatos

Un lien vers Jatos :

<https://www.jatos.org/>

Non retenue dans la sélection finale.

#### 4.1.2 Mindprobe

Un lien vers Mindprobe :

<https://mindprobe.eu/>

Non retenue dans la sélection finale.

#### 4.1.3 Cognition.run

— Un lien vers Cognition.run : <https://www.cognition.run/>

Cognition.run est une solution d'hébergement intéressante car elle a été spécialement conçue pour accueillir des expériences qui ont été développées sous jsPsych. Le service permet à la fois de déployer des expériences en ligne de façon simple, mais dispose également d'avantages rendant son utilisation simple pour les chercheurs peu habitués au code.

Cognition.run héberge le code JavaScript et uniquement ce code-ci. Il n'est donc pas nécessaire de créer un document HTML classique qui servirait de visualisation à l'expérience, ni de procéder d'une façon plus globalement à un développement web poussé. Il faut se contenter d'insérer le code

contenu entre nos habituelles balises HTML `<script>`, de choisir la version de la librairie jsPsych concernée, et de mettre en ligne.

Un gain de temps considérable est donc obtenu dans le fait qu’il est possible de déployer rapidement une expérience, sans se préoccuper particulièrement des packages de jsPsych et des éléments qu’il faudrait mettre en place pour déployer une expérience de façon classique.

- Fenêtre de pré-visualisation

Cognition offre en plus d’une solution d’hébergement classique, une interface graphique ressemblant à un IDE.

Il est en effet possible d’éditer son code en ligne, en prévisualisant le résultat sous la forme d’un affichage dynamique sur la fenêtre de droite.

La même fenêtre permet également de choisir la version de jsPsych utilisée, des stimuli que l’on souhaite importer, mais également des éléments additionnels en CSS ou JS.

Si l’on utilise le DevTools proposé par Google, ou d’une manière plus générale la console du navigateur, il est possible de développer une expérience directement en ligne, mais aussi de procéder à d’éventuels débugs en visualisant directement les erreurs renvoyées par le navigateur.

- Gestion des données expérimentales

Une expérience en ligne classique nécessite de prendre en compte la récupération des données. Il faut donc coder une partie pour récupérer sur le serveur les différents fichiers .csv obtenus par jsPsych.

Avec Cognition.run, le service propose directement de naviguer dans les résultats en ligne, de voir si les participants ont tous fini l’expérience, d’avoir des informations intéressantes pour permettre au chercheur de gagner du temps pour ordonner ses données. Il est possible enfin de télécharger l’intégralité, d’une façon intuitive.

- Couplage avec Prolific

Cognition.run offre un service intuitif pour être utilisé avec Prolific (<https://www.prolific.co/>) pour gérer la gratification des participants.

#### 4.1.4 Serveur Ircam

Une autre solution possible est d’utiliser directement un serveur dédié à l’Ircam, qui permettrait une gestion sur-mesure de n’importe quel cas d’utilisation.

En particulier afin de respecter la législation RGPD concernant les données personnelles, il serait possible de maîtriser complètement le serveur et l’utilisation des données qui en est faite.

Un autre avantage offert par cette solution serait de pouvoir gérer le trafic des participants sur les expériences.

Par exemple, il serait possible d’héberger trois versions du code pour une même expérience qui correspondraient à trois conditions expérimentales et d’utiliser un outil pour s’assurer que les trois conditions reçoivent le même nombre de participants, ce qui n’est pas aussi simple à faire directement dans le code d’une expérience (répartition aléatoire qui risque de ne pas satisfaire au nombre requis sur de petits échantillons).

## 4.2 Documentation, démonstrateur et mise à disposition des chercheurs

Un démonstrateur est actuellement disponible par le biais d’un serveur Github.

En voici le lien : <https://github.com/MatthieuFra/jsPsychPDS>

Nous allons à présent détailler l’organisation de ce dépôt, ainsi que la façon de l’utiliser et d’y contribuer.

### 4.2.1 Organisation

Le dépôt s’organise en différentes parties de la façon suivante :

1. Démonstrateur
2. Tutoriels

Entrons plus en détail dans chacune de ces parties.

