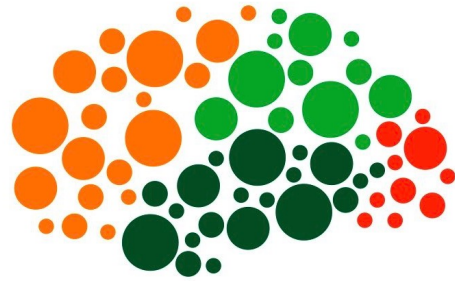


# jsPsych pour l'expérimentation en ligne



jsPsych

# Résumé de la réunion précédente

- Comparaison de js.lab et jsPsych
  - Facilité de prise en main pour lab.js
  - Manque de possibilités pour lab.js
- Réduction à un seul outil : *jsPsych*

# Résumé de la réunion précédente

- Test de l'outil d'interface graphique,
- Mise en situation de l'outil dans différentes situations :
  - Expérience BWS,
  - Expérience type adaptative en psychoacoustique,
  - Regarder à l'intérieur des plugins

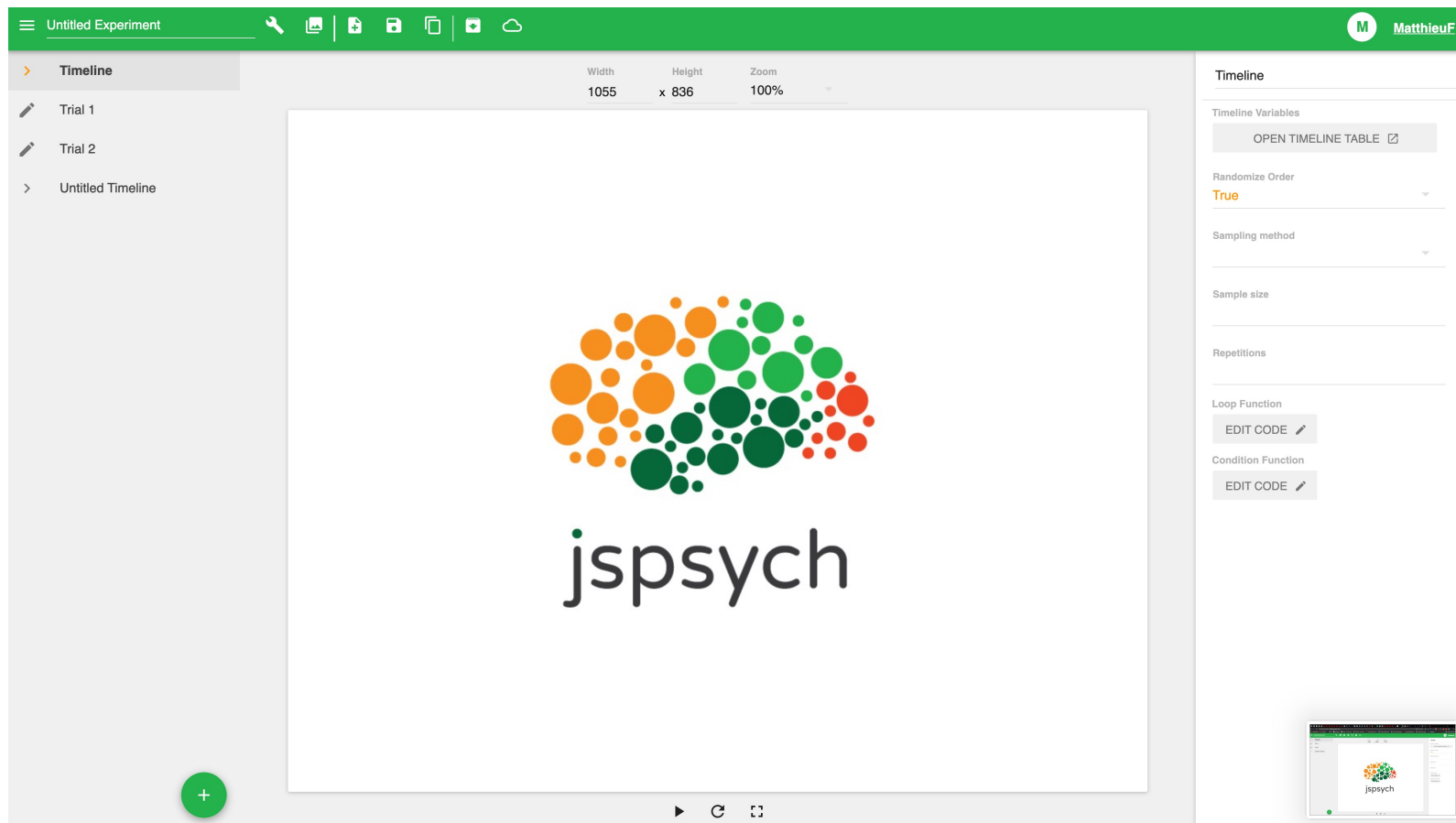
# Questions de la précédente réunion

- 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 ?



# jsPsych - builder

Test de la GUI proposée par la communauté jsPsych





# jsPsych - builder

## jsPsych.init properties

default_iti:	0
on_finish	<>
on_trial_start	<>
on_trial_finish	<>
on_data_update	<>
on_interaction_data_update	<>
show_progress_bar	<input type="radio"/>
auto_update_progress_bar	<input checked="" type="radio"/>
show_preload_progress_bar	<input checked="" type="radio"/>
preload_audio:	
preload_images:	
max_load_time (ms):	60000
Exclusions:	
min_width:	0
min_height:	0

## Untitled Timeline

### Timeline Variables

OPEN TIMELINE TABLE

### Randomize Order

True

### Sampling method

### Sample size

### Repetitions

### Loop Function

EDIT CODE

### Condition Function

EDIT CODE

## Untitled Trial

Plugin: audio-keyboard-response

### Stimulus

Select Media

This parameter is required.

### Choices

[ALL KEYS]

### Prompt

### Trial duration

### Response ends trial

True

### Trial ends after audio

False

### Data

EDIT OBJECT

### On start

[UNDEFINED]

### On finish

[UNDEFINED]

### On load

[UNDEFINED]

### Post trial gap



# jsPsych - builder

- + Facilité de prise en main ,
- + Peu besoin de code,
- Version en bêta,
- Nécessité d'entrer du code pour certains réglages,
