

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

Matthieu Fraticelli

July 25, 2022

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 certains 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 à 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 à 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 XXXXXX.

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

2.2.2 Librairie open source

2.2.3 Software

2.2.4 Service d'hébergement spécialisé

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 une langage de programmation en raison d'une interface graphique poussée, ou autre.

Nous nous contenterons de lister sous forme du tableau suivant, les langages utilisés par chacun des outils :

INSERT TABLE

2.3 Résultats de l'évaluation

Voici le résultat de l'évaluation.

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**.

3.1 lab.js

Un lien vers lab.js :

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

3.2 jsPsych

Un lien vers jsPsych :

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

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. 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 déroulé 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/>

4.1.2 Mindprobe

Un lien vers Mindprobe :

<https://mindprobe.eu/>

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>` et `</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 débogs 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. Construction toolkit

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/>.

4.2.2 Utilisation

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 expériences 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 cœur 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.

Nous allons nous attarder plus spécifiquement en ce qui nous concerne, sur un point ayant trait à la maîtrise des conditions expérimentales : les conditions d'écoute.

• Maîtrise des conditions d'écoute

Dans les expériences qui impliquent l'audio, il est crucial de maîtriser certains éléments comme le niveau d'écoute, sa constance au cours du temps, le système utilisé, le niveau de bruit ambiant, ...

Ces éléments sont facilement maîtrisables dans des conditions

4.3.3 Gestion de la bonne passation des expériences en ligne

5 Ircam PDS toolkit for jsPsych

6 Remerciements

je tiens à remercier

References