- Démonstrateur

Le démonstrateur est la partie destinée à montrer les expériences développées au sein de l'équipe, mais aussi pour montrer aux chercheurs eux mêmes quelles sont les briques de code élémentaires qui sont mises à leur disposition de façon à construire des expériences en ligne, ou bien pour développer leur paradigmes en ligne.

Le démonstrateur est aussi une vitrine qui permet de mettre en valeur les paradigmes classiques de l'équipe, et de rendre disponible à la communauté de chercheurs les codes correspondants.

Le démonstrateur est un site internet statique hébergé sur GitHub qui utilise GitHub Pages. En voici le lien : <https://matthieufra.github.io/jsPsychPDS/>.

— Tutoriels

La partie dédiée aux tutoriels permet à l'équipe de mieux s'approprier la ressource offerte par le démonstrateur jsPsych.

Cette partie contient des liens vers des ressources vidéos qui concernent jsPsych, mais également des notes écrites concernant la plate-forme de démonstration.

Il est également possible d'y trouver ce résumé en PDF, ainsi que les différentes présentations orales données lors de la mission, accessibles ici : [https://github.com/MatthieuFra/jsPsychPDS/tree/main/pdf/oral\\_presentations/](https://github.com/MatthieuFra/jsPsychPDS/tree/main/pdf/oral_presentations/)

#### 4.2.2 Utilisation

Pour utiliser la ressource, le plus simple est de passer directement par le site proposé plus haut, et de parcourir les différents modules qu'il propose. Une section de tutoriels est disponible pour aider le novice à appréhender jsPsych, mais il y a également une page entière de démonstrateur que l'on peut essayer directement sur le site, et dont le code est directement accessible pour y accéder. Cela rendra la navigation plus fluide et permettra directement d'utiliser ce qui est nécessaire.

#### 4.2.3 Contribution

Pour contribuer au dépôt GitHub de l'équipe, il sera nécessaire de mettre en place un système de review internet, associé à des pull request sur Git, afin de permettre un fonctionnement commun à l'équipe et le travail collaboratif pour la mise en ligne des expériences.

Il est probable que malgré l'augmentation des connaissances sur la librairie jsPsych, il faille qu'une personne se tienne plus régulièrement au courant des avancées de la librairie, et du maintien en compatibilité des différents paradigmes.

Afin de contribuer, un fichier Readme sera disponible sur l'utilisation des briques complémentaires, et un template pour coder ses expériences ainsi que ses plugins sera disponible.

### 4.3 Problématiques et possibilités pour le futur

Il a été possible au cours de cette mission d'avancer sur la problématique de l'expérience en ligne, de définir des bonnes pratiques ainsi qu'une charte expérimentale cohérente au sein de l'équipe.

Cependant, certains points demanderont un approfondissement Nous nous proposons d'en discuter certains, émis à la fin de la mission.

#### 4.3.1 Gestion d'un serveur

L'utilisation actuelle de Cognition.run présente des avantages importants, en terme de facilité d'utilisation et de déploiement.

Mais, le service est devenu payant durant la mission, il est donc important de reconsidérer ces avantages et envisager éventuellement l'utilisation d'un serveur Ircam comme spécifié plus haut.

#### 4.3.2 Standardisation des tests d'écoute

L'expérimentation en ligne pose certaines problématiques. Malgré le fait qu'elle permette l'utilisation de panels très importants, spécifiques et représentatifs, une attention doit être posée quant aux conditions expérimentales.

En effet les participants sont seuls à passer l'expérience devant leurs ordinateurs (dans le cadre d'une expérience réellement en ligne, l'utilisation de jsPsych comme langage pour faire passer des expérience en local au laboratoire ne pose pas ce genre de problème).

Il convient alors de s'assurer de différents points :

1. Maîtrise au connaissance des conditions expérimentales
2. Conservation de la motivation,
3. Bonne compréhension des consignes,
4. Respect des consigne au fil du temps,

Ces différents points sont au coeur d'une réflexion qu'il est nécessaire de mener avant toute démarche d'expérience en ligne afin d'éviter des résultats bruités, ou de gâcher un batch expérimental.