- Fonctionnel uniquement sur la librairie de base,



# Cognition.run

**Cognition.**

[Features](#)

[FAQ](#)

[Go to Tasks](#)

Run cognitive  
experiments **online.**

Focus on science, not on IT.

[Create an account](#)

[Features](#)







# Cognition.run

## Link

Share this link with your participants.

<https://btbhf5km5s.cognition.run>

## Design

Edit your task paradigm, submit your stimuli and define the Informed Consent.

[Configuration](#) [Source code](#) [Informed consent](#) [Collaborators](#)

## Data collection

Manage the data generated by runs.

There are no records to display. Once a participant visits the task's link, this is where you'll be able to see and download the data.



# Cognition.run

Petite visite de cognition.run



# Cognition.run

- Facilité de prévisualisation
- Facilité de debug
- Code légèrement simplifié
- Possibilité d'ajout de librairie externes
- Fonctionnel avec toutes les versions de jsPsych
- Compatible avec Prolific
- Gratuit



# Mindprobe

- Serveur gratuit pour utiliser JATOS
- Compatible avec jsPsych





# jsPsych - BWS

Mise en situation de l'outil pour le Best-Worst-Scaling



# jsPsych - BWS

- Besoin :
  - Faire écouter 4 sons par le sujet,
  - Puis questionnaire pour déterminer *Best / Worst*,



# jsPsych - BWS

- Besoin :
  - Faire écouter 4 sons par le sujet, autant qu'il le veut,
  - Puis questionnaire pour déterminer *Best / Worst*,



# jsPsych - BWS

- Besoin :
  - Faire écouter 4 sons par le sujet, autant qu'il le veut,
  - Puis questionnaire pour déterminer *Best / Worst*,
  - Recommencer avec 4 autres sons,





# jsPsych - BWS

- Besoin :
  - Faire écouter 4 sons (de la première séquence) par le sujet, autant qu'il le veut,
  - Puis questionnaire pour déterminer *Best / Worst*,
  - Recommencer avec les 4 autres sons de la seconde séquence,



# jsPsych - BWS

Expérience :

- Ecoute 4 sons,
- Questionnaire.



# jsPsych - BWS

Expérience :

- Ecoute 4 sons *autant qu'il veut*
- Questionnaire



# jsPsych - BWS

Expérience :

<b>htmlButtonResponse</b>	4 choix de sons
<b>audioKeyboardResponse</b>	Lecture du son correspondant

- Questionnaire



# jsPsych - BWS

Expérience :

<b>htmlButtonResponse</b>	4 choix de sons + <i>continuer</i>	
<b>audioKeyboardResponse</b>	Lecture du son correspondant	<i>cond.</i>

- Questionnaire



# jsPsych - BWS

Expérience :

**htmlButtonResponse**      4 choix de sons + *continuer*

**audioKeyboardResponse**      Lecture du son correspondant      *cond.*

- Questionnaire



# jsPsych - BWS

Expérience :

<b>htmlButtonResponse</b>	4 choix de sons + <i>continuer</i>	
<b>audioKeyboardResponse</b>	Lecture du son correspondant	<i>cond.</i>
<b>surveyMultiChoice</b>	Deux questions.	



# jsPsych - BWS

Séquence .json

Expérience :

**htmlButtonResponse** 4 choix de sons + *continuer*

**audioKeyboardResponse** Lecture du son correspondant *cond.*

**surveyMultiChoice** Deux questions.





# jsPsych - BWS

Séquence .json

Expérience :

<b>htmlButtonResponse</b>	4 choix de sons + <i>continuer</i>	
<b>audioKeyboardResponse</b>	Lecture du son correspondant	<i>cond.</i>
<b>surveyMultiChoice</b>	Deux questions.	
<b>htmlButtonResponse</b>	Message d'erreur	<i>cond.</i>



# jsPsych - BWS

Séquence .json

Expérience :

<b>htmlButtonResponse</b>	4 choix de sons + <i>continuer</i>	
<b>audioKeyboardResponse</b>	Lecture du son correspondant	<i>cond.</i>
<b>surveyMultiChoice</b>	Deux questions.	
<b>htmlButtonResponse</b>	Message d'erreur	<i>cond.</i>

Message de fin d'expérience etc.

Data .json



# jsPsych - BWS

Présentation de l'expérience



# jsPsych - psychoacoustique

- Plugin développé par un chercheur :

Kuroki, D. *A new jsPsych plugin for psychophysics, providing accurate display duration and stimulus onset asynchrony*. Behav Res 53, 301–310 (2021). <https://doi.org/10.3758/s13428-020-01445-w>

- Possibilité de contrôler précisément les temps de présentation,
- Possibilité de jouer plusieurs sons en même temps,
- ...



# jsPsych - psychoacoustique

Expérience :

- ?



# jsPsych - psychoacoustique

Expérience :

**jsPsychPsychophysics**

Jouer le stimulus + réponse



# jsPsych - psychoacoustique

Expérience :

**jsPsychPsychophysics**

Jouer le stimulus + réponse

**htmlKeyboardResponse**

*choices: [NOKEYS]*



# jsPsych - psychoacoustique

Expérience :

**jsPsychPsychophysics**

Jouer le stimulus + réponse

**htmlKeyboardResponse**

*choices: [NOKEYS]*

*on\_start: function() {*

*if(response = false) {*

*stimulus = stimulus.up },*

*else if(response = true) {*

*stimulus = stimulus.down*

*}*

*};*





# jsPsych - psychoacoustique

Expérience :

**jsPsychPsychophysics**

Jouer le stimulus + réponse

**htmlKeyboardResponse**

*choices: [NOKEYS]*

*on\_start: function() {*

*if(response = false) {*

*stimulus = stimulus.up },*

*else if(response = true) {*

*stimulus = stimulus.down*

*}*

*};*

*on\_trial: function() { ...*

*},*



# jsPsych – sorting / rating ...

- Plugin développé par une équipe :

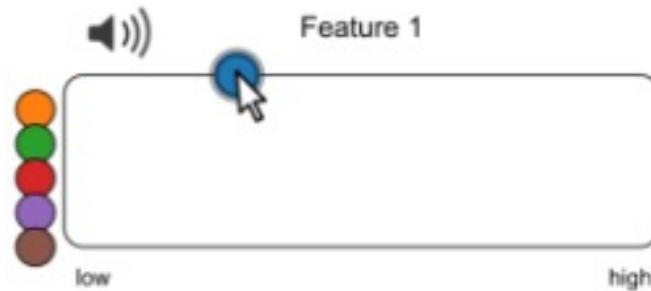
*Donhauser, P., & Klein, D. (2021, October 16). Audio-Tokens: a toolbox for rating, sorting and comparing audio samples in the browser. <https://doi.org/10.31234/osf.io/3j58q>*

- Association d'un son à une figure (un *token*),
- Possibilité de manipulation sur un espace visuel,
- Classement, échelles de notations etc.



# jsPsych – sorting / rating ...

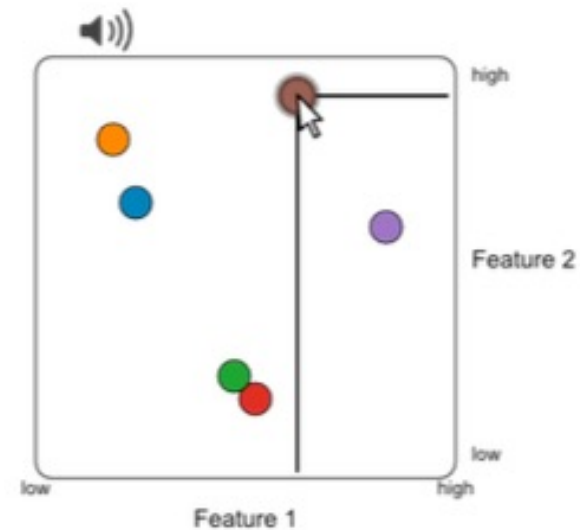
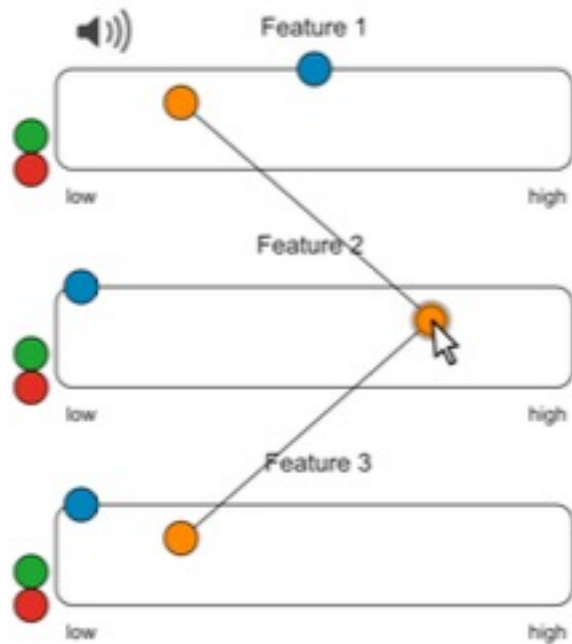
- Evaluation sur une échelle:





# jsPsych – sorting / rating ...

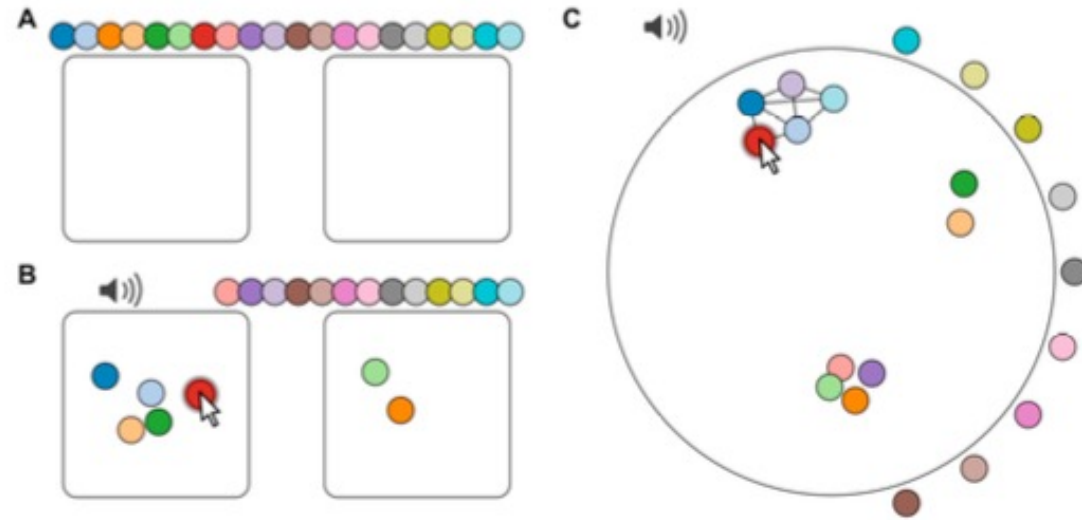
- Evaluation sur plusieurs échelles:





# jsPsych – sorting / rating ...

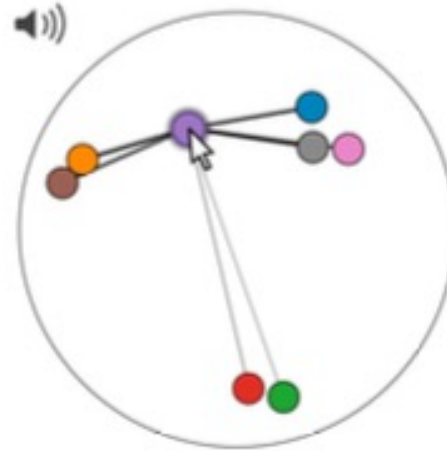
- Classement:





# jsPsych – sorting / rating ...

- Similarité:





# jsPsych – sorting / rating ...

Expérience :

- ?



# jsPsych - psychoacoustique

Expérience :

<b>jsPsychAudioTokens</b>	Jouer les stimulus + réponses
---------------------------	-------------------------------

<b>htmlButtonResponse</b> <i>choices: [« continuer »]</i> <i>data: {type : Tache 1            },</i>
--

<b>jsPsychAudioTokens</b>	Jouer les stimulus + réponses
---------------------------	-------------------------------

<b>htmlButtonResponse</b> <i>choices: [« continuer »]</i> <i>data: {type : Tache 2            },</i>
--





# jsPsych

Mise à jour du tableau.



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



# jsPsych – Plugins

- Possibilité de modifier à sa guise :
  - Mise en forme (*modification des règles css etc.*),
  - Fonctionnalité pure,
  - Mise en forme des données.



# jsPsych – Plugins

- Exemple:

## *Ecran 1*

Ecoutez ces quatres sons.

Son 1	Son 2	Son 3	Son 4	Continuer
-------	-------	-------	-------	-----------

## *Ecran 2*

Quel est le **plus** XXX\*

- ☐ Son 1
- ☐ Son 2
- ☐ Son 3
- ☐ Son 4

Quel est le **moins** XXX\*

- ☐ Son 1
- ☐ Son 2
- ☐ Son 3
- ☐ Son 4

Continue



# jsPsych – Plugins

- Exemple:

Ecoutez ces quatres sons.

Son 1 Son 2 Son 3 Son 4

Quel est le **plus** XXX\*

☐ Son 1

☐ Son 2

☐ Son 3

☐ Son 4

Quel est le **moins** XXX\*

☐ Son 1

☐ Son 2

☐ Son 3

☐ Son 4

Continue



# jsPsych – Plugins

Expérience :

**htmlButtonResponse**

« Bienvenue dans cette expérience »



# jsPsych – Plugins

Expérience :

<b>htmlButtonResponse</b>	« Bienvenue dans cette expérience »
---------------------------	-------------------------------------

<b>htmlButtonResponse</b>	<i>Instructions</i>
---------------------------	---------------------



# jsPsych – Plugins

Expérience :

<b>htmlButtonResponse</b>	« Bienvenue dans cette expérience »
---------------------------	-------------------------------------

<b>htmlButtonResponse</b>	<i>Instructions</i>
---------------------------	---------------------

<b><i>BwsAudioPlugin</i></b> <i>sequence: seq.json</i> <i>prompt: « ... »</i> <i>attribute : « Amical »</i> <i>randomization: true</i> <i>limit_trial : true/false</i> <i>data: {</i> <div></div> <i>},</i>
--



# jsPsych – Plugins

- Coder des plugins jsPsych :
  - Template dédié aux développeurs,
  - Maîtrise de la librairie jsPsych,
  - Maîtrise HTML, JS et CSS.
- Contact de Guillaume Pellerin *– réponse à suivre –*





# Approfondissement et suite

- Mise en place d'exemples pour les deux paradigmes manquants.
- Proposition :
  - Création d'un dossier de référence avec des exemples typiques par expériences.



# Approfondissement et suite

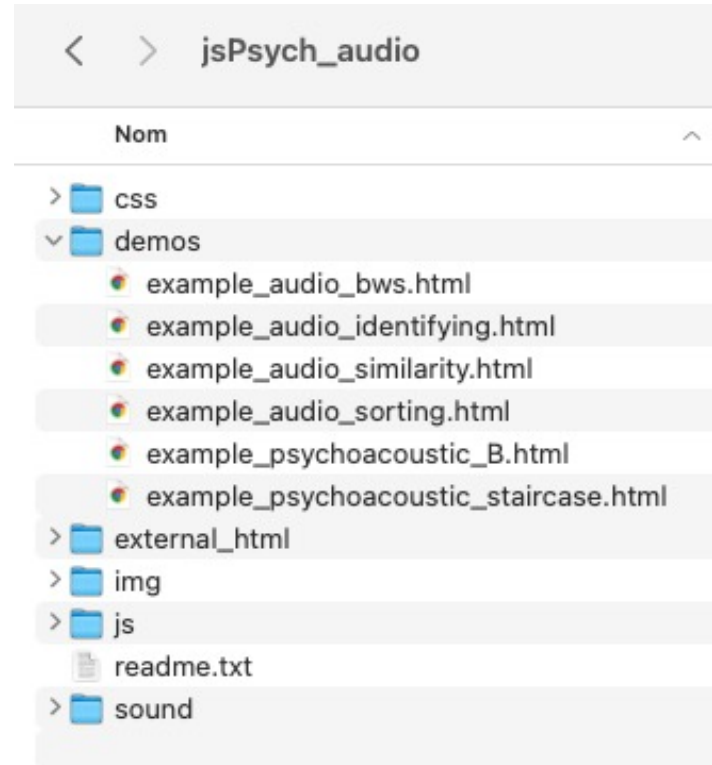
- Mise en place d'un dossier contenant des exemples fonctionnels de code pour chacun des paradigmes pouvant être pertinents.
  - Etape 1 : Utilisation de la librairie de base jsPsych et plugins codés,
  - Etape 2 : Développement d'une gamme de plugins dédiés à l'audio,



# Approfondissement et suite

- Calqué sur le dossier *example* de jsPsych, créer un dossier pour l'audio.

- jsPsych\_audio





# Approfondissement et suite

Discussion