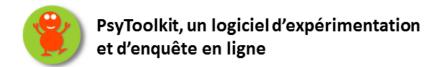
## Documentation en ligne de PsyToolkit

Gijsbert Stoet, Jean-Marc Meunier (Trad)

2021-12-22

# Contents

4 CONTENTS



Psytoolkit version 3.4.0

#### Note du traducteur

PsyToolkit est un logiciel deux-en-un permettant de faire des expériences de psychologie et de faire des enquêtes en ligne. A notre connaissance, il n'existe pas d'autre dipositif gratuit pour l'expérimentation. Pour les questionnaires, l'offre est en revanche pléthorique, alors pourquoi prendre du temps pour apprendre à utiliser un nouvau dispositif ? Parce que psyToolkit permet des choses que ne permettent pas les autres logiciels d'enquête en ligne :

- aléatorisation de l'ordre des items dans une question, des questions ou blocs de questions
- enregistrement des temps de réponse
- Calcul des variables en cours de passation et affichage de feedbacks
- affichage d'une ou plusieurs images, sons ou vidéos dans une question
- intégration d'une expérience dans un questionnaire
- génération automatique de code pour le retest
- Archivage des enquêtes et expériences à des fins de partage ou d'enseignement

Considérant ces quelques caractéristiques (et il y en a bien d'autres), il m'est apparue indispensable de former nos étudiants à un tel outil pour leur donner accès à un conception professionnelle de leurs enquêtes et expériences. C'est pourquoi j'ai entrepris la traduction de la documentation de PsyToolkit durant mes congés d'été et intégrer l'usage de PsyToolkit dans mes cours de licence et

6 CONTENTS

de master et dans mes recherches. Je n'ai pas traduit la totalité de la documentation et me suis concentré sur ce qui semblait devoir m'être utile. Certaines parties sont encore en cours de traduction et seront publiées au fur et à mesure. A chaque paragraphe, j'ai indiqué à l'aide d'une icône cliquable le lien vers le paragraphe original dans la documentation anglaise.

Je me suis permis également de structurer un peu différemment la documentation tout en conservant dans la mesure du possible les liens entre les différentes sections. Cela tient avant tout au fait que ce document est un support de cours pour moi. L'enchainement des chapitres suit donc la progression que je souhaite suivre avec mes étudiants.

Je n'ai pas la prétention de proposer une traduction parfaite, ni exhaustive. Elle comporte sûrement des erreurs et des coquilles que je corrigerai si vous voulez bien me les signaler. Je suis ouvert également à des propositions, voir à un coup de main pour la traduction (notamment pour sous-titrer les vidéos dans les leçons). Pour tout cela, vous pouvez utiliser le fil d'annotation. Pour faire des annotations, le tutoriel est ici. Chacun pourra ainsi consulter les commentaires et propositions de correction. En cas de doute ou d'incompréhension d'un passage, n'hésitez pas à le consulter ou à regarder la section originale en anglais.

La traduction des bibliothèque de questionnaires et d'expériences a été entreprise par ma collègue Elisabetta Zibetti et sera bientôt ajoutée à ce document.

Ce document est mis à disposition de la communauté avec l'accord de l'auteur et sous licence creative commons CC-BY-Nc. Cela signifie que vous pouvez l'utiliser sans le modifier, en citant l'auteur et les traducteurs et que les utilisations commerciales ne sont pas autorisées.

Pour citer ce document, utilisez la référence suivante :

• Stoet, G. (2020) Documentation de PsyToolkit (version 3.2.0), J.M. Meunier & E. Zibetti, Trad. https://jmeunierp8.github.io/Psytoolkit/

Nous rappelons par ailleurs qu'à la demande de l'auteur, pour utiliser Psy-Toolkit, vous devez citer les deux articles suivants :

- Stoet, G. (2010). PsyToolkit A software package for programming psychological experiments using Linux. Behavior Research Methods, 42(4), 1096-1104. http://link.springer.com/article/10.3758%2FBRM.42.4.1096
- Stoet, G. (2017). PsyToolkit: A novel web-based method for running online questionnaires and reaction-time experiments. Teaching of Psychology, 44(1), 24-31. https://journals.sagepub.com/doi/abs/10.1177/0098628316677643?journalCode=topa

### Chapter 1

## Présentation de PsyToolkit



PsyToolkit est une boîte à outils gratuite pour la démonstration, la programmation et l'exécution d'expériences et d'enquêtes cognitivo-psychologiques, y compris des tests de personnalité. PsyToolkit est fréquemment utilisé pour des études universitaires, des projets d'étudiants et pour l'enseignement de la psychologie cognitive et de la personnalité.

- PsyToolkit est un site web tout-en-un. Faites fonctionner votre questionnaire/expérience rapidement et facilement.
- Concevez et réalisez des expériences ou des enquêtes par questionnaire en ligne (par navigateur) ou hors ligne
- Collecte, stockage, analyse et téléchargement de données en ligne
- Pas de limite au nombre de questions ou de participants
- Interface d'enquête multilingue (comprenant l'espagnol, le français, l'allemand, , et bien d'autres)
- Documentation en ligne complète (version anglaise originale) et chaîne YouTube avec tutoriels vidéos.
- PsyToolkit est adapté à l'enseignement de la psychologie, à partir de l'enseignement secondaire
- Idéal pour les projets de recherche des étudiants et des professionnels
- Idéal pour l'aide à l'enseignement de la psychologie cognitive ou de la personnalité
- Développement rapide possible grâce à l'utilisation de bibliothèques de questionnaires et d'expériences.
- Non commercial et libre d'utilisation
- Fait pour et par des universitaires
- Hébergement fiable dans un centre de données professionnel

- Il utilise des ressources logicielles gratuites (cliquez pour les remerciements)
- Il y a une section Foire aux questions (FAQ)
- Intégration facile avec SONA si nécessaire.
- Intégration facile avec MTurk si nécessaire.
- Un large éventail de fonctionnalités
- Excellent gestion du temps comparable à celui de l'E-prime (démontré par un groupe de recherche indépendant)

#### Pourquoi utiliser PsyToolkit? 🟶 1.1



- PsyToolkit est le seul site web gratuit qui propose des expériences et des enquêtes psychologiques en ligne programmables.
- Parce que des milliers d'autres étudiants et universitaires dans le monde entier l'utilisent.
- Parce qu'il prend en charge de nombreuses langues différentes dans les enquêtes en ligne
- PsyToolkit est bien documenté avec de nombreux exemples que vous pouvez copier pour votre propre projet.
- Vous pouvez programmer votre expérience, collecter des données et les analyser avec ce seul site web.
- Vous pouvez mener des enquêtes psychologiques complexes et utiliser plus de 100 enquêtes de la bibliothèque d'enquêtes.
- Si vous utilisez également des expériences psychologiques cognitives, vous pouvez simplement les copier à partir de la bibliothèque d'expériences.
- PsyToolkit est rapide et fiable (le serveur fonctionne à partir d'un centre de données professionnel).
- PsyToolkit est géré de manière fiable par un professeur de psychologie du Royaume-Uni.

#### Environnement web/login \$\frac{1}{3}\$ 1.2



Cliquez ici pour entrer (ou vous inscrire) dans l'environnement web gratuit pour concevoir/réaliser des expériences et des questionnaires en ligne. C'est idéal pour travailler depuis chez soi et collecter des données en ligne dans le monde entier!

- Il fonctionne dans votre navigateur, sans aucun plugin et sans logiciel supplémentaire
- Vous pouvez programmer et mener des expériences en ligne

- Vous pouvez partager vos expériences avec d'autres
- Vous pouvez demander à d'autres personnes de participer en ligne, et recueillir et enregistrer des données en ligne
- Vous pouvez effectuer des enquêtes par questionnaire en ligne, des expériences en ligne ou combiner ces deux types d'activités
- Il propose des questionnaires en ligne dans de nombreuses langues, dont , le néerlandais, l'espagnol, l'allemand et le français. \*La documentation en ligne est complète et gratuite. Enregistrez votre compte gratuit dès aujourd'hui!

#### Les caractéristiques de PsyToolkit en bref 1.3



- PsyToolkit est un logiciel informatique gratuit
- 1. pour la création d'expériences psychologiques sur ordinateur (pour afficher des stimuli à l'écran, mesurer les temps de réponse, etc.)
- 2. pour la création d'enquêtes par questionnaire sur ordinateur
- 3. pour combiner les expériences cognitives et les enquêtes dans le cadre des études et de la collecte de données en ligne

Il vous permet de mener des expériences de deux manières différentes : 1. Vous pouvez créer et exécuter des études à partir d'un navigateur sur n'importe quelle plateforme informatique (Mac, Windows, Linux). Idéal pour l'enseignement et les études en ligne. 2. Vous pouvez l'exécuter sur un ordinateur Linux. Ce mode hors ligne est idéal pour les laboratoires, bien que la plupart des personnes qui le liront utiliseront probablement la version en ligne.

L'interface en ligne pour les enquêtes et les expériences en ligne prend en charge plusieurs langues.

- PsyToolkit ne coûte pas d'argent. Mais il est important de noter que PsyToolkit est la propriété intellectuelle de son créateur, le Dr Gijsbert
- PsyToolkit n'est pas un simple logiciel de type "pointer-cliquer". Pour concevoir les recherches et les expériences, vous devrez taper votre code (de nombreux exemples sont fournis). Cela nécessite une façon différente de penser et d'apprendre, mais en fin de compte, cela permet à l'utilisateur de faire preuve de beaucoup de créativité. En particulier, la mise en place de sondages en ligne reste très facile.
- PsyToolkit est développé par le professeur de psychologie Gijsbert Stoet.

#### Leçons # 1.4

Cliquez ici pour en savoir plus sur les expériences et les concepts psychologiques

cognitifs courants, et sur la manière de mettre en place des projets de recherche dans le PsyToolkit (en anglais)

- VOus trouverez des textes sur différents phémnomènes et voir à quoi ils ressemblent
- Vous trouverez également quelques leçons sur l'analyse des données
- Apprenez à créer vos propres stimuli
- Apprenez les bases de la création de votre propre questionnaire en ligne
- Apprenez les bases de la création de votre propre expérience sur le temps de réaction
- Apprenez à intégrer une expérience sur le temps de réaction dans un questionnaire en ligne
- Introduction simple à l'utilisation du logiciel de statistiques R
- Des guides complets du début à la fin sur le montage de projets (avec des tutoriels vidéo)

#### 1.5 Liens utiles



Des questions? Consultez la FAQ (en anglais).



Un questionnaire ? Consultez la bibliothèque de questionnaires.



Une expérience ? consultez la bibliothèque d'expériences.



Utilisateur Linux? Téléchargez ici la version dédiée.



courriel

Manque-t-il quelque chose dans cette documentation ? Envoyer un



Envie de supporter PsyToolkit? Voyez ici comment faire



Lisez les notes juridiques avant d'utiliser le PsyToolkit.

\_\_\_\_\_\_

## Chapter 2

# Vue d'ensemble 🟶

Cet aperçu s'adresse aux personnes qui souhaitent utiliser PsyToolkit pour mettre en place une expérience psychologique cognitive ou un questionnaire en ligne. 1. Cliquez ici pour lire une introduction de 2 minutes sur les objectifs de Psy-Toolkit

- 2. Un guide de 5 minutes pour débutants absolus pour la mise en place des études
- 3. Mise en place d'expériences (langage de script d'expérience)
- 4. Mise en place de questionnaires/enquêtes (langage de script d'enquête)
- 5. Foire aux questions



Lisez les notes juridiques avant d'utiliser le PsyToolkit

#### Expériences et enquêtes # 2.1



#### Programmation d'expériences 🚏 2.1.1

L'une des principales caractéristiques de PsyToolkit est que vous pouvez programmer des expériences de psychologie cognitive. Il existe les ressources suivantes :

Niveau	Temps de lecture	Thèmes
	iccidic	THEIRES
Facile	5 minutes	Brève introduction à l'idée des scénarios
		d'expérimentation. Lire ici
Facile	5 minutes	Comprendre la présentation des stimuli dans le
		PsyToolkit Lire ici
Avancé	15	Comment programmer le feedback pour les des
	minutes	participants dans le cadre des expériences. Lire ici
Facile	2 minutes	Partager ou envoyer une expérience à quelqu'un
		d'autre. Lire ici
Facile	2 minutes	Comment mener une expérience en ligne et collecter
		des données en ligne. Lire ici
Facile +	Voir les	Syntaxe de script détaillée, toutes les commandes
avancé	références	sont expliquées. Lire ici

#### Exemples d'expériences avec PsyToolkit 2.1.2



Souvent, il est plus facile de tirer des leçons des exemples. Il existe différentes façons de le faire : 1. Regardez les exemples de codes dans la section des leçons. 2. Regardez les exemples de code dans la bibliothèque d'expériences. 3. Si vous utilisez Linux, regardez les exemples de code qui accompagnent le package. Ils se trouvent dans /usr/share/doc/psytoolkit/3.4.0/examples/. Lisez ici comment les utiliser.

## Enquêtes/questionnaires en ligne



Vous pouvez faire plus que de simples expériences avec PsyToolkit. Vous pouvez mettre en place des enquêtes par questionnaire en ligne, et si vous le souhaitez, vous pouvez intégrer des expériences dans ces questionnaires en ligne et ensuite collecter les données des réponses au questionnaire et les données expérimentales hors ligne.

Niveau	Temps de lecture	Thème
Facile Facile	15 minutes 5 minutes	Introduction au questionnaire en ligne. Lire ici Exemple complet . Lire ici
Facile + Avancé	Consultez les références	Tous les détails sur la façon d'écrire les questionnaires. Lire ici

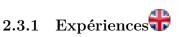
#### Expériences en ligne et hors ligne # 2.2



Faire des expériences dans le navigateur, c'est bien, mais vous pouvez aussi utiliser PsyToolkit sur les ordinateurs de bureau du laboratoire. C'est plutôt pour les utilisateurs avancés qui sont prêts à faire un pas de plus, bien que tout le monde puisse le faire. Vous devez installer Linux, et même cela est beaucoup plus simple que vous ne le pensez.

Cliquez ici pour en savoir plus sur l'installation et la configuration de PsyToolkit sur Linux

### Comment analyser les données #



Lorsque vous réalisez une expérience, un fichier de données est créé pour chaque participant. Le fichier de données est un simple fichier texte et peut être ouvert avec n'importe quel programme de statistiques ou de tableur. Plus d'informations ici.

#### Enquêtes # 2.3.2

Lorsque les participants remplissent des enquêtes en ligne, toutes leurs données sont stockées sur le serveur du PsyToolkit. Cela vous facilite la vie.

Vous pouvez télécharger les résultats de vos enquêtes en ligne à tout moment sous forme de fichier zip. Ce fichier zip contient un tableur appelé data.csv (en format CSV). Ce format est compatible avec la quasi totalité des logiciels de statistique : R, jamovi, JMP etc.. Vous pouvez également les télécharger au format Excel, qui est plus facile à importer dans SPSS. En outre, tous les fichiers de données brutes (en format texte) sont également disponibles, bien que vous n'en ayez généralement pas besoin.

- Le fichier data.csv (ou data.xlsx) contient une ligne de données de chaque participant. Chaque colonne de ce fichier représente une réponse.
- Le fichier data times.csv (ou data times.xlsx) contient le temps de réponse, c'est-à-dire letemps qu'il faut à un participant pour appuyer sur le bouton "continuer" pour chaque question (en millisecondes).

### Enquêtes avec expériences intégrées 🕏 2.3.3



Si votre enquête contient des expériences, les fichiers de données des expériences seront également sauvegardés. L'obtention des données est relativement simple. Vous pouvez obtenir les temps de réponse moyens et les taux d'erreur par condition pour chaque participant dans un fichier de tableur. Consultez la section "Analyser" des enquêtes pour plus d'informations. Vous pouvez également obtenir tous les fichiers de données brutes de chaque participant.

## Chapter 3

# Départ pour débutant absolu #

#### Ce que vous pouvez faire avec PsyToolkit # 3.1



- Programmez une expérience psychologique cognitive, telle que la tâche Simon, et exécutez-la dans votre navigateur (sans aucun plugin).
- Créer un questionnaire d'enquête en ligne.
- Exécutez des expériences et des enquêtes par questionnaire en ligne, et recueillez des données auprès des participants en ligne.
- Bien que le système basé sur le web soit pratique, vous pouvez à la place installer un ordinateur de laboratoire autonome en utilisant PsyToolkit (système d'exploitation Linux requis). Celui-ci prend en charge les claviers externes courants (Blackbox, Cedrus et autres matériels dédiés).

### Pour qui est fait PsyToolkit? #

- Étudiants en psychologie
- Chercheurs (psychologues, chercheurs en comportement)

PsyToolkit peut être utilisé dans le monde entier. Il peut être utilisé gratuitement (mais il ne peut pas être utilisé à des fins commerciales). Utilisez-le de manière responsable (voir les conditions juridiques).

### Combien de temps faut-il pour apprendre? 3.3

Vous pouvez l'apprendre même sans aucune connaissance en programmation. Apprendre quelque chose de nouveau prend du temps !!! Le temps d'apprentissage prévu pour les étudiants en psychologie (avec au moins une année d'études derrière eux) est, approximativement, le suivant :

- 1. Comprendre les bases du PsyToolkit : Au moins 1 heure (jusqu'à quelques heures).
- 2. Apprendre à mettre en place une étude par questionnaire : 1 à 5 heures.
- 3. Apprendre à mettre en place une expérience sur le temps de réaction : 2 à 10 heures.
- 4. Apprendre à analyser les données d'un questionnaire : 2 heures.
- 5. Apprendre à analyser les données d'une expérience : 5 heures.
- 6. Apprendre à dessiner des stimuli avec Inkscape : 20 minutes à 1 jour (selon le niveau de complexité).
- 7. Comprendre tout cela très bien : Une semaine

#### Étapes de l'apprentissage de PsyToolkit 🚏 3.4



Il existe différentes façons d'utiliser PsyToolkit:

- Utilisation de base : Il suffit de parcourir le site web pour découvrir les expériences et questionnaires psychologiques cognitifs existants. Pour cela, il est recommandé de lire les leçons en ligne et d'utiliser la bibliothèque d'enquêtes ou la bibliothèque d'expériences.
- Mener ses propres études : Vous pouvez utiliser PsyToolkit pour concevoir et mener vos propres études, et vous pouvez utiliser le serveur web de PsyToolkit pour collecter des données en ligne. Vous devrez apprendre le codage du PsyToolkit. N'ayez crainte, il y a beaucoup de documentation et de nombreux exemples complets à copier et à coller provenant des bibliothèques gratuites. Toutes les informations nécessaires pour apprendre PsyToolkit sont en ligne.

Le meilleur moyen d'apprendre le PsyToolkit est de regarder des exemples, de lire la documentation en ligne et de travailler en essayant et en faisant des erreurs jusqu'à ce que cela fonctionne (essai et erreur). Un exemple détaillé de A à Z avec des vidéos peut être consulté ici.

### 3.4.1 Étape 1

Assurez-vous de bien comprendre les bases des expériences psychologiques cognitives. Cliquez sur ce lien où ces concepts sont expliqués.

### 3.4.2 Étape 2

Les expériences ont besoin de stimuli. Vous avez besoin d'un programme de "dessin" pour créer de tels stimuli. Voici une leçon sur la façon de le faire avec Inkscape, qui est un outil gratuit

### 3.4.3 Étape 3

Vous êtes maintenant prêt à créer un compte PsyToolkit. Il vous suffit d'envoyer un courriel et de créer votre compte via la page principale (créez votre compte).

### 3.4.4 Étape 4

Passez en revue ces leçons détaillées sur la façon de mettre en place une enquête ou une expérience en ligne avec PsyToolkit.

### Chapter 4

# Programmer des expériences

### Les Scripts d'd'expérimentation dans Psy-Toolkit #

Les Scripts d'expérimentation est une partie importante de PsyToolkit (à ne pas confondre avec les scripts d'enquête). Le script d'expérimentation indique à l'ordinateur quand et comment montrer les stimuli, comment mesurer les réponses et comment écrire les données dans un fichier de données.

Le script d'expérimentation est comme la programmation d'un ordinateur. Vous ne pouvez l'apprendre qu'en faisant, en regardant des exemples et en étant patient. En fin de compte, il est beaucoup plus souple et plus facile à utiliser que les programmes de type "pointer-cliquer". Au début, il peut cependant sembler beaucoup plus difficile.

#### Définition d'un script 4.1.1



Un script est essentiellement un simple document textuel. Il contient toutes les informations nécessaires pour mener une expérience. Lorsque vous utilisez PsyToolkit dans le navigateur, le script s'affiche dans une zone de texte (voir image ci-dessous). Tout comme dans un éditeur de texte ordinaire, vous pouvez enregistrer le fichier. Si vous travaillez avec le navigateur, le fichier sera en fait enregistré sur le serveur PsyToolkit, et vous pourrez accéder au fichier partout dans le monde.

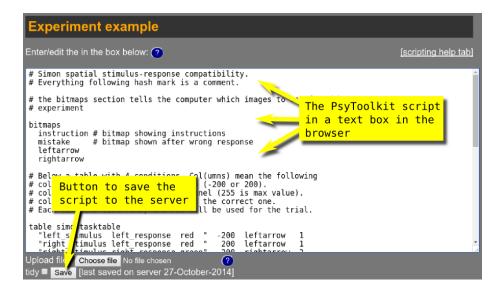


Figure 4.1: Un exemple de script

Si vous travaillez hors ligne sous Linux, vous pouvez modifier votre fichier de script dans un éditeur de texte, tel que gedit ou kate ou emacs.

### Étapes typiques d'une expérience



Vous passerez par un certain nombre d'étapes. PsyToolkit entre en jeu après l'étape 2 ci-dessous.

- 1. Concevez votre expérience sur papier.
- Pensez aux conditions expérimentales dont vous disposez
- Pensez aux instructions et aux stimuli que vous avez
- Écrivez les événements qui se produisent dans le cadre de votre expérience (quand et comment ils doivent se produire, ce que l'on attend des participants, etc.)
- 2. Créer les stimuli sous forme de bitmaps et/ou de fichiers audio
- Il est recommandé de créer les stimuli sous forme de fichiers images (par exemple, au format png, bmp, jpg). Le logiciel recommandé est le suivant : www.inkscape.org

- 3. Ecrivez le code PsyToolkit. Habituellement, chaque ligne PsyToolkit correspond à un événement des étapes précédentes. Le code peut être écrit dans un navigateur ou dans un éditeur de texte. Le code PsyToolkit est juste un fichier texte combiné avec les fichiers de stimulation.
- 4. Compilez le code PsyToolkit en une expérience exécutable. Si vous compilez sur un ordinateur Linux, vous pouvez exécuter le code sur un ordinateur local et même utiliser du matériel externe spécial, comme des claviers Cedrus. Si vous compilez dans un navigateur, vous pouvez exécuter le code dans un navigateur et même demander aux gens de participer en ligne.

## Les stimuli dans PsyToolkit #

Dans PsyToolkit, il y a des stimuli visuels et auditifs. Les stimuli visuels peuvent être divisés en images bitmap (chargées sous forme de fichiers bitmap dans n'importe quel format bitmap<sup>1</sup>), en rectangles et en texte. Les stimuli auditifs sont simplement des sons (chargés sous forme de fichiers sonores).

#### Comment créer des stimuli pour PsyToolkit 4.2.1



Il faut créer des stimuli. Par exemple, les stimuli visuels sont simplement des images bitmaps. Vous pouvez en trouver beaucoup sur internet (par exemple, avec Google images, mais assurez-vous que vous travaillez dans le respect de la législation applicable en matière de droits d'auteur). Vous pouvez également utiliser les images numériques de votre appareil photo (elles sont généralement beaucoup plus grandes que la résolution de votre écran, il vous faut donc un logiciel pour les redimensionner).

La taille d'écran par défaut est de 800 par 600 pixels. Vous pouvez la modifier avec l'option résolution Les images numériques peuvent être lourdes, surtout lorsqu'elles sont prises par un appareil photo numérique. Veillez à vérifier la taille en pixels. Un stimulus typique peut être de l'ordre de 100 par 100 pixels.

Le logiciel recommandé pour créer des stimuli de haute qualité est Inkscape. Ce logiciel est gratuit, peut être téléchargé, fonctionne sur tous les ordinateurs, est de haute qualité et comporte de nombreux tutoriels. Il faut peut-être une journée pour s'y habituer, mais c'est un temps d'apprentissage bien utilisé<sup>2</sup>.

Pour les stimuli auditifs, vous pouvez utiliser Audacity, qui est l'un des meilleurs et des plus populaires éditeurs de sons gratuits.

<sup>&</sup>lt;sup>1</sup>Une image bitmap est une image vectorielle. Sur l'ordinateur, elles sont enregistrées dans des fichiers image et portent l'extension/type png, jpg, bmp, gif, etc.

<sup>&</sup>lt;sup>2</sup>NdT. Pour ma part, j'ai une préférence, sous windows pour paint.net

également rechercher des fichiers sonores en ligne et les couper à la bonne durée, et les modifier en utilisant Audacity.

#### Comment charger les stimuli 4.2.2



Il existe des instructions spéciales pour indiquer à l'ordinateur de charger des stimuli. Il y a une instruction pour les images (bitmaps), les sons et les polices de caractères. Lorsque vous chargez des stimuli, vous pouvez spécifier le nom du stimulus que vous souhaitez utiliser et le nom du fichier. Voici un exemple de chargement de stimuli. Plus tard, vous pourrez vous référer à ces deux fichiers images avec "MyBitmap" et "anotherone".

#### Chargement des stimuli visuels, appelés bitmaps

```
bitmaps
 MyBitmap some_image.jpg
  anotherone bitmaps/anotherimage.png
```

Dans l'instruction bitmaps de l'exemple, vous avez deux choses. D'abord le nom de l'image tel que vous l'utiliserez dans le code suivant (par exemple, "MyBitmap"), suivi de son nom de fichier.

Si vous utilisez des bitmaps PNG, il vous suffit de donner le nom du PNG sans l'extension, et PsyToolkit reconnaîtra que pour le nom il y aura un nom de fichier sous ce nom avec l'extension PNG. Par conséquent, les fichiers PNG sont les plus pratiques à utiliser.

PNG est le format d'exportation des images par défaut d'Inkscape, un programme fantastique pour dessiner des stimuli.

#### Comment présenter les stimuli 4.2.3



Dans PsyToolkit, vous pouvez présenter des stimuli dans la session "tâche" en utilisant la commande "show".

### Afficher les bitmaps

```
task mytask
   show bitmap MyBitmap ## bitmap number 1
   show bitmap anotherone 100 300 ## bitmap number 2
   clear 1
   delay 1000
   clear 2
```

# 4.2.4 Système de coordonnées PsyToolkit 🖶

Dans PsyToolkit, un stimulus visuel est par défaut présenté exactement au centre de l'écran. Si vous souhaitez une position différente de 0,0, vous pouvez présenter les coordonnées x et y comme dans l'exemple ci-dessus (plus d'exemples ci-dessous).

Par défaut, la coordonnée 0,0 réfère au centre de l'écran. Si vous souhaitez que 0,0 soit en haut à gauche de l'écran, vous pouvez le spécifier dans une option, comme dans l'exemple ci-dessous.

#### Définir un autre système de coordonnées

```
options origin topleft
```

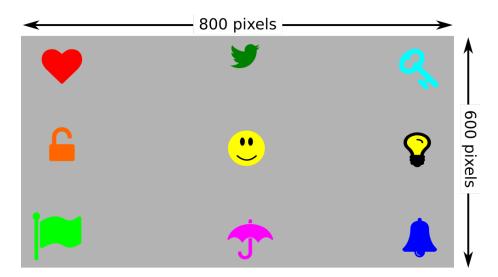


Figure 4.2: Exemple d'affichage de stimuli

#### Show the stimuli in a task

bitmaps
heart
bird
key
lock
smiley
bulb
flag

```
umbrella
  bell
task test
  show bitmap smiley
  show bitmap heart -300 -200
  show bitmap bird 0 -200
  show bitmap key 300 -200
  show bitmap lock -300 0
  show bitmap bulb 300 0
  show bitmap flag -300 200
  show bitmap umbrella 0 200
  show bitmap bell 300 200
  delay 1000
 block testblock
  tasklist
    test 10
  end
```

Ainsi, la coordonnée x,y 300 200 est présentée en bas à droite et la coordonnée x,y -300 -200 est présentée dans le quadrant supérieur gauche de l'écran. En d'autres termes, les coordonnées négatives réfèrent à la moitié supérieure de l'écran.

J'ai dessiné le smiley et pris les autres images dans le fantastique jeu d'icônes fontawesome. Téléchargez ici. Le jeu d'icônes fontawesome est livré avec des fichiers SVG, vous pouvez donc les importer dans Inkscape et les agrandir sans perte de qualité. Lisez une explication sur la mise à l'échelle des images ici.

#### Questions de temps 4.2.5



Un moniteur typique se "met à jour" 60 fois par seconde, c'est-à-dire 60 Hz, ou toutes les 17 millisecondes environ. Certains moniteurs spéciaux peuvent faire beaucoup mieux. Sur la version Linux/Desktop, vous pouvez contrôler le timing des stimuli en utilisant la fonction vsync (qui est activée par défaut et qui est typique des expériences psychologiques cognitives en laboratoire). Dans la version basée sur le web, vous ne pouvez pas faire cela, ce qui signifie que vous ajouterez du bruit à vos données expérimentales.

#### Syntaxe pour les feedback # 4.3



Le retour d'information permettra à un participant de donner son avis sur les performances, par exemple sur les temps de réponse moyens dans un bloc. Le retour d'information peut être programmé dans une section "bloc" uniquement (et non sur une base d'essai par essai).

La gestion des feedback n'est pas destiné aux débutants de PsyToolkit. Le feedback est une fonction avancée du codage des expériences PsyToolkit, et vous ne devez l'utiliser qu'une fois que vous avez maîtrisé la programmation des aspects plus basiques de l'expérience. Vous en trouverez de bons exemples dans les expériences téléchargeables dans la bibliothèque d'expériences.

#### Contexte $extstyle{\#}$ 4.3.1



Les chercheurs pourraient vouloir donner aux participants un retour d'information (feedback) sur leurs performances. Voici quelques raisons de donner un feedback :

- Le feedback peut motiver les participants à répondre plus précisément ou plus rapidement
- Les chercheurs peuvent vouloir utiliser les informations en retour lors de la conception de la tâche (mais ne pas les donner aux participants à la fin)

#### 4.3.2 Ce dont vous avez besoin pour pouvoir donner un feedback #

Afin d'utiliser un feedback pour les participants, vous devrez vous assurer que vous avez réfléchi aux points suivants :

- Dans quelle police de caractères souhaitez-vous présenter les informations de retour d'information? Les commentaires utiliseront la première police spécifiée.
- Sur quelles données voulez-vous donner un feedback? Vous devez vous assurer que vous avez enregistré ces données en utilisant l'instruction "save" dans la description de la tâche.

Quel type de retour d'information souhaitez-vous donner? Le plus souvent, il s'agit simplement d'indiquer à l'utilisateur le nombre d'erreurs qu'il a commises ou la vitesse moyenne de réponse.

#### Syntaxe # 4.3.3



Les instructions de retour d'information sont données dans la déclaration "bloc". Dans les exemples ci-dessous, la partie "feedback" des blocs sera cependant indiquée. Vous trouverez un exemple complet en bas de cette page. Vous trouverez des exemples complets dans la bibliothèque d'expériences. Il y a actuellement 5 composants principaux pour donner un feedback. Il est probablement bon de regarder d'abord l'exemple ci-dessous pour avoir une impression rapide (le codage est souvent appris plus rapidement en regardant les exemples).

- set
- text
- lineplot
- xyplot
- save

Elles sont décrites plus en détail ci-dessous.

## 4.3.4 text #

Le texte vous permet de présenter des valeurs variables (la commande "set" vous permet de calculer les moyennes, les min, max et les pourcentages des réponses des participants), et plus encore (chaque fonction décrite ci-dessous).

Les guillement autour d'un texte littéral dans un feedback ne sont pas obligatoires , bien qu'elles soient recommandées. Dans les commandes de la tâche "show text", les guillemets doivent cependant être utilisés.

#### Exemple de texte

```
feedback
  text 0 -150 "Press space bar to continue"
end
```

#### Exemple de texte

```
feedback
  set &MyVar perc 10 ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Percentage errors (should really be below 10%):" ; postend
```

Vous devez renvoyer à la colonne du fichier de données (dans l'exemple où nous sélectionnons la 4ème colonne, le préfixe c indique la colonne. Il s'agit de la colonne du fichier de données et non du tableau!

Le mot-clé select ne fonctionne qu'avec des chiffres, pas avec des noms. Ainsi, si vous voulez sélectionner uniquement les essais d'un certain bloc, ne vérifiez pas cela avec son nom mais avec son numéro. Vous pouvez utiliser BLOCKNUMBER dans la ligne **save** Notez également qu'un message autonome augmente le compteur BLOCKNUMBER.

# 4.3.4.1 attributs de texte $\oplus$



Vous pouvez aligner le texte à gauche si nécessaire. Vous pouvez utiliser les mêmes attributs de texte dans les expériences. Par exemple :

#### Exemple de texte

```
feedback
text color yellow
text align left
text 0 0 "Line 1"
 text 0 100 "Line 2 with some more text"
end
```

Un autre exemple de texte

```
feedback
text align center
 text color 00FF00
 text 0 0 "Line 1"
 text 0 100 "Line 2 with some more text"
end
```

#### définir des variables 4.3.5



Vous pouvez définir des variables globales (utilisez le signe &). Il s'agit très probablement de variables que vous n'avez pas utilisées dans les tâches ou les blocs, bien que vous puissiez les montrer ici aussi!

Il existe un certain nombre de fonctions que vous pouvez utiliser, chacune étant expliquée par un exemple ci-dessous. La partie "select" d'une ligne "set" dans la section "feedback" vous permet de n'utiliser que les essais de certaines lignes (voir ci-dessous pour plus d'explications)

## 4.3.5.1 mean #

Calculez la moyenne

#### Exemple avec mean

```
feedback
 set &MyVar mean c6 ; select c4 == 1
 text 0 -150 &MyVar; prefix "Mean response time:"; postfix "ms."
end
```

Dans l'exemple ci-dessus, imaginez que la colonne 6 des données contient le temps de réaction et la colonne 4 le statut (la valeur 1 étant correcte, comme c'est le cas par défaut dans PsyToolkit). Dans ce cas, &MyVar sera réglé sur le temps de réaction moyen, mais uniquement pour les essais dans lesquels aucune erreur n'a été commise (car seuls les essais pour lesquels la quatrième colonne est 1 sont pris en compte).

### 4.3.5.2 min #

Calculez la plus petite valeur

#### Exemple avec min

```
feedback
  set &MyVar min c6 ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Shortest response time:" ; postfix "ms."
end
```

Dans l'exemple ci-dessus, imaginez que la colonne 6 des données contient le temps de réaction et la colonne 4 le statut (la valeur 1 étant correcte, comme c'est le cas par défaut dans PsyToolkit). Dans ce cas, &MyVar sera réglé sur le temps de réaction minimum (c'est-à-dire le plus court), mais uniquement pour les essais dans lesquels aucune erreur n'a été commise (car seuls les essais pour lesquels la quatrième colonne est 1 sont pris en compte).

## 4.3.5.3 Max #

Calucler la plus grande valeur

#### Exemple de maximum

```
feedback
  set &MyVar max c6 ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Longest response time:" ; postfix "ms."
end
```

Dans l'exemple ci-dessus, imaginez que la colonne 6 des données contient le temps de réaction et la colonne 4 le statut (la valeur 1 étant correcte, comme c'est le cas par défaut dans PsyToolkit). Dans ce cas, &MyVar sera réglé sur le temps de réaction maximum (c'est-à-dire le plus long), mais uniquement pour les essais dans lesquels aucune erreur n'a été commise (car seuls les essais pour lesquels la quatrième colonne est 1 sont pris en compte)

# 4.3.5.4 sum

Calculer la somme des valeurs

#### exemple avec sum

```
feedback
 set &MyVar sum c6 ; select c4 == 1
 text 0 -150 &MyVar; prefix "Total response time:"; postfix "ms."
end
```

Dans l'exemple ci-dessus, imaginez que la colonne 6 des données contient le temps de réaction et la colonne 4 le statut (la valeur 1 étant correcte, comme c'est le cas par défaut dans PsyToolkit). Dans ce cas, &MyVar sera fixé à la somme totale de tous les temps de réaction (pour tous les essais), mais uniquement pour les essais dans lesquels aucune erreur n'a été commise (car seuls les essais pour lesquels la quatrième colonne est 1 sont pris en compte). Bien sûr, cette fonction est rarement utilisée, mais la somme peut être une fonction pratique pour les calculs ultérieurs.

## 4.3.5.5 count #



Calculer le nombre d'essais

#### Exemple avec count

```
feedback
 set &MyVar count ; select c4 != 1
 text 0 -150 &MyVar; prefix "Number of errors:"
end
```

Dans l'exemple ci-dessus, imaginez que la colonne 4 contient le statut (avec la valeur 1 étant correcte, comme c'est le cas par défaut dans PsyToolkit). Dans ce cas, &MyVar sera réglé sur le nombre d'erreurs (seuls les essais pour lesquels la quatrième colonne n'est pas correcte sont comptés).

## 4.3.5.6 perc

Il calcule le pourcentage (c'est-à-dire une valeur de 0 à 100) de l'occurrence. Cela n'a de sens qu'en combinaison avec l'option de sélection dans la même ligne (voir ci-dessous). Par défaut, le pourcentage est pris de tous les essais dans l'ensemble de données jusqu'à présent, sauf si un nombre spécifique est donné (il doit s'agir d'une valeur entière, il ne peut s'agir d'une variable).

L'exemple ci-dessous calcule le pourcentage d'essais pour lesquels la valeur de la colonne 4 est égale à 1. Imaginez qu'il y ait un total de 100 essais, et que dans 90 d'entre eux, la quatrième colonne contienne un 1, et dans 10 d'entre eux un 2. Dans ce cas, MyVar contiendrait 90, sur la base de 90/100\*100.

#### Exemple 1 avec perc

```
feedback
  set &MyVar perc ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Percentage errors:" ; postfix "%"
end
```

L'exemple ci-dessous calcule le pour centage de procès pour lesquels la valeur de la colonne 4 est égale à 1. I maginez qu'il y ait un total de 100 procès, et que dans 90 d'entre eux, la quatrième colonne contienne un 1, et dans 10 d'entre eux un 2. Dans l'exemple, le nombre total de procès est ignoré, et la valeur 200 est utilisée à la place dans le calcul du pour centage. Ainsi, dans ce cas, MyVar contiendrait 45, sur la base de 90/200\*100. Ce la est souvent utile lorsque vous souhaitez calculer un pour centage pour un sous-ensemble d'essais dont vous connaissez le nombre d'essais.

#### Exemple 2 avec perc

```
feedback
  set &MyVar perc 200 ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Percentage errors:" ; postfix "%"
end
```

## 4.3.5.7 régler une variable à partie de la valeur d'une autre variable $\blacksquare$

Exemple 1 de paramétrage d'une variable à partir de la valeur d'une autre variable

```
feedback
  set &current_variable &old_variable
end
```

ou vous pouvez le faire après le feedback :

Exemple 2 de paramétrage d'une variable à partir de la valeur d'une autre variable

```
feedback
  ...code...
end
set &current_variable &old_variable
```

## 4.3.5.8 Montrer la valeur calculée au sujet



#### Exemple de paramétrage et de texte

```
feedback
  set &StroopCompatible mean c5 ; select c6 == 1
  set &StroopIncompatible mean c5 ; select c6 == 2
  \verb§set &MyStroopEffect expression &StroopIncompatible - &StroopCompatible | \\
  text 0 -150 &MyStroopEffect; prefix "Your Stroop compatibility effect:"; postfix " ms"
end
```

# 4.3.5.9 paramétrage en bref



En bref, il y a deux types de paramétrage (set) pour le feedback :

- 1) Le paramétrage avec une expression pour calculer les nouvelles valeurs à afficher. C'est pratique pour montrer la différence entre les conditions comme dans l'exemple ci-dessus.
- 2) Le paramétrage avec une variable globale en calculant la moyenne, le minimum, le maximum ou le pourcentage de la valeur d'une colonne dans votre fichier de données. Imaginez que la 5ème colonne de votre fichier de données contient les temps de réponse, alors vous pouvez faire ce qui suit

#### Exemple de paramétrage

```
feedback
  set &MyAverage mean c5
  set &MyAverage min c5
  set &MyAverage max c5
  set &MyAverage perc ; select c6 == 1
  set &MyAverage perc 100 ; select c6 == 1
  set &MyAverage &some_other_variable
  set &MyAverage expression &MyAverage * 100
  set &MyAverage expression 100
end
```

Imaginez maintenant que la colonne 6 contienne le statut, 1 étant correct, et tout ce qui est incorrect. Imaginez également qu'il y ait 200 essais au total. Vous pouvez maintenant rechercher le pourcentage d'essais corrects, et vous pouvez aussi calculer pour les essais corrects seulement :

#### Exemple de paramétrage

```
feedback
  set &MyAverage mean c5 ; select c6 == 1
  set &MySlowest min c5 ; select c6 == 1
  set &MyFastest max c5 ; select c6 == 1
  set &MyErrorRate perc ; select c6 != 1
  text 0 -50 "Some feedback about your performance:"
  text 0 0 &MyAverage ; prefix "Average response time" ; postfix "ms"
  text 0 50 &MySlowest ; prefix "Slowest response time" ; postfix "ms"
  text 0 100 &MySlowest ; prefix "Slowest response time" ; postfix "ms"
  text 0 150 &MyErrorRate ; prefix "Error percentage" ; postfix "ms"
end
```

La fonction "perc" peut prendre un nombre supplémentaire, ou vous pouvez ne pas en tenir compte. Si vous donnez le nombre supplémentaire, il sera utilisé comme total à la place le nombre total d'essais. Cette option est utile dans certaines situations.

### 4.3.5.10 Select #

L'option select d'une ligne de paramétrage vous permet de ne sélectionner que les lignes de votre fichier de données qui vous intéressent. Cette option de sélection utilise sa propre syntaxe. Chaque instruction select commence évidemment par "select", suivi d'un certain nombre de comparaisons. Vous pouvez utiliser && pour le "et" logique || pour le "ou" logique.

Vous pouvez demander si les valeurs d'une colonne spécifique dans votre fichier de données (telle que créée par vos instructions "save") ont une valeur numérique spécifique, ou la valeur d'une variable globale, ou le BLOCKNUMBER.

Si vous voulez seulement sélectionner les essais du dernier bloc, vous pouvez le faire comme suit. Tout d'abord, déterminez quelle ligne de votre fichier de données représente le BLOCKNUMBER.

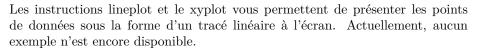
Imaginez que la 5ème colonne représente le temps de réponse (TR), la 6ème colonne le STATUT et la première colonne le numéro d'un bloc d'essai (BLOCK-NUMBER), alors Vous pouvez facilement utiliser ce qui suit pour ne prendre que les moyennes du bloc sur lequel vous donnez votre feedback :

Exemple d'ensemble avec des valeurs uniquement sur le dernier bloc

#### feedback

```
set &MyAverage mean c5; select c6 == 1 && c1 == BLOCKNUMBER
set &MySlowest min c5; select c6 == 1 && c1 == BLOCKNUMBER
set &MyFastest max c5; select c6 == 1 && c1 == BLOCKNUMBER
set &MyErrorRate perc; select c6 != 1 && c1 == BLOCKNUMBER
text 0 -50 "Some feedback about your performance:"
text 0 0 &MyAverage; prefix "Average response time"; postfix "ms"
text 0 50 &MySlowest; prefix "Slowest response time"; postfix "ms"
text 0 100 &MySlowest; prefix "Slowest response time"; postfix "ms"
text 0 150 &MyErrorRate; prefix "Error percentage"; postfix "ms"
end
```

## 4.3.6 Lineplot et xyplot 🟶



Le point-virgule pour séparer les sections des commandes de feeback doit être entouré d'espaces (des deux côtés). Sans cela, il sera ignoré.

## 4.3.7 wait\_for\_key

Par défaut, le feedback reste à l'écran jusqu'à ce que la barre d'espacement soit enfoncée. Il n'est pas nécessaire d'inclure une "wait\_for\_key" dans la section "feedback". Mais si vous souhaitez qu'une touche différente de l'espace soit utilisée, vous pouvez le spécifier avec "wait for key".

Cela peut être utile si vous utilisez un clavier externe, comme le clavier de la BlackBox.

#### Exemple de jeu de valeurs uniquement sur le dernier bloc

#### feedback

```
set &MyAverage mean c5 ; select c6 == 1 && c1 == BLOCKNUMBER
set &MySlowest min c5 ; select c6 == 1 && c1 == BLOCKNUMBER
set &MyFastest max c5 ; select c6 == 1 && c1 == BLOCKNUMBER
set &MyErrorRate perc ; select c6 != 1 && c1 == BLOCKNUMBER
text 0 -50 "Some feedback about your performance:"
text 0 0 &MyAverage ; prefix "Average response time" ; postfix "ms"
text 0 50 &MySlowest ; prefix "Slowest response time" ; postfix "ms"
text 0 100 &MySlowest ; prefix "Slowest response time" ; postfix "ms"
text 0 150 &MyErrorRate ; prefix "Error percentage" ; postfix "ms"
wait_for_key d
end
```

## 4.3.8 save #

As in a task, you can use the save statement to save global variables. If you use this, it is highly recommended not to use save in tasks, because it may mess up your data output.

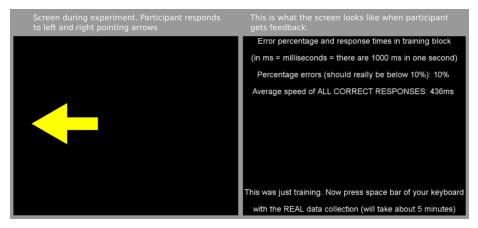
#### Exemple de paramétrage avec value et save

```
feedback
  set &MyAverage mean c5; select c6 == 1 && c1 == BLOCKNUMBER
  set &MySlowest min c5 ; select c6 == 1 && c1 == BLOCKNUMBER
  set &MyFastest max c5 ; select c6 == 1 && c1 == BLOCKNUMBER
  set &MyErrorRate perc ; select c6 != 1 && c1 == BLOCKNUMBER
 text 0 -50 "Some feedback about your performance:"
  text 0 0 &MyAverage; prefix "Average response time"; postfix "ms"
 text 0 50 &MySlowest; prefix "Slowest response time"; postfix "ms"
 text 0 100 &MySlowest; prefix "Slowest response time"; postfix "ms"
 text 0 150 &MyErrorRate; prefix "Error percentage"; postfix "ms"
  save BLOCKNUMBER &MyAverage &MySlowest &MyFastest &MyErrorRate
end
```

#### Exemple complet de tâche avec feedback 4.3.9



Dans cet exemple, les participants répondent avec la touche "a" ou "l" à une flèche pointant vers la gauche ou vers la droite (un essai avec une flèche pointant vers la gauche dans l'image ci-dessous). Le retour d'information indique au participant la vitesse moyenne à laquelle il s'est trouvé dans les essais corrects, et le pourcentage d'erreur (exemple à droite dans l'image ci-dessous).



Les scripts de feed-back paraissent relativement mystérieux au début, mais ils sont très simples.

Exemple de la façon d'utiliser l'instruction "feedback" dans un script PsyToolkit

```
# the following is an example of a Simon task with feedback about
# error rates and response times
bitmaps
 leftarrow
 rightarrow
 fixpoint
fonts
 arial arial.ttf 18
table simontasktable
  "L_pos L_arrow" -200 1 leftarrow
  "R_pos R_arrow" 200 2 rightarrow 2
  "R_pos L_arrow" 200 3 leftarrow
  "L_pos R_arrow" -200 4 rightarrow 2
task simon
 table simontasktable
 keys a 1
 show bitmap fixpoint
 delay 100
 show bitmap @4 @2 0
 readkey @5 2000
 delay 5000
 save BLOCKNAME @1 @3 TABLEROW KEY STATUS RT
block mytrainingblock
 tasklist
   simon 20
  end
 feedback
   set &MyPerc perc ; select c7 != 1
   set &MyMean mean c8 ; select c7 == 1
   set &MyCount count ; select c7 == 1
   text 0 -250 "Error percentage and response times in training block"
   text 0 -200 "(in ms = milliseconds = there are 1000 ms in one second)"
   text 0 -150 &MyPerc ; prefix "Percentage errors (should really be below 10%):"; postfix "%'
   text 0 -100 &MyMean ; prefix "Average speed of ALL CORRECT RESPONSES:"; postfix "ms"
   text 0 -50 &MyCount; prefix "Total number of correct trials:"
   text 0 200 "This was just training. Now press space bar of your keyboard"
   text 0 250 "with the REAL data collection (will take about 5 minutes)"
  end
```

### 4.4 Partager une expérience 🟶

Imaginez que vous avez créé une belle expérience. Et qu'ensuite, vous aimeriez la donner à votre professeur, à un collègue, à un étudiant, etc. Il y a deux façons différentes de le faire :

Télécharger l'expérience : Toutes les expériences peuvent être téléchargées sur votre disque dur local, puis téléchargées plus tard sur un autre compte. Ainsi, vous pouvez télécharger et envoyer le fichier d'expérience téléchargé à votre ami.

Partagez l'expérience : Au lieu de télécharger et d'envoyer par e-mail, il existe un système interne qui permet d'envoyer une expérience à un autre utilisateur. Il suffit de préparer votre expérience, et lorsque vous êtes prêt, cliquez sur le bouton de partage *share* dans le menu de gauche. Cela permet d'envoyer l'état actuel de l'expérience à votre ami. L'autre utilisateur n'a plus accès à votre expérience et ne verra pas les modifications que vous apportez. Il s'agit simplement d'une façon interne d'envoyer une expérience.

# 4.5 Collecter des données expérimentales en ligne $\clubsuit$

Imaginez que vous avez créé une belle expérience, et que vous voulez collecter des données en ligne. Vous pouvez le faire, et c'est l'une des particularités attrayantes PsyToolkit. Voici comment cela fonctionne :

- Vous créez une expérience
- ou bien Vous créez une enquête et "intégrez" l'expérience
- Vous faites compiler l'enquête et la rendez disponible en ligne (il y a une option pour cela dans les enquêtes).

Vous obtiendrez un lien à partir de PsyToolkit, et vous pourrez l'envoyer à vos étudiants, collègues, ou à qui vous voulez. Veillez à respecter les directives éthiques de votre établissement.

Une fois que les personnes ont participé, vous pouvez télécharger les fichiers de données et les analyser sur votre propre ordinateur.

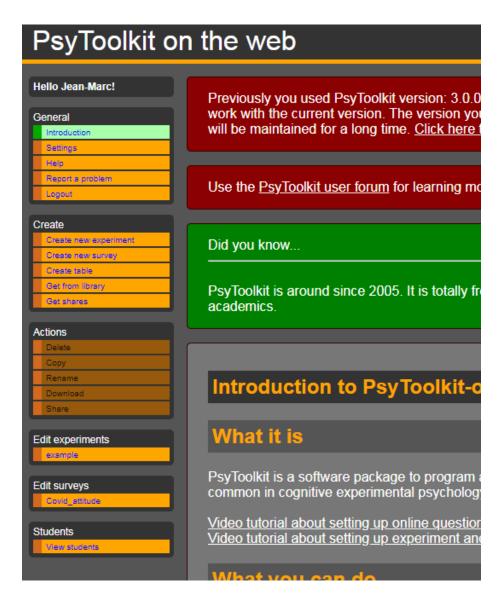


Figure 4.3: Partager une expérience

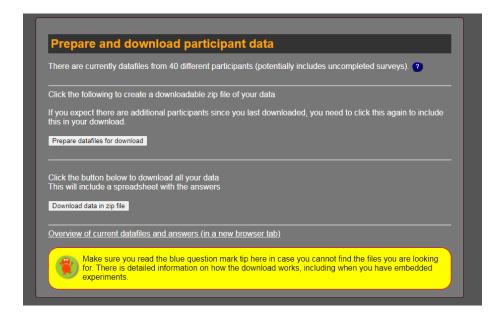


Figure 4.4: Télécharger les données

### Chapter 5

# Syntaxe pour les scripts #

Cette page est simplement destinée à rechercher les instructions des scripts du PsyToolkit. Elle suppose que vous savez plus ou moins ce que vous recherchez. Si vous êtes nouveau à PsyToolkit, il est plus facile de commencer à regarder certains des exemples ici ou des exemples plus détaillés dans la bibliothèque d'expériences.

Lien rapide vers toutes les instructions

Autres documents pertinents (en anglais):

- Informations générales sur PsyToolkit
- Informations générales sur les scripts
- Aide détaillée sur les scripts et la version hors ligne

# 5.1 Comment utiliser cette page de présentation des scripts d'expériences #

Cette page énumère tous les éléments possibles d'un script PsyToolkit. Elle donne également une brève description syntaxique. Une syntaxe est une description de la manière dont les instructions peuvent être utilisées. Dans la description syntaxique, les arguments obligatoires d'une instruction sont entre crochets []. Les arguments optionnels entre crochets normaux (). Un signe | indique différentes valeurs valides pour un argument.

Par exemple, prenons l'exemple de l'instruction "show rectangle". L'exemple ci-dessous montre qu'il y a 4 arguments obligatoires : X,Y,W,H. Il s'agit de la coordonnée x, de la coordonnée y, de la largeur et de la hauteur. Les arguments Rouge, Vert et Bleu sont optionnels (par défaut, un rectangle est blanc, mais vous pouvez lui donner des couleurs différentes)

• show rectangle X,Y,W,H

Les parenthèses dans la description de la syntaxe ne sont bien sûr pas tapées dans le vrai script ; c'est juste une façon de décrire une syntaxe. Regardez attentivement les exemples donnés pour chaque instruction pour voir comment utiliser les instructions.

#### Structure des scripts # 5.2



Il existe différentes sections dans les scripts, et chaque section est séparée des autres par au moins une ligne vide. Tout ce qui suit un hashtag (#) est un commentaire uniquement pour les yeux humain, l'ordinateur l'ignorera. Les principales sections qui contiennent chacune plusieurs lignes sont les suivantes:

- options de base décrit des paramètres pratiques du PsyToolkit qui ne sont pas par défaut, tels que la taille de l'écran
- options avancées décrit les options avancées que les débutants peuvent ignorer
- options linux uniquement options avancées spéciales lors de l'utilisation de la version Linux
- bitmaps/sons/fonts décrit les stimuli à utiliser
- La tâche peut être utilisée plusieurs fois pour décrire les essais de paradigmes
- Le tableau peut être utilisé plusieurs fois pour décrire les conditions expérimentales
- le bloc peut être utilisé plusieurs fois pour exécuter des blocs d'essais
- l'ordre des blocs peut être fixé de manière différente (utilisateurs avancés uniquement)

### Option de base # 5.3



Chaque fois que vous programmez une expérience, vous pouvez vouloir vous écarter des valeurs par défaut. Par exemple, la taille d'écran par défaut d'une expérience est de 800 par 600 pixels, mais vous pouvez la modifier en utilisant l'option "resolution". Les options disponibles sont énumérées ci-dessous. Au bas de cette section se trouve un exemple complet.

Il existe de nombreuses options, mais elles sont surtout destinées aux utilisateurs avancés. Si vous êtes nouvel utilisateur de PsyToolkit, ignorez dans un premier temps les options.

# 5.3.1 resolution $\bigoplus$

Cela détermine la résolution de l'expérience. Cette résolution par défaut est de 800 par 600. Lorsque vous spécifiez la résolution, il s'agit de la largeur par la hauteur, séparées seulement par un espace. Voir l'exemple ci-dessous.

Exemple de plein écran (avec un rectangle de saut aléatoire)

```
options
fullscreen
resolution 1000 800

task MyTask
set $x random -300 300 # random x position
set $y random -300 300 # random y position
set $mysize random 10 100
show rectangle $x $y $mysize $mysize 255 0 0 # show red rectangle
delay 100
clear -1

block MyBlock
tasklist
MyTask 200
end
```

# 5.3.2 frame $\bigoplus$

Ceci définit la taille du cadre de l'expérience dans une expérience en ligne lorsqu'elle n'est pas en mode plein écran. Cette fonction est rarement nécessaire, mais il est bon de savoir que vous pouvez l'ajuster.

Dans l'exemple ci-dessous, vous travaillez avec une résolution de 1000 800. Mais sans l'option *frame*, elle serait réduite à 800 par 600 en mode non plein écran. Si vous voulez que dans le mode non plein écran, la boîte apparaisse plus grande, vous pouvez ajuster cela avec *frame*.

### Exemple de frame

```
options
fullscreen
resolution 1000 800
frame 1000 800
```

Frame n'est peut-être pas souvent nécessaire. Lorsque vous l'utiliserez, il aura probablement les mêmes valeurs que le plein écran.

## 5.3.3 scale $\clubsuit$

L'option d'échelle scale vous permet de mettre l'écran à l'échelle. Ceci est pratique en mode plein écran ou pour des résolutions plus grandes que la zone par défaut de 800 par 600 pixels. PsyToolkit conservera toujours le rapport d'aspect correct.

Quand cela prend-il effet ?

- Lorsque vous passez en mode plein écran (la taille augmente).
- Lorsque vous avez une résolution supérieure à 800 x 600 et que vous n'êtes pas en mode plein écran (l'échelle sera réduite).

### Exemple de scale

```
options
 fullscreen
 scale
 resolution 1600 1200
```

## 5.3.4 background color #



Vous pouvez spécifier la couleur de fond de l'écran. Par défaut, il est noir, mais vous pouvez le régler ici sur une couleur différente. Vous pouvez la spécifier en utilisant un nom de couleur (rouge, vert, bleu, blanc, jaune, rose, violet, noir, orange) ou un triplet RVB (255 0 0 pour le rouge), ou une combinaison hexadécimale de six lettres/chiffres (FF0000 pour le rouge).

#### Exemple de fond

```
options
  background color white
task MyTask
  set $x random -300 300 # random x position
  set $y random -300 300 # random y position
  set $mysize random 10 100
  show rectangle $x $y $mysize $mysize
                                         255 255 0 # show yellow rectangle
  delay 100
  clear -1
block MyBlock
  tasklist
   MyTask 200
```

end

Un autre exemple utilisant un triplet RGB (dans ce cas, blanc)

```
options
  background color 255 255 255
```

Un autre exemple utilisant un hexa code

```
options
  background color FFFF00
```

### fullscreen # 5.3.5



Exécute l'expérience en mode plein écran. Sous Linux, c'est le mode par défaut, alors que dans le navigateur, ce n'est pas le cas.

Dans la version en ligne (Javascript), le mode plein écran ne fonctionne pas dans le navigateur Safari d'Apple (Apple n'autorise pas le mode plein écran et la saisie au clavier).

Exemple de plein écran (avec un rectangle de saut aléatoire)

```
task MyTask
  set $x random -300 300 # random x position
  set $y random -300 300 # random y position
  set $mysize random 10 100
  show rectangle $x $y $mysize $mysize
                                         255 255 0 # show yellow rectangle
  delay 100
  clear -1
block MyBlock
  tasklist
   MyTask 200
  end
```

### $\text{mouse} \; {}^{\bigoplus}_{}$ 5.3.6

Par défaut, le curseur de la souris n'est pas affiché, car d'un point de vue psychologique, il peut être un stimulus distrayant. Mais pour certaines expériences, la souris est nécessaire en tant que dispositif de pointage. Cette option permet de s'assurer que le curseur de la souris est visible. La seule valeur utile est "on".

Exemple d'affichage de la souris dans une expérience

```
options
 mouse on
```

L'option ne met la souris en marche qu'au début de l'expérience. Vous pouvez toujours l'éteindre plus tard dans une tâche en utilisant les instructions de masquage et d'affichage de la souris. Cliquez ici pour en savoir plus sur ces deux instructions.

## 5.3.7 set #

Cela vous permet de définir une variable globale. Cela peut être pratique si vous souhaitez tester un paramètre expérimental (par exemple, l'intervalle intertrial) à une valeur spécifique tout au long de l'expérience. Si vous n'utilisez la variable nulle part, cette valeur sera simplement ignorée.

### Exemple de "set" dans les options

```
options
  fullscreen
  set &my_intertrialinterval 100
task MyTask
  set $x random -300 300 # random x position
  set $y random -300 300 # random y position
  set $mysize random 10 100
  show rectangle $x $y $mysize $mysize
                                          255 0 0 # show red rectangle
  delay 100
  clear -1
  delay &my_intertrialinterval
block MyBlock
  tasklist
    MyTask 200
  end
```

Il est parfois pratique d'avoir défini certains paramètres de base dans les options en tant que variables globales. De cette façon, vous pouvez facilement les retrouver. De bons exemples sont les intervalles entre les essais, les temps de réponse maximums autorisés, la taille du stimulus, etc.

### start messages 5.3.8



Dans la version en ligne, chaque expérience commence par une case rouge et le message en anglais "Click to start". En outre, selon le temps nécessaire pour

charger les messages, vous verrez un message attendre une seconde. Vous pouvez modifier ces deux messages avec les deux options suivantes :

### Exemple de modification des messages de démarrage

```
options
startbutton text My own click-to-start message
loading text My own wait-a-second message
```

### 5.4 Options avancées \$\Psi\$

## 5.4.1 var in $\bigoplus$

Voir l'échange d'information entre les questionnnaires et les expériences

Lorsque vous utilisez PsyToolkit dans le cadre d'enquêtes en ligne, vous pouvez échanger des variables numériques entre les expériences et une enquête.

Vous ne pouvez changer que des variables numériques en nombre entier. C'està-dire des nombres tels que 12 ou 3, mais pas des lettres, des combinaisons de chiffres et de lettres, ni des nombres fractionnaires (tels que 12.1 ou 3.4)

Par exemple, vous pouvez poser une question d'enquête demandant l'âge d'une personne, puis utiliser l'âge comme variable dans l'expérience. Vous devez préciser le nom de l'enquête. Vous trouverez ci-dessous un exemple.

Exemple d'une question d'enquête en premier lieu. Note : il ne s'agit donc pas d'un code de script d'expérimentation !

```
1: sleep
t: range
q: how many hours sleep did you have?
- {min=0,max=10} Select number

1: alcohol
t: range
q: how many glasses of alcohol did you dring?
- {min=0,max=20} Select number

1: the_experiment
t: experiment
t: experiment
- my_experiment

1: information
t: info
q: Your score is {$xxx} and {$yyy}.
```

Dans l'expérience suivante, les participants sont invités à répondre à un rectangle. Supposons que les personnes endormies ou celles qui consomment de l'alcool ont besoin de plus de temps, nous voulons leur donner plus de temps, afin de leur donner le maximum de temps pour répondre plus longtemps. Nous créons une nouvelle variable MyMaxTime basée sur leur consommation d'alcool et leurs heures de sommeil. Plus ils ont bu ou moins ils ont dormi, plus &My-MaxTime sera élevé. Ceci montre simplement comment vous pouvez utiliser les variables

Notez que les variables sommeil et alcool sont traitées comme des variables globales, et doivent être précédées du signe &.

Exemple d'utilisation d'une réponse à une enquête dans le cadre d'une expérience

```
options
  var in sleep alcohol

task myTask
  keys space
  show rectangle 0 0 100 100
  set &MyMaxTime expression 1000 + 100 * &alcohol + 100 * ( 10 - &sleep )
  readkey 1 &MyMaxTime
  save RT

block
  tasklist
   myTask 10
  end
```

# 5.4.2 var out **\$\frac{1}{4}**

Voir l'échange d'information entre les questionnnaires et les expériences

L'option var out peut être utilisée pour sauvegarder une valeur créée dans une expérience directement dans l'enquête. Dans ce cas, la valeur sauvegardée sera disponible dans le feuille de calcul du questionnaire.

Exemple d'utilisation d'une réponse à une enquête dans le cadre d'une expérience

```
options
  var out my_score
```

Exemple de question d'enquête fonctionnant avec cela. Notez que ce script est destiné aux enquêtes en ligne uniquement.

```
1: my_feedback
```

t: info

q: Your score was {\$my\_score}

#### origin 贵 5.4.3

Les stimuli visuels sont présentés à l'écran, et vous devez donner une coordonnée X et une coordonnée Y pour les placer. L'option d'origine indique à l'ordinateur où se trouve le point X=0,Y=0 (ou 0,0).

Par défaut, le point de coordonnées 0,0 est au centre de l'écran. Vous pouvez également définir le coin supérieur gauche de l'écran comme origine (ce qui était la valeur par défaut dans les anciennes versions de PsyToolkit).

### Exemple d'utilisation de l'instruction origin

```
options
  origin topleft
```

# fontdir,<br/>datadir,<br/>bitmapdir,<br/>sounddir $\P$



Les options fontdir, datadir, bitmapdir, sounddir vous permettent de modifier les dossiers dans lesquels l'ordinateur s'attend à ce que les stimuli soient enregistrés. Vous n'avez pas besoin de le préciser si les fichiers de stimuli se trouvent dans le même répertoire que votre code de script.

Cette option est particulièrement utile si vous travaillez en C ou si vous faites de grandes expériences.

#### Options pour Linux # 5.5



### escape (C seulement) 5.5.1

Dans la version Linux Desktop, cela permet de terminer le programme en appuyant sur la touche d'échappement. Veuillez noter que l'ordinateur vérifie uniquement si la touche d'échappement a été enfoncée à la fin de chaque essai. Par conséquent, si vous utilisez cette option, et si vous souhaitez interrompre votre expérience, vous devez maintenir le bouton d'échappement (ou tout autre bouton utilisé pour l'échappement) pendant la durée d'au moins un essai.

Dans le navigateur, vous pouvez toujours vous échapper de votre expérience, et là, cette option n'est pas nécessaire.

## 5.5.2 parallelport (C seulement)

Il s'agit d'un codage avancé du port parallèle.

### 5.5.3 pcidio24 (C seulement)

This is for advanced coding of the pcidio24 port

# 5.5.4 cedrus (C seulement)

Cela indique que vous avez attaché un clavier Cedrus. En option, vous pouvez spécifier un modèle. Si vous spécifiez un modèle, le script ne fonctionnera qu'avec ce modèle. Cette rigueur n'est nécessaire que dans le cas où vous voulez forcer les gens à utiliser un clavier Cedrus spécifique.

## 5.5.5 iolab (C seulement)

Il semble que cet appareil ne soit plus produit. PsyToolkit le supporte toujours.

Cela indique que vous avez connecté le dispositif IoLab (un clavier spécial avec des boutons poussoirs). En option, vous pouvez spécifier les paramètres des touches vocales (voir psycc -s)

## 5.5.6 et (Linux seulement)

Le soutien à l'eye-tracker Tobii est en cours de développement. Actuellement, le code est en cours de développement pour le modèle tx300. Cela ne fonctionne que sur linux

Exemple de comment informer PsyToolkit sur votre eye-tracker Tobii

```
options et tobii 12345
```

Le numéro 12345 est ici un exemple d'identification d'un tracker Tobii. Il est facultatif de donner ce numéro d'identification.

### sprites first (Linux seulement) 5.5.7



#### Utilisation avancée uniquement

L'option "sprites first" attirera les sprites avant qu'elle n'attire d'autres stimuli. Cela signifie que les sprites seront dessinés en arrière-plan, ce que vous souhaitez pour certaines situations.

### executable # 5.5.8



Le nom de fichier du programme exécutable. Par défaut, c'est experiment, mais il peut être défini avec l'option de ligne de commande -o et avec cette option.

L'option -o remplacera cette option. Dans la version en ligne, cette option sera ignorée.

# 5.5.9 egi (C only) #



Avec le nom ou l'adresse IP de l'hôte et le port IP (facultatif), il établit une connexion avec le système d'Electrical Geodesics, Inc. Ce système n'a pas été bien testé et est basé sur un ancien code. Vous n'avez pas besoin d'utiliser d'options si vous ne le souhaitez pas.

### Exemple de toutes les options de base

options resolution 1024 800 fullscreen mouse off set &my\_variable 100

# 5.5.10 vsync on off $\oplus$



Non pertinent pour la version en ligne et pour utilisateurs avancés uniquement. En mode Linux, il est "activé" par défaut, mais peut être désactivé, car certaines cartes graphiques ne permettent pas d'utiliser le vsync. PsyToolkit affichera un message d'erreur s'il n'est pas disponible.

S'il est réglé sur "off", l'ordinateur n'attend pas la synchronisation du vsync (c'est la valeur par défaut en Javascript). C'est pratique pour les tests, car l'expérience ne nécessite pas l'autorisation de l'administrateur principal lors de la compilation. En C, si vous utilisez l'option -t, c'est la valeur par défaut.

### Exemple de vsync désactivé

option vsync off

## 5.5.11 window (C only) #



Cela vous permet de mener l'expérience dans une fenêtre plutôt qu'en plein écran. C'est pratique pour les tests. Notez que le mode plein écran est le mode par défaut dans Linux, sauf si vous exécutez en mode test.

Comme PsyToolkit peut fonctionner dans deux environnements de programmation différents, certaines fonctions ne marchent pas dans tous les environnements. Les fonctions C ne fonctionnent que sur la version Linux qui est basée sur le langage de programmation C. Si vous utilisez la version en ligne, vous pouvez les ignorer complètement.

## screensize and screendistance (C only)



Vous pouvez définir les dimensions de l'écran et la distance entre les yeux et l'écran. Les unités sont en millimètres. Ces chiffres sont nécessaires si vous voulez utiliser l'option coordonnées polaires ou si vous voulez utiliser l'option de ligne de commande de la taille du stimulus du rapport. Si vous utilisez un système de coordonnées polaires (ce qui est rare), toutes les coordonnées xy doivent être spécifiées en centièmes de degrés (donc 100 = 1 degré, 200 = 2 degrés d'angle visuel). Ceci n'est pas encore bien implémenté (faites-moi savoir si vous en avez besoin).

### version # 5.5.13



Ce service est réservé aux utilisateurs avancés. Cela permettra uniquement de compiler un script avec la version psycc correspondante. C'est pratique si vous avez plusieurs versions de psycc installées, et si vous voulez juste vous assurer que le code fonctionne. Cette commande n'a aucun effet dans la version en ligne.

# 5.5.14 check\_refresh\_rate

Linux seulement. Vous pouvez vérifier si le taux de rafraîchissement est conforme aux attentes, et si ce n'est pas le cas, un message sera affiché à l'écran.

Ceci est utile si vous avez un moniteur avancé et que vous voulez vous assurer que le taux de rafraîchissement n'est pas réinitialisé à une valeur inférieure par un autre gestionnaire de bureau. Notez que seul XFCE dispose d'un moyen agréable de définir la fréquence de rafraîchissement (settings—display).

### Exemple de check\_refresh\_rate

```
options
resolution 1920 1080
check_refresh_rate 144
```

Dans l'exemple ci-dessus, nous avons réglé l'écran sur ce qui est aujourd'hui une résolution standard (pour en savoir plus sur ce qu'on appelle le "full HD"). Ensuite, nous disons à l'ordinateur que nous nous attendons à ce que la carte graphique et l'écran fonctionnent à un taux de rafraîchissement de 144 Hz. PsyToolkit ne peut pas le régler, mais il peut le vérifier. Si PsyToolkit constate que l'écran ne fonctionne pas à la fréquence de rafraîchissement prévue, il envoie un message puis s'arrête.

### Exemple de toutes les options de base

```
options
  fontdir
              /usr/lib/fonts
  datadir
              /home/user/mydatadir
  bitmapdir
              /usr/local/bitmaps
  sounddir
              /usr/local/snd
  origin
              topleft
  window
  resolution 1024 800
  screensize 1000 600
  screendistance 500
  coordinates polar
  vsync off
  escape
  egi 10.0.0.42
  parallelport in data out 1 5
  pcidio24 in a b out c low c high
  cedrus
  iolab
  executable myexperiment
  set &myinterval 100
  sprites first
```

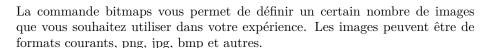
## 5.6 Chargement des stimuli 🏶

L'ordinateur doit savoir quels stimuli vous utilisez. Les stimuli sont généralement des images ou des sons, qui doivent être chargés. Vous pouvez également

charger des polices de caractères. Vous trouverez ci-dessous des explications sur la manière de charger des images, des sons et des polices.

Les noms de fichiers bitmap et sons ne doivent pas commencer par un chiffre, ni contenir d'espaces ou de caractères spéciaux (à l'exception du soulignement).

# 5.6.1 bitmaps #



La ligne bitmaps n'a pas de paramètres et est suivie de lignes décrivant l'image, chaque image ayant un nom et une description de fichier.

Ne mettez rien entre guillemets, et n'utilisez pas d'espaces dans les descriptions ou dans les noms de fichiers.

Si vous n'ajoutez pas de nom de fichier, cela suppose qu'un nom de fichier existe avec l'extension .png. Dans l'exemple ci-dessous, PsyToolkit suppose qu'un fichier house.png existe car aucune autre information bitmap n'est donnée. C'est en fait la méthode de travail recommandée (c'est certainement la plus simple).

### Exemple de bitmaps et comment s'y référer dans une tâche

```
bitmaps
house
funnyface funny.bmp
cookie stimuli/cookie.jpeg
smiley smiley

task MyTask
show bitmap funnyface
show bitmap cookie 100 0
```

Notez que pour les expériences en Javascript dans un navigateur, il existe deux autres façons de charger les bitmaps :

1. en utilisant le préfixe http:// (donc, au lieu d'un nom de fichier complet, donnez l'URL). Cette façon de procéder est logique dans des circonstances très particulières et n'est généralement pas recommandée, sauf si vous savez vraiment ce que vous faites. La raison est que cela peut réduire considérablement le temps de téléchargement des expériences en ligne. En effet, par défaut, tous les stimuli sont intégrés dans le fichier HTML via data:uri, ce qui peut conduire à des fichiers HTML volumineux.

Voici comment vous procédez dans PsyToolkit

### Exemple de chargement de bitmaps à partir de sites web externes

bitmaps house http://www.my-website.com/house.png task MyTask show bitmap house

2. utilisation de data:image/ uris

Pour plus d'informations sur le fonctionnement de l'URI, voir Wikipedia

Les données uri ne sont utilisées que dans des cas très rares et spéciaux, et ne sont généralement pas recommandées (parce qu'elles ne sont pas vraiment nécessaires, mais il existe des raisons techniques inhabituelles pour lesquelles quelqu'un pourrait vouloir les obtenir dans des conditions très inhabituelles)

### Exemple de chargement d'images utilisant une URI

bitmaps

testimage data:image/png;base64,iVBORwOKGgoAAAANSUhEUgAAAAQAAAADCAIAAAA71jmRAAAAGE1EQVQIW2P4Dwo

task MyTask show bitmap testimage

### $\mathrm{move}\ ^{\bigoplus}$ 5.6.2



move et relocate ne sont pas souvent utilisés. Ils sont surtout réservés aux animations de base et sont alors très pratiques. move ne donne pas une animation fluide, c'est juste un changement de lieu. Voir la leçon sur les animations simples pour plus de détails. Parfois, vous voulez changer l'emplacement d'un stimulus que vous avez déjà créé à l'écran. Cela peut être utile pour les animations simples.

Vous pouvez le faire avec deux fonctions différentes dans les tâches :

- *move*
- relocate (décrit ci-après)

L'exemple suivant montre un stimulus au centre de l'écran et, après 1000 ms, déplace le stimulus de 20 pixels vers la droite, sans changer l'emplacement y. Ainsi, le déplacement donne la quantité de changement X et Y (alors que le déplacement donne une nouvelle position X/Y complète).

### Exemple de déplacement d'un stimulus

```
bitmaps
  smiley.png
task MyTask
  show bitmap smiley
  delay 1000
  move 1 20 0
```

# 5.6.3 relocate #



relocate est très similaire à move, sauf que vous ne faites que donner les nouvelles coordonnées.

### Exemple de relocalisation d'un stimulus

```
bitmaps
  smiley.png
task MyTask
  show bitmap smiley
  delay 1000
  relocate 1 100 150
```

# 5.6.4 sounds #



Cela fonctionne comme les commandes "bitmaps", simplement pour indiquer à l'ordinateur quels sons sont chargés. Vous pouvez utiliser n'importe quel format de fichier son, bien que le format de fichier wav soit le plus largement utilisé et recommandé pour les études en ligne.

### Exemple de sons et comment s'y référer dans une tâche

```
sounds
  barkingdog bark.wav
 phone
             ringingphone.mp3
task MyTask
  sound phone
 delay 300
```

Comme pour les images bitmap, vous pouvez également charger un son à partir d'un serveur web (en utilisant http://). L'avantage est que cela permet d'accélérer le temps de chargement.

Vous pouvez également utiliser des sons dans le bloc, par exemple pour faire jouer de la musique lorsque le participant effectue la tâche. Cela peut être utilisé pour tester l'effet de différents types de sons, de musique, de bruit de fond sur l'exécution de la tâche. Voir le son dans un bloc.

## 5.6.5 sound frequencies

Apart from playing sound files, you can also play specific tones or frequencies. This also uses the sound instruction, but now followed by the word freq. Below are various examples:

Outre la lecture de fichiers sonores, vous pouvez également lire des tonalités ou des fréquences spécifiques. Pour ce faire, on utilise également l'instruction sound, mais elle est suivie du mot freq. Vous trouverez ci-dessous plusieurs exemples :

Jouez un son sinusoïdal de 500 Herz (Hz) pendant 1 seconde (1000 ms)

sound freq sine 400 2000

Jouez un son d'onde carrée de  $400~{\rm Herz}$  (Hz) pendant 2 secondes (2000 ms)

sound freq sine 400 2000

Jouez un son d'onde triangulaire de 700 Herz (Hz) pendant 1,5 seconde (1500 ms)

Jouez un son d'onde triangulaire de 700 Herz (Hz) pendant 1,5 seconde (1500 ms).

Faites jouer une onde en dents de scie de 600 Herz (Hz) pendant une demi-seconde (500 ms)

sound freq sawtooth 600 500

Comment arrêter une fréquence sonore. Pour cela, vous utilisez l'instruction silence, mais vous devez utiliser un nombre négatif. Le dernier son commencé est -1, celui qui a commencé avant est -2, etc. Voir l'exemple ci-dessous.

Arrêter un son de fréquence avant sa durée réelle

```
sound freq sine 400 5000
delay 500
sound freq sawtooth 700 5000
delay 500
silence -1 # stops the sawtooth
delay 500
silence -2 # stops the sine
```

# 5.6.6 fonts #



Par défaut, si vous ne précisez pas à l'ordinateur quelle police vous souhaitez utiliser, PsyToolkit utilisera Arial 20 points. Vous n'avez pas besoin de définir une police si vous ne voulez rien d'autre qu'un Arial de taille 20 points.

Vous pouvez décrire ici les polices à utiliser. Vous pouvez utiliser les polices intégrées (arial, times, courier) ou charger les polices ttf (depuis la version) 2.3.3 également en Javascript. Si vous les utilisez en ligne, vous devez télécharger votre ttf comme vous téléchargez n'importe quel autre stimulus.

Notez que les mêmes polices portent des noms différents. Si vous ne connaissez pas les types les plus courants, consultez WikiPedia pour obtenir des informations sur Arial, des informations sur Times New Roman et des informations sur Courier.

Vous pouvez utiliser l'un des trois types de police par défaut : arial, times ou courier. Si vous utilisez l'une de ces trois polices, vous n'avez pas besoin de télécharger ou d'inclure un fichier de police, car ces polices sont déjà disponibles sur les ordinateurs. Sous Linux, Arial=FreeSans.ttf, Times=FreeSerif.ttf, et Courier=FreeMono.ttf.

### Exemple de polices et comment s'y référer dans une tâche

```
fonts
  arial 18
 myfont times 20
 mysmallfont arial.ttf 20
 mybigfont
              arial.ttf 40
task MyTask
  font mysmallfont
  show text "hello"
  delay 1000
  font mybigfont
  show text "world"
  delay 1000
  font arial
```

5.7. LES TÂCHES 59

```
show text "Some text in standard arial"
font myfont
show text "Some text in Times New Roman"
delay 1000
```

voir aussi Show text

# 5.6.7 videos #

Vous pouvez utiliser des vidéos dans votre expérience. Vous devez indiquer à PsyToolkit où les vidéos sont stockées sur Internet. Vous ne pouvez pas les stocker sur le serveur PsyToolkit lui-même.

Il est plus facile de télécharger un fichier vidéo sur le Google Drive ou sur la Dropbox. Il y a une leçon sur la façon de le faire en détail (en anglais).

Vous pouvez donner n'importe quel nom à la vidéo, par exemple, dans l'exemple ci-dessous, clip1 et clip2 ont été choisis, mais il pourrait tout aussi bien s'agir de my\_video1 et my\_video2.

### Exemple d'une section vidéo avec deux clips

```
videos
```

```
clip1 https://www.dropbox.com/s/kzo7sx7hkdai16i/beetle.mp4?dl=0
clip2 https://www.dropbox.com/s/bzbs6qpk7ht9yyu/frog.mp4?dl=0
```

### Les tâches 🟶



Dans PsyToolkit, une tâche décrit (typiquement) la séquence d'un seul essai d'une expérience.

Les tâches sont la partie la plus difficile du langage de script, car elles impliquent l'écriture de véritables instructions informatiques séquentielles pour montrer des stimuli, attendre des réponses, etc. Beaucoup de choses sont possibles, et il existe de nombreux types d'instructions différentes. Chaque type d'instruction est décrit ci-dessous.

Avant de décrire les instructions, un certain nombre d'éléments fondamentaux du langage sont décrits, à savoir la structure des tâches, les variables et les repères temporelles

### Comment les tâches sont structurées 5.7.1



Les tâches servent en fait à programmer ce qui se passe dans un essai de paradigme expérimental. Une tâche n'est qu'une séquence d'instructions, terminée par une ligne vide. Les instructions contiennent les grandes parties d'information suivantes :

- Où se trouve une description des différentes conditions expérimentales. Cela peut être fait avec l'état de la table
- Comment le participant doit-il réagir exactement (par exemple, quelles touches du clavier, ou avec la souris). Cela peut être fait, par exemple, avec keys et readkeys.
- Les stimuli utilisés, ainsi que le moment et le lieu où ils doivent être présentés. Cela peut se faire avec show et sound.
- Les données qui doivent être enregistrées dans un fichier pour une analyse ultérieure des données. Cela se fait avec l'instruction save

### le concept de variables 5.7.2



Comme tout langage informatique, vous pouvez utiliser des variables. Il existe des variables globales et des variables locales. En principe, vous pouvez toujours utiliser des variables globales. Les variables locales ne sont utilisées que dans une tâche, tandis que les variables globales peuvent être utilisées n'importe où dans la tâche et le code de bloc (voir aussi blocs). Vous pouvez définir les valeurs initiales des variables globales dans les options.

Vous pouvez définir une variable avec l'instruction set.

#### Exemple de définition et d'utilisation des variables

```
task MyTask
  set $x 100
  set &y 10
  show bitmap FunnyFace $x &y
```

- Les variables ne peuvent avoir que des valeurs entières. Un entier est un nombre entier. Par exemple, \$x est 100, il ne peut pas être 100.342, car ce dernier n'est pas un nombre entier.
- Le signe égal n'est pas utilisé pour attribuer des valeurs aux variables ! C'est-à-dire pas de symbole "=" comme on le ferait dans la plupart des langages de programmation.
- Les variables ne peuvent pas commencer par un nombre, ne peuvent pas contenir d'espaces, de signes moins, de points ou de caractères spéciaux, à l'exception du soulignement.

5.7. LES TÂCHES

De bons exemples.

- \$MyVariable1
- ma variable 1
- \$xyz



Les écritures suivantes ne sont pas autorisées !!!!!!

- \$My Variable 1 (raison : pas d'espaces autorisés)
- \$My.variable.1 (raison : pas de points autorisés)
- \$My-variable-1 (raison : pas de signes moins autorisés)
- \$My\*variable (raison : aucun caractère spécial tel que "\*" n'est autorisé)

61

• 123my\_variable (raison : ne peut pas commencer par un numéro)

Il existe également des types particuliers de variables écrites en majuscules, comme RT, STATUS, TABLEROW, TASKNAME, BLOCKNAME, BLOCK-NUMBER, BLOCKORDER. Elles vous donnent simplement accès à des informations importantes sur une réponse ou sur l'état de la tâche.

## 5.7.3 Le concept d'horodatage



l'horodatage (timestamps) est un type spécial de variable, mais il n'est utilisé que dans des scripts très avancés et n'est normalement pas nécessaire. Chaque fois que vous définissez un timestamps, l'heure courante est enregistrée.

# 5.7.4 La fonction set #



Avec set, vous pouvez assigner des valeurs à des variables.

set [\$&] [nom de la variable] [nouvelle valeur, variable ou expression]

Comme nous l'avons expliqué à propos des "variables", la fonction set peut fixer la valeur des variables.

- Une variable est un élément de base dans tout langage de programmation. Une variable est simplement un symbole qui contient une valeur. Par exemple, x = 10. PsyToolkit ne peut fonctionner qu'avec des nombres entiers comme valeurs.
- set est une fonction relativement avancée, et lors de l'apprentissage de PsyToolkit, vous n'en avez pas immédiatement besoin, et vous pouvez écrire des scripts sans set.

• set est une fonction relativement avancée, et lorsque vous apprenez Psy-Toolkit, vous n'en avez pas immédiatement besoin, et vous pouvez écrire des scripts sans set. set est une variable locale qui a une valeur (le \$ indique qu'elle est locale)

Définir une variable locale à une valeur (le \$ indique qu'elle est locale)

```
set $x 10
```

définir une variable globale à une valeur (le & indique qu'elle est golabale)

```
set &x 10
```

Dans ce qui suit, la nature locale ou globale de la variable n'est pas pertinente pour expliquer l'ensemble, et nous travaillons simplement avec \$x

Prendre la valeur d'une autre variable

```
set $x &y
```

Augmenter la valeur d'une variable

```
set $x 10
set $x increase 2 # after this, $x has value 12
```

Diminuer la valeur d'une variable (le pas par défaut est de 1)

```
set $x 10
set $x decrease # after this, $x has value 9
```

Utiliser une expression mathématique

```
set x = x = x = x + 1
```

Tirer une valeur aléatoire

```
set $x random 1 10
```

Tirer une valeur aléatoire avec un pas de 2

```
set $x random 2 100 2
```

5.7. LES TÂCHES 63

Tirer une valeur aléatoire d'un ensemble spécifique de valeurs

```
set $x random from 1 2 3 5 10
```

Régler la valeur de la carte d'entrée/sortie pcidio24 (si vous en avez une !) (pour les utilisateurs avancés, Linux uniquement)

```
set $x pcidio24 a # set to value of register A
```

Régler sur les millisecondes qui se sont écoulées depuis le début de l'expérience (pour les utilisateurs avancés)

```
set $x time-since-start
```

Régler sur la différence en millisecondes de deux horodatages (pour les utilisateurs avancés)

```
timestamp MyTime1
delay 500
timestamp MyTime2
set $x timestamp-diff MyTime1 MyTime2 # now x should be 500
```

obtenir l'horodatage en secondes et millisecondes (pour les utilisateurs avancés)

```
timestamp MyTime1
dely 2500
timestamp MyTime2
set $x timestamp-seconds MyTime2 # value should be 2
set $x timestamp-milliseconds MyTime2 # value should be 2500
```

obtenir le temps en millisecondes depuis le 1<br/>er janvier 1970 (pour les utilisateurs avancés)

```
set $x unix-time
save $x
```

obtenir le numéro de l'image actuellement sous la souris (pour les utilisateurs avancés)

```
bitmaps
myface
yourface
smileyface

task test
show bitmap yourface -100 0 ## first bitmap
show bitmap smileyface 200 200 ## second bitmap
readmouse 1 1 1000
set $myMouseX MOUSE_X # current mouse x-coordinate
set $myMouseY MOUSE_Y # current mouse x-coordinate
set $x bitmap-under-mouse $myMouseX $myMouseY ## $x should be 1
```

Vous pouvez définir des variables de texte. Les variables de texte commencent toujours par le caractère %, juste pour indiquer à PsyToolkit que nous avons une variable de texte ici.

Vous pouvez créer une nouvelle variable de texte en collant différents éléments ensemble, puis l'afficher. Ceci est particulièrement utile pour donner à un participant un feedback sur le TR ou le nombre d'erreurs, comme dans l'exemple ci-dessous.

### définir une variable texte

```
task test
  keys x
  show rectangle 0 0 100 100 255 0 0
readkey 1 2000
  set $myvar random 1 100
  set %y "Hello " RT " " $myvar
  show text %y
  delay 2000

block testx
  tasklist
    test 2
  end
```

fixer la variable à la durée du stimulus sonore (fichier)

```
set $x duration my_sound
```

mettre une variable texte comme sous-chaı̂ne d'une autre variable texte ; ici, on prend les lettres 1 à 5

5.7. LES TÂCHES 65

```
task test
  keys x
  show rectangle 0 0 100 100 255 0 0
  readkey 1 2000
  set $myvar random 1 100
  set %y "Hello " RT " " $myvar
  set %k %y[1,5]
  show text %k
  delay 2000
block testx
  tasklist
      test 2
  end
```

Pour les sous-chaînes, vous utilisez un système d'indexation :

```
set %text1 "PsyToolkit"
set %text2 text1[1] \rightarrow text2 est "P" set %text2 text1[-1] \rightarrow text2 est "t" (dernière
lettre) set \%text2 text1[-2] \rightarrow text2 est "i" (avant-dernière lettre)
set %\text{text2 text1}[1,-1] \rightarrow \text{text2 est "PsyToolkit"} (première à dernière lettre) set
\%text2 text1[1,3] \rightarrow text2 est "Psy" (lettre 1 à lettre 3) set \%text2 text1[-3,-1]
→ text2 est "kit" (troisième à dernière lettre)
```

### définir une variable liste 🕏 5.7.5



Ces instruyctions sont destinés aux utilisateurs un peu plus expérimentés. Assurez-vous d'abord de comprendre les instructions relatives aux tableaux. Dans PsyToolkit, un tableau est simplement une liste de nombres entiers, tels que 1 2 3 4 5. \* Chaque variable de tableau est une variable globale, c'est-à-dire qu'elle reste la même tout au long des essais et des différentes tâches.

En bref, vous pouvez appliquer les fonctions suivantes (vous trouverez des exemples plus détaillés plus loin):

```
Instruction | Ce que cela fait | Exemple | |-------
                    une suite de valeurs | set &&x 4 2 10 -1 | | sort | trier une suite de valeur | set
&&x sort | | range | Remplir une liste avec un intervalle | set &&x range 1 5 | |
shuffle | mélanger les valeurs d'une liste de façon aléatoire | set &&x shuffle
| remove | supprimer une valeur d'une liste | set &&x remove first | | sample
prendre n valeurs dans une liste de façon aléatoire | set &&x &&y sample 10
| | uniq | supprimer toutes les valeurs apparaissant plus d'une fois | set &&x
uniq | append | ajoute une valeur à la fin de la liste | set &&x prepend 10
```

Toutes les instructions de cette sous-section s'appliquent uniquement aux tableaux entiers (texte ou numérique). Une section distincte est consacrée à l'extraction de valeurs de tableaux.

### Définir une liste de nombre

```
task test
set &&my_array 1 2 5 2 4 1 0 10
```

Supprime toutes les valeurs d'une (après quoi elle sera vide)

```
task test
  set &&my_array clear
```

Remplir la liste avec un intervalle, dans ce cas 1 2 3 4 5

```
task test
  set &&my_array range 1 5
```

L'instruction ci-dessous supprime une valeur du tableau, mais vous ne saurez pas laquelle (voir ci-dessous pour l'autre suppression pour faire cela).

Si une liste a une valeur, supprimer cette valeur (dans ce cas, la valeur 3)

```
task test
  set &&my_array 1 2 3 4
  set &&my_array remove value 3
```

Supprimez la première valeur de la liste Par exemple : 1 2 3 4 devient 2 3 4

```
task test
  set &&my_array 1 2 3 4
  set &&my_array remove first
```

Supprimez la dernière valeur de la liste. Par exemple : 1 2 3 4 devient 1 2 3

```
task test
  set &&my_array 1 2 3 4
  set &&my_array remove last
```

5.7. LES TÂCHES 67

Retirer une position spécifique de la liste. Par exemple :  $10\ 4\ 20\ 100$  devient  $10\ 20\ 100$ 

```
task test
  set &&my_array 10 4 20 100
  set &&my_array remove position 2
```

Retirer au hasard une position de la liste

```
task test
  set &&my_array remove random
Fill the array with the value 2, repeated 10x (thus becoming 2 2 2 2 2).
task test
  set &&my_array 2 times 5
```

Remplir la liste avec 3 valeurs aléatoires entre 5 et 10

```
task test
  set &&my_array random 5 10 times 3
```

Vous pouvez également répéter des listes Ainsi, dans l'exemple suivant, nous avons une liste &&mes\_numéros avec le contenu 1 2 3. Après avoir fait cela fois 3, nous avons 1 2 3 1 2 3 1 2 3.

Répétez unne liste 3 fois

```
task test
  set &&my_numbers 1 2 3
  set &&my_array &&my_numbers times 3
```

Trier les valeurs d'une liste (du plus petit au plus grand)

```
task test
  set &&my_numbers sort
```

Trier les valeurs d'une liste (de la plus grande à la plus petite)

```
task test
  set &&my_numbers sort reverse
```

Supprimez toutes les valeurs d'une liste qui apparaissent plus d'une fois

```
task test
 set &&my_numbers uniq
```

Vous pouvez également utiliser des tableaux de texte, en les faisant commencer par %%, voir l'exemple ci-dessous.

Supprimez toutes les valeurs d'une liste qui apparaissent plus d'une fois

```
task test
  set %%my_list "Hello" "this" "is" "PsyToolkit"
```

## définir une variable à partir d'une liste



- Ces instructions sont destinées aux utilisateurs un peu plus expérimentés. Assurez-vous d'abord de bien comprendre les instructions set.
- Dans PsyToolkit, un tableau est simplement une série de nombres entiers, tels que 1 2 3 4 5.

```
valeur dans la liste | | remove | prendre et enlever d'une liste (first, last ou une
position spécifique dans la liste) | | use | utiliser mais pas enlever d'une liste
(first, last, random ou une position spécifique dans la liste) | | size | prendre un
nombre d'items dans une liste | | join | mettre dans une varaible text la réunion
d'éléments textuels d'une liste | |-----|
```

Dans l'exemple ci-dessous, nous pouvons définir une liste avec les valeurs 4 5 2 10. Dans cette liste, la deuxième valeur (c'est-à-dire la position 2) contient 5. Ainsi, &x sera égal à 2 après cela.

Repérer la position d'une valeur spécifique dans une liste

```
task test
  set &&my_area 4 5 2 10
  set &x &&my_array locate 5
```

Dans l'exemple suivant, la variable globale &x aura la valeur 4 (la première), et la liste &&my\_area sera réduit à 5 2 10.

Obtenir et supprimer une valeur d'une liste

5.7. LES TÂCHES 69

```
task test
  set &&my_area 4 5 2 10
  set &x &&my_array remove first
```

Dans l'exemple suivant, la variable globale &x obtiendra la valeur 10 (la dernière), et la liste &&my area sera réduit à 4 5 2.

### Obtenir et supprimer une valeur d'une liste

```
task test
  set &&my_area 4 5 2 10
  set &x &&my_array remove last
```

Dans l'exemple suivant, la variable globale &x prendra la valeur 5 (qui se trouve à la position retirée 2). Le tableau sera alors 4 2 10

### Obtenir et supprimer une valeur d'une liste

```
task test
  set &&my_area 4 5 2 10
  set &x &&my_array remove 2
```

• Note: au lieu de remove, vous pouvez utiliser l'instruction use. La seule différence est que le tableau ne sera pas modifié. vous pouvez également supprimer une valeur au hasard.

Obtenir la taille d'une liste (dans l'exemple, &x deviendra donc 4)

```
task test
  set &&my_area 7 5 2 10
  set &x &&my_array size
Sort and join an array with spaces
task test
  set %%my_array "Smith" "hello" "John"
  set %%my array sort
  set %my_sentence %%my_array join " "
```

### Synthèse de la fonction set ## 5.7.7



Le set n'est pas nécessaire pour les débutants! Il permet vraiment une flexibilité dans votre code, si vous le voulez et en avez besoin. Pour chaque fonction set, vous pouvez utiliser des variables locales ou globales. Ne vous inquiétez pas de la différence entre les variables locales et globales ; en cas de doute, utilisez des variables globales (en commençant par le signe &).

### Usages basiques

Instructions de base Assigner une valeur à une variable locale	Exemple set \$x 10	Résultat \$x contient 10 (seules les valeurs entières sont permisses)
Assigner une valeur à une variable globale	set &g 12	&g contient 12
Assigner une valeur à partir d'une autre variable	set &h RT	avec une commande <i>readkey</i> plus tôt, donne la valeur de RT à &h
Tirer une valeur aléatoire	set &x random 1 5	&x contient une des valeurs 1,2,3,4 ou 5
Tirer une valeur	set &x random	&x contient une des valeurs
aléatoire avec un pas	1 10 2	1,3,5,7 ou 9
Augmenter la valeur d'une variable	set &x increase	si &x contient 5, après la commande il contiendra 6
Augmenter la valeur d'une variable avec une valeur	set &x increase	si &x contient 5, après la commande il contiendra 8
Diminuer	set &a decrease	fonctionne comme increase, sauf qu'elle diminue la valeur
Assigner une valeur à partir d'une expression	set $y$ expression $x * 3 + 5$	si &x contient 5 alors \$y sera de 20

### Usages avancés

gestion du temps Temps depuis le début de l'expérience en ms	Exemple set \$j time- since-start	Resultat si votre expérience a démarré depuis 5 secondes \$j contiendra 5000
Calculer la différence entre deux horodatages	set \$t timestamp-diff Time1 Time2	si Time1 a été mesuré plus tôt que Time2, \$t contiendra la différence en millisecondes
Calculer la différence entre deux horodatages en secondes	set \$t timestamp- seconds Time1	l'horodatage en secondes
Calculer la différence entre deux horodatages en millisecondes	set \$ms timestamp- milliseconds Time1	l'horodatage en millisecondes

Contrôles de la souris Exemple Resultat

Trouver le numéro du stimulus cliqué	set \$b bitmap-	si le dernier stimulus cliqué est le 5, \$b contient 5
Trouver le numéro du stimulus cliqué en ne considérant que certains numéros	under-mouse set \$b bitmap- under-mouse range 10 15	si le dernier stimulus cliqué est 11 a été cliqué, \$b contiendra 11. Les autres stimuli seront ignorés. Ceci est pratique pour les stimuli qui se chevauchent.

Autres fonctions	Exemple	Resultat
avancées		
Trouver le dernier	set \$c	Si vous avez montré le stimulus
stimulus affiché	show-counter	10, \$c contient 10

Autres fonctions Linux seulement	Exemple	Resultat
Récupérer la valeur	set \$z	mettre \$z à la valeur de 8 bits du
d'une carte	pcidio24 a	canal a (valeur comprise entre 0 et
PCI-DIO24		255)
Position x courante à	set \$x	Si on affiche un symbole 2, met dans
l'écran du symbole	sprite-x 2	\$x la coordonnée x à l'écran du symbole
Position y courante à	set \$y	Si on affiche un symbole 2, met dans
l'écran du symbole	sprite-y 2	\$y la coordonnée y à l'écran du symbole

### Instructions communément utilisées # 5.8



Il existe de nombreuses instructions différentes pour montrer les stimuli, enregistrer et sauvegarder les réponses. Elles sont toutes énumérées ci-dessous avec un bref exemple de leur utilisation. Cette section décrit ceux qui sont le plus souvent utilisés, et que vous devez vraiment comprendre.

Avant de commencer, voici une liste alphabétique des instructions les plus couramment utilisées :

### présentation des stimuli

- Show montre un stimulus sur l'écran
- clear supprime un stimulus de l'écran
- sound joue un son

### la mesure des réponses et des délais de réponse

- keys indique à l'ordinateur quelles sont les touches utilisées dans une tâche
- readkey attend une pression sur le clavier
- readkeys attent qu'on appuie plusieurs fois sur le clavier
- readmouse attend un clic ou un mouvement de souris
- textbox permet d'entrer du texte, y compris des caractère spéciaux
- choose la sélection de stimuli multiples avec la souris
- rate, la façon la plus simple d'inclure l'échelle de Likert dans votre expérience

### Gérer le temps

• delay d'attente en millisecondes

### sauvegarde des données

• save, sauvegarde les données dans un fichier

### informer l'ordinateur sur les conditions d'expérimentation

• table indique où se trouvent les conditions de l'expérience

### conditions

- if, faire quelque chose qui dépend de d'une ou des variable(s)
- while faire quelque chose qui dépend de d'une ou des variable(s)
- set définir une variable
- set form array paramétrer une variable à partir d'une valeur d'une liste

Chacune de ces instructions est détaillée ci-dessous.

# 5.8.1 keys #

#### syntaxe

keys [liste de touches]

Au début de la description d'une tâche, vous devez indiquer à l'ordinateur les touches du clavier sur lesquelles les participants devront éventuellement appuyer. Cette ligne doit être l'une des premières lignes de la description de la tâche.

Les touches données seront associées à un numéro de clavier, commençant par un. Vous pouvez l'utiliser lors de la sauvegarde des données. La variable KEY contiendra cette valeur lorsqu'une touche a été enfoncée.

#### exemple d'utilisation de keys utilisant deux touches

#### keys a z

Liste de toutes les touches disponibles sur toutes les plateformes :

• lettres : a à z les chiffres: 0 à 9

• touches spéciales : enter, capslock, tab, space, end, home, insert • touches spéciales : escape, slash, backslash, quote, comma, period

touches fléchées : haut bas droite gauche

clavier numérique (keypad) : kp0 kp1 kp2 kp3 kp4 kp5 kp6 kp7 kp8 kp9 clavier numérique : kp period kp slash kp star kp minus kp plus kp\_enter

Les touches suivantes ne sont pas disponibles en mode Javascript

• touches de déplacement : lshift rshift • touches de contrôle : lcontrol rcontrol

• touches alt : lalt ralt

• touches logo: lsuper rsuper

#### mouse on or off $\P$ 5.8.2



Vous pouvez masquer (off) ou afficher (on) le curseur de la souris. Regardez les lignes suivantes que vous pouvez utiliser dans une tâche.

#### Exemple d'affichage ou de masquage du curseur de la souris dans une tâche

```
task MyTask
  show bitmap smileyface
  mouse hide
  delay 1000
  show bitmap smileyface
  mouse show
  delay 1000
```

## 5.8.3 table $\clubsuit$

syntaxe

table [nom de la table]

Une instruction table permet de définir une table (qui est définie séparément de la tâche, et elle contiendra des informations sur les différentes instructions expérimentales).les tables vous permettent d'avoir une description de tâche pour différentes conditions. De cette façon, vous pouvez faire varier la couleur, la position ou la taille des stimuli. Chaque ligne d'une table sera considérée comme la description d'une condition expérimentale différente. C'est un élément clé des scripts PsyToolkit, bien que vous puissiez aussi écrire des scripts sans table !

Si vous avez plus d'une tâche et plus d'une table, vous devez spécifier dans votre tâche quelle table lui appartient. La ligne du tableau choisie sur une épreuve donnée se trouve dans la variable TABLEROW.

#### Exemple de table

table MyTable

### 5.8.4 show **\$\frac{1}{4}**

syntaxe

show [bitmap|text|rectangle|circle|video](X,Y)(...)

C'est une instruction essentielle qui permet de montrer les stimuli des types suivants:

- un rectangle
- un cercle
- une ligne de texte
- une image
- une vidéo
- un fond de couleur

Le nombre de paramètres donnés pour une instruction *show* peut varié. Vous trouverez ici une explication détaillée de la façon dont vous présentez vos stimuli à l'écran, là où vous voulez qu'ils soient.

### 5.8.5 show bitmap \$\frac{\pi}{4}\$

#### syntaxe

show bitmap [nom de l'image](X,Y)

Imaginez que vous avez chargé une image dans la section "bitmap" de votre description de tâche. Vous pouvez maintenant la montrer. Il vous suffit de spécifier le fichier image, et celui-ci s'affichera au centre de l'écran. Vous pouvez également spécifier les coordonnées X et Y (facultatif).

#### Exemple d'usage de show bitmap

show bitmap MyBitmap show bitmap MyBitmap 200 10

Plus d'informations sur la façon de montrer les images sont disponibles dans cette leçon.

## 5.8.6 show rectangle

\*syntaxe\*\*

show rectangle X,Y,Width,Height, Red,Green,Blue

Vous pouvez afficher des rectangles colorés, en spécifiant la position X et Y ainsi que la largeur et la hauteur. Vous devez également spécifier les valeurs Rouge, Vert et Bleu. Cela signifie que vous aurez 7 paramètres supplémentaires!

Dans l'exemple ci-dessous, un rectangle est affiché à la position 0,0 et il a une largeur et une hauteur de 100 pixels. Les valeurs rouge/vert/bleu sont fixées à 255/0/0. La valeur maximale dans chacun de ces trois canaux de couleur est 255. Étant donné que seul le rouge est spécifié, le rectangle sera rouge.

PsyToolkit utilise le modèle de couleur Rouge-Vert-Bleu (RVB). Chaque couleur peut être spécifiée comme une combinaison de rouge, de vert et de bleu. Pour chacune de ces trois couleurs de base, vous devez donner une valeur comprise entre zéro et 255. Vous pouvez créer n'importe quelle couleur, voir le modèle RVB sur wikipedia. Il existe des sites web où vous pouvez créer vos propres couleurs :

Couleur	Rouge	Vert	Bleu
Blanc	255	255	255
Rouge pur	255	0	0
Bleu pur	0	0	255
vert pur	0	255	0
Jaune	255	255	0
Gris	128	128	128
Orange	255	128	0
Rose	255	100	180

#### Exemple d'usage de show rectangle

show rectangle 0 0 100 100 255 0 0

### 5.8.7 show circle $\clubsuit$



Montrer un cercle. Simailaire à show rectangle.

#### syntaxe

show circle X,Y,Radius, Red,Green,Blue

Le troisième paramètre est le rayon en pixels (donc le diametre est le double du rayon).

#### Exemple d'usage de show circle

show circle 0 0 100 255 0 0

## 5.8.8 show background #



#### syntaxe

show background (Red, Green, Blue)

Il existe également un moyen plus simple de le définir en utilisant l'option couleur de fond.

Parfois, vous voulez que l'ensemble de l'écran ait une couleur spécifique. En principe, vous pouvez simplement dessiner un rectangle de la taille de l'écran. C'est exactement ce que fait la fonction "show background". Il vous suffit de donner les trois paramètres de couleur et l'écran entier sera rempli de cette

couleur. Sinon, elle est traitée comme un stimulus d'écran normal, elle a un numéro et peut être effacée.

#### Exemple d'usage de show background

```
show background 100 100 100 ## creates a grey background
delay 500
show rectangle 100 200 200 0 0 ## draw red rectangle
delay 1000
clear -1 ## erase the the rectangle
```

Vous n'avez pas besoin d'effacer le fond d'écran. Il restera jusqu'au prochain essai (ce qui est en général ce que les gens veulent)

#### show text 5.8.9



L'instruction show text montre un texte.

#### Exemple basique d'usage de show text pour Linux

```
table MyFirstTable
  10 "some text"
  15 "some other text"
task MyTask
  show text "Some green text" 0 0 \, 0 255 0
  show text @2 0 0 0 255 0
```

Par défaut, si vous ne dites pas à l'ordinateur quelle police vous voulez utiliser, PsyToolkit utilisera Arial 20 points. Vous n'avez pas besoin de définir une police si vous ne voulez rien d'autre qu'un Arial de taille 20 points. Il existe de nombreuses polices ttf disponibles gratuitement sur le web. Par exemple avec Font Squirrel. Si vous le souhaitez, vous pouvez également spécifier une police et une taille de police dans la section fonts.

#### syntaxe

```
show text [text](X,Y)(Red,Green,Blue)
```

Exemple basique d'usage de show text avec une font spécifique

```
fonts
MyArial comic.ttf 20

table MyFirstTable
10 "some text"
15 "some other text"

task MyTask
font MyArial
table MyFirstTable
show text "Some green text" 0 0 0 255 0
show text @2 0 0 0 255 0
```

L'alignement par défaut est "centré". Cela signifie que si vous demandez de présenter quelque chose à la position 10,10, c'est là que se trouve le centre du mot. Voir la façon de définir l'alignement dans l'exemple ci-dessous. Notez également que vous pouvez définir la couleur du texte pour toutes les commandes "show text" suivantes, vous pouvez ainsi afficher un mot en couleur.

Alors que "show text …" sert à afficher le texte, vous pouvez régler l'alignement des couleurs séparément. C'est pratique si vous voulez par exemple présenter plusieurs textes dans la même couleur.

#### Exemple d'instructions text et show text avec alignement et couleur

```
fonts
   arial 20

task MyTask
   text color yellow
   text align center
   show text "Some text with alignment: center" 0 -100
   text align left
   show text "Some text with alignment: left" 0 0
   text align right
   show text "Some text with alignment: right" 0 100
   delay 10000

block test
   tasklist
   MyTask 1
   end
```

• Les couleurs de mots suivantes sont actuellement implémentées : blanc, jaune, rouge, vert, bleu, rose, violet, noir, gris, orange, rose.

• Pour la "couleur du texte", vous pouvez également spécifier la couleur avec trois valeurs décimales ou sous forme de code hexadécimal à 6 caractères. Dans l'exemple ci-dessous pour la même couleur jaune est déclarée de trois manières différentes. Cet exemple n'a aucun sens dans une expérience réelle (pourquoi spécifier la même couleur trois fois de suite?), mais montre simplement comment utiliser l'instruction "text color".

#### Exemple d'usage de text color

```
# specify color as word:
text color yellow
# specify color as Red Green Blue code:
text color 255 255 0
# specify color as Hexadecimal value:
text color FFFF00
```

Vous pouvez également utiliser un nom de couleur directement dans la ligne de texte de l'instriction show, comme suit :

#### Exemple de nom de couleur

```
show text "Hello" 0 0 red
```

## 5.8.10 show video $\clubsuit$



#### Exemple de vidéo au centre de l'écran

```
show video my_clip
```

Exemple d'une vidéo qui peut être placée comme les autres stimuli

```
show video my_clip -100 100
```

#### Exemple d'une vidéo muette (sans son)

```
videocontrol my_clip mute
show video my_clip -100 100
```

### 5.8.11 clear #

syntaxe

clear [Numéro d'ordre de l'image ou liste de numéro d'ordre d'images]

Pour effacer un stimulus que vous mettez à l'écran avec "show", vous pouvez l'effacer avec "clear". L'appel "clear" prend au moins un paramètre. Le nombre correspond au numéro des appels de show. Par exemple, si vous voulez effacer un bitmap particulier qui a été montré comme le deuxième événement de show dans votre tâche, utilisez "clear 2". Vous pouvez également utiliser des numéros de référence négatifs pour vous référer aux stimuli précédents : -1 correspond à la dernière image bitmap présentée! L'utilisation des références négatives est en fait beaucoup plus facile à utiliser, car de cette façon vous n'avez pas besoin de compter les bitmaps.



Le compteur des images est remis à zéro à chaque essai!

#### Exemple d'utilisation de clear

```
show bitmap redcircle
show bitmap greencircle
delay 500
clear 2 # clears the second presented bitmap
# or
show bitmap redcircle
show bitmap greencircle
delay 500
clear -1 # clears last presented bitmap, that is "green circle"
# or
show bitmap redcircle
show bitmap greencircle
delay 500
clear 1 2 # clears both bitmaps
```

Parfois, vous voulez effacer tellement de stimuli qu'il est désagréable de devoir taper tous les numéros S'ils sont dans une plage. Vous pouvez spécifier la plage. Par exemple, clear range 1 10 efface les stimuli 1 2 3 4 5 6 8 9 10. Il est rare que vous souhaitiez effacer tout l'écran. Ce n'est pas recommandé pour raisons de temps, car cela peut être plus lent que de changer de petites parties de l'écran, mais vous pouvez vouloir effacer tout l'écran au début ou à la fin d'un essai. Pour ce faire, vous pouvez utiliser la fonction clear screen.

# 5.8.12 hide/unhide $\clubsuit$



La fonction de hide fait exactement ce que fait la fonction de clear, sauf que de cette façon, vous pouvez aussi à nouveau afficher le stimulus (unhide). En utilisant hide, vous ne changez pas le numéro d'ordre du stimulus. Cela peut parfois être pratique.

#### 5.8.13 rotate

Il est possible de faire pivoter les images présentées avec la commande "show bitmap". Un bon exemple est donné dans l'exemple de Mackworth dans la bibliothèque d'expériences PsyToolkit.

L'angle est indiqué en dixième de degrés. Ainsi un angle de 90 degrés est noté

#### Exemple de rotation d'une image

rotate next 900 show bitmap MyBitmap

#### Exemple de rotation d'une image déjà affichée à l'écran

show bitmap delay 500 rotate 1 900

## 5.8.14 delay #

L'instruction de delay met le programme en pause pendant le nombre de millisecondes spécifié. Vous aurez souvent besoin de cette fonction, par exemple, si vous souhaitez afficher un stimulus pendant un intervalle de temps spécifique, ou pour attendre entre deux essais.

#### syntaxe

delay[millisecondes]

#### Exemple de pause montrant comment afficher un stimulus pendant 200 ms

show bitmap MyBitmap delay 200 clear -1

Le paramètre de delay (dans l'exemple 200) peut être une variable ou une entrée de tableau. Regardez bien les exmples pour savoir comment faire.

## 5.8.15 readkey **\$**



L'instruction readkeys demande à l'ordinateur d'attendre un appui touche au clavier. Le premier argument est la touche correcte corresondant aux touches dans l'instruction keys.

#### syntaxe

readkey[numéro de la touche correcte][temps de réponse maximum]

Dans l'exemple suivant, il y a deux touches possibles dans la tâche (appelée My-Task), la touche a et la touche z. Imaginez qu'un stimulus nécessite d'appuyer sur la touche z, c'est-à-dire la deuxième touche (a étant la première de la ligne "keys"). La commande de readkey sera alors de 3 secondes pour une pression sur la touche. Si les participants appuient sur la touche z, l'ETAT sera réglé sur CORRECT (qui est égal à la valeur numérique 1). Si les participants appuient sur la mauvaise touche, celle-ci sera FAUSSE (valeur numérique 2), et s'il n'y a pas de réponse du tout dans un délai de 3000 ms, le code de STATUT sera TIMEOUT (valeur numérique 3).

#### Exemple avec readkey

task MyTask keys a z show bitmap PressTheZkey readkey 2 3000

Après un appel à readkey, l'utilisateur peut utiliser les variables suivantes :

- RT (avec le temps de réponse en millisecondes, c'est-à-dire l'heure de l'événement d'appui de la touche)
- TT (l'heure à laquelle la touche a été libérée)
- STATUT (avec les valeurs CORRECT, FAUX, ou TIMEOUT, ou 1,2, ou 3)

#### keystatus 贵 5.8.16



La commande keystatus est réservée à un usage avancé et ne fonctionne que sous Linux (c'est-à-dire pas la version web) Sous certaines conditions, vous souhaitez vérifier directement l'état du clavier. Vous pouvez le faire avec la commande

keystatus. Vous pouvez (en option) spécifier quelle touche vous souhaitez que le clavier soit correct.

#### Exemple avec keystatus

```
task checkKey
  keys a 1
  show rectangle 0 0 50 50
                            255 0 0
  timestamp MyTime1
  timestamp MyTime2
  while x < 5000 \# \text{mu} run while-loop for 5 secs
  keystatus ## read the keyboard status
   if KEY == 1 ## key "a"
     unhide 1 ## display the rectangle
   fi
   if KEY == 2 ## key "1"
     hide 1 ## hide the rectangle if 1 is pressed
   set x timestamp-diff MyTime1 MyTime2 # now x should be 500
   nap ## give the computer some rest in the while loop
  while-end
```

## 5.8.17 readkeys $\clubsuit$

La fonction "readkey" est utile pour la plupart des expériences, où il suffit d'appuyer sur une seule touche en réponse à un stimulus. Dans un certain nombre d'expériences, vous pouvez souhaiter que les gens puissent taper un mot (ou une séquence de lettres). La commande readkeys vous permet de le faire. Regardez l'exemple ci-dessous. La commande readkeys comporte de nombreuses options, ce qui peut sembler un peu plus difficile (voir la liste ci-dessous).La commande permet d'effacer un caractère. La touche Retour "termine" la saisie des readkeys. Il y a également un exemple de code dans la leçon (en anglais) sur la mesure des temps de réponses avec readkeys.

#### Exemple avec readkeys

```
bitmaps
   apple ## a picture of an apple
   empty_box ## an empty rectangle used as letter placeholder

fonts
   arial 20

task my_task
   text color yellow
```

```
show bitmap apple
readkeys "apple" 10000
save RT STATUS
```

Une fonction plus complexe utilise les options de readkeys.

Notez que, par défaut :

- les lettres tapées seront affichées à l'écran (vous pouvez les cacher avec l'option readkeys hide)
- il n'y a pas de places réservées (voir l'exemple ci-dessous pour savoir comment afficher les places réservées)
- la couleur de la police est blanche par défaut (vous pouvez la définir différemment avec, par exemple, textcolor yellow) \*la taille maximale du tableau est de 100 (ce qui est en pratique beaucoup plus long que ce dont vous pourriez avoir besoin)
- les lettres seront affichées à partir du centre de l'écran
- la première police définie (dans la section des polices) sera utilisée
- l'espacement par défaut des lettres sera de 25 pixels

#### Options de readkeys :

- readkeys option size number : combien de lettres peuvent être tapées au maximum.
- readkeys option show : les lettres tapées seront affichées à l'écran (à partir du centre de l'écran)
- readkeys option show x\_position y\_position : les lettres tapées seront affichées comme indiqué par la position x/y
- hide : les lettres tapées ne seront pas affichées à l'écran
- readkeys option space number : Les lettres seront espacées par ce nombre de pixels
- readkeys option placeholders image : Il y aura des places réservées sous la forme d'image (telle que définie dans les bitmaps)
- readkeys option placeholders Width Height : Les places réservées seront des rectangles de taille *Largeur* par *Hauteur*

#### Voici un exemple avec certaines options

```
bitmaps
  apple ## a picture of an apple
  empty_box ## an empty rectangle used as letter placeholder

fonts
  my_arial arial 20
```

```
task my_task
  text color yellow
  show bitmap apple
  text color red
  readkeys option size 5
  readkeys option show 100 50
  readkeys option placeholders empty_box
  readkeys font font1
  readkeys "apple" 10000
  save RT STATUS
```

Quelques explications sur l'exemple ci-dessus :

- text color red : La commande readkeys affichera les lettres en rouge
- option readkeys size 5 : 5 lettres au maximum peuvent être tapées
- readkeys option placeholders empty\_box : L'espace réservé sera l'image bitmap empty\_box telle que donnée par l'utilisateur dans la liste des bitmaps
- readkeys option show 100 50 : les lettres tapées sont affichées à l'écran à la position 100 50.
- readkeys option font my arial : La police my arial sera utilisée

Dans l'exemple ci-dessus, la dernière ligne de readkeys attend pendant  $10\,000~\mathrm{ms}$ que les personnes aient entré une ou plusieurs lettres (le participant appuyant sur la touche retour/entrée met fin à l'entrée). Si les personnes ont saisi "apple", leur statut est réglé sur "correct" (c'est-à-dire la valeur 1).

## 5.8.18 textbox #



L'instruction textbox crée une boîte dans laquelle les gens peuvent saisir du texte. Contrairement à l'instruction readkeys, la textbox permet également la saisie de caractères spéciaux (y compris la saisie de texte en chinois, les caractères spéciaux des langues européennes, etc.

Elle a été introduite dans la version 3.4.0 et sera développé ultérieurement. Actuellement, une zone de texte est affichée avec l'instruction textbox new et vous pouvez fournir plusieurs arguments. Voici un exemple dans lequel les arguments suivants sont utilisés: x-position, y-position, width, height, textcolor, background color, font size et font family (comme arial ou times).

```
task my_task
  show text "Enter words" 0 -280
  show rectangle 300 250 100 100 green
  textbox new 0 0 500 400 white blue 30 arial
```

```
readmouse 1 2 99999 range 2 2
set %%my_list textbox
save %%my_list
```

Utilisez set pour enregistrer le texte dans une variable de texte (comme %my\_word) ou une liste de texte (comme %%my\_list).

#### $_{\rm readmouse} \, \P$ 5.8.19



Pour la version sans clic de readmouse, cela a légèrement changé dans la version 3.1.0

Comme pour la touche de lecture, vous pouvez vérifier si le participant a cliqué sur la touche de la souris et, si oui, si la souris était à l'endroit où elle devait être. Vous pouvez aussi simplement attendre que la souris soit déplacée dans une certaine zone de l'écran. En fait, vous pouvez vérifier si l'emplacement d'un stimulus spécifique a été touché par la souris (bitmap, ou rectangle, ou texte).

Voici les différentes options

- Attendre que la souris soit cliquée et que la souris soit au bon endroit
- Attendre que le participant déplace la souris au bon endroit

Dans l'exemple suivant, nous montrons deux rectangles, et nous voulons que le participant passe la souris sur le rectangle vert (le premier bitmap), qui est positionné à gauche (-200). Nous donnons au maximum 5 secondes, c'est-à-dire 5000 millisecondes.

#### Exemple de readmouse sans cliquer

```
task checkMouse
  show rectangle -200 0 40 40 0 255 0 # green rect, left
  show rectangle 200 0 40 40 255 0 0 # red rect, right
  readmouse 1 5000
  save STATUS RT
```

Dans l'exemple ci-dessus, vous obtenez un STATUT correct si le rectangle de gauche a é

#### Exemple de readmouse sans cliquer et utilisant range

```
task checkMouse
  show rectangle 0 0 700 500
                               100 100 100
  show rectangle -200 0 40 40 0 255 0 # green rect, left
  show rectangle 200 0 40 40 255 0 0 # red rect, right
 readmouse 2 5000 range 2 3
  save STATUS RT
```

Dans l'exemple ci-dessus, le premier rectangle est un rectangle gris qui n'est qu'un fond. Comme ce rectangle est presque aussi grand que l'ensemble de l'écran d'expérimentation, nous devons dire à PsyToolkit que nous ne nous soucions pas du survol de ce stimulus par la souris. Pour ce faire, nous utilisons range dans l'instruction de readmouse. Cela signifie, dans ce cas, qu'il ne faut vérifier que si le stimulus 2 ou le stimulus 3 a été survolé.

Dans l'exemple suivant, nous avons exactement la même tâche, mais nous voulons un clic de souris dans le premier stimulus. Nous ajoutons l'argument "l" représentant le bouton gauche de la souris. En Javascript, il est recommandé de n'utiliser que le bouton gauche de la souris, car le bouton droit peut afficher un menu contextuel dans le navigateur , ce que vous ne souhaitez évidemment pas.

#### Exemple avec readmouse

```
task checkMouse
show rectangle -200 0 40 40 0 255 0 # green rect, left
show rectangle 200 0 40 40 255 0 0 # red rect, right
readmouse 1 1 5000
save STATUS RT
```

Un problème se pose lorsque les stimuli se chevauchent. Imaginez la situation suivante. Vous avez un grand rectangle qui est juste leur pour montrer un rectangle jaune clair, et au dessus de cela, un rectangle beaucoup plus petit. Si vous voulez que les gens cliquent sur ce second rectangle plus petit, vous devez dire à l'ordinateur que vous ne vous souciez pas du premier. La façon de le faire est de spécifier la gamme de bitmaps qui vous intéresse. Voir l'exemple ci-dessous :

#### Exemple avec readmouse

```
task checkMouse show rectangle -200 0 400 400 255 255 0 # big yellow rectangle show rectangle 50 0 40 40 255 0 0 # small red one readmouse 1 2 5000 range 2 2 # wait for second one being clicked save STATUS RT
```

Enfin, il existe un cas spécial de survol, où vous pouvez demander à readmouse de ne répondre que si la souris survole la cible demandée, et de ne pas répondre par erreur pour tout autre stimulus. Pour cela, il vous suffit d'ajouter l'option find. Exemple de readmouse sans cliquer et utilisant find

```
task checkMouse
show rectangle -200 0 40 40 0 255 0 # green rect, left
show rectangle 200 0 40 40 255 0 0 # red rect, right
readmouse 1 5000 find
save STATUS RT
```

Dans le cas ci-dessus, vous pouvez survoler le rectangle de droite, mais Psy-Toolkit l'ignorera complètement et ne considérera que le stimulus de gauche. Cela peut être utile dans certaines situations. Par exemple, si vous voulez qu'un participant trouve un objet parmi d'autres. Le participant peut passer le curseur sur tous les objets jusqu'à ce quil ait vu l'objet cible. Vous pouvez bien sûr aussi le faire avec une range, mais cette solution est un raccourci car ici, vous n'avez pas besoin de connaître exactement les numéros du paramètre range, ce qui simplifie les choses.

### 5.8.20 drag #

La fonction *drag* vous permet de faire glisser un stimulus d'un endroit à un autre. Il existe de nombreuses options pour ce faire. Souvent, il est plus facile de regarder d'abord un exemple simple. Le mieux est de lire la leçon sur drag pour des exemples détaillés.

Vous trouverez ci-dessous des exemples simples, mais d'abord un aperçu. Notez que les valeurs en italique doivent être remplacées par les numéros des variables. Les valeurs entre crochets sont des options différentes. Plus de détails plus bas.

#### Aperçu de drag

```
drag option from [*&&array* or *sequence*]
drag option to [*&&array* or *sequence*]
drag option snap ([left|center|right][top|center|bottom][no_overlap *&&array* or *sequence*]
drag option *xlim value value*
drag option *ylim value value*
drag *maximum_duration*
```

Dans sa forme la plus simple possible, vous pouvez avoir un exemple comme celui-ci. Notez que vous devrez toujours fournir des lignes *drag option* pour indiquer à PsyToolkit quel stimulus doit être déplacé.

Dans cet exemple, vous montrez un stimulus (stimulus numéro 1) en jaune (le jaune est RGB : 255 255 0). Vous dites à PsyToolkit que le stimulus qui peut être glissé de quelque part est dans ce cas le stimulus numéro 1. Vous donnez jusqu'à 60 000 millisecondes (c'est-à-dire 60 secondes) pour le déplacement.

#### Exemple super simple et fonctionnel de déplacement

```
task my_drag
show rectangle 0 0 100 100 255 255 0
drag option from 1
drag 99999
```

block test

```
tasklist
   my_drag 1
end
```

Il y a d'autres ptions. Par exemple, avec PsyToolkit vous pouvez avoir plusieurs de stimuli qui peuvent être déplacés. Dans \*drag option from" vous pouvez spécifier plusieurs stimuli ou même une liste, comme dans l'exemple ci-dessous.

Vous pouvez faire glisser le rouge et le vert, mais pas le rectangle blanc

```
task my_drag
  show rectangle -100 0 100 100
                                  255 0 0
                                               # red
  show rectangle
                   0 0 100 100
                                 0 255 0
                                               # green
  show rectangle 100 0 100 100
                                 255 255 255 # white
  drag option from 1 2
  drag 99999
block test
  tasklist
     my_drag 1
  end
```

L'instruction *drag* permet de spécifier la ou les valeurs d'un ou de plusieurs stimuli. Dans sa forme la plus simple, vous pouvez avoir un exemple tel que celui-ci :

Faites glisser le rouge sur le rectangle vert

```
task my_drag
  show rectangle 100 0 120 120
                                    0 255 0
                                                # green
  show rectangle -100 0 100 100
                                    255 0 0
                                                  # red
  drag option from 2
  drag option to 1
  drag 99999
  show text "Now wait 1000 ms" 0 150
  delay 1000
block test
  tasklist
     my_drag 1
  end
```

Dans l'exemple ci-dessus, vous auriez pu ajouter *drag option snap* pour que le rectangle rouge s'accroche exactement sur le rectangle vert. Vous pouvez même

spécifier la manière dont vous souhaitez que l'accrochage se fasse, voici quelques exemples:

exemples d'accrochage par glisser-déposer

```
drag option snap # par défaut, l'accroche se fait au centre du rectangle
drag option snap center center # même chose, mais en indiquant explicitement l'aligneme
drag option snap left center # cet alignement est horizontal à gauche mais vertical au
```

Plus d'informations sur les variables liées à l'instruction ici.

### 5.8.21 dragging #



La fonction dragging permet à votre participant de continuer à faire glisser les stimuli sur l'écran jusqu'à ce qu'il ait terminé. Lorsqu'il a terminé (mais avant le temps de réponse maximal), il doit appuyer sur le bouton de sortie. Pour cela, vous pouvez spécifier un stimulus spécifique.

Toutes les options de drag peuvent également être utilisées avec dragging, mais il existe une option supplémentaire.

#### Aperçu de dragging

```
drag option exit *stimulus*
dragging *maximum_duration*
```

Dans l'exemple ci-dessous, deux rectangles rouges et deux rectangles verts sont représentés. Un cinquième rectangle jaune (sur lequel figure le texte "Exit") fait office de bouton de sortie. Vous devez faire glisser les rectangles rouges sur les rectangles verts.

Notez que si vous ne les faites pas glisser sur les cibles, ils reviennent à leur position initiale.

#### Exemple super simple et fonctionnel de dragging

```
task my_dragging
                                                  # green
  show rectangle 100 -100 120 120
                                       0 255 0
  show rectangle 100 100 120 120
                                       0 255 0
                                                  # green
  show rectangle -100 -100 100 100
                                       255 0 0
                                                  # red
  show rectangle -100 100 100 100
                                       255 0 0
                                                  # red
  show rectangle 250 0 100 80
                                255 255 0
  show text "Exit" 250 0
                           0 0 0
 drag option from 3 4
 drag option to 1 2
  drag option snap center center
```

```
drag option exit 5
  dragging 60000
  save RT
block test
  tasklist
    my_dragging 1
  end
```

Plus d'informations sur les variables liées l'instruction dragging ici.

### variables utilisées avec drag et dragging



Les variables suivantes peuvent être utilisées pour accéder aux informations relatives aux actions de glisser-déposer. Vous n'en avez pas nécessairement besoin.

#### Vue d'ensemble

```
DRAGGED the stimulus that was being dragged
DRAGGED_ON the stimulus that it was move on (only with "drag option to")
DRAG CHANGED if a stimulus had been moved at all (1 if changed, 0 if not)
DRAG OLD X the stimulus X coordinate when drag started
DRAG_OLD_Y the stimulus Y coordinate when drag started
DRAG_X the stimulus X coordinate of (last) dragged stimulus
DRAG_Y the stimulus Y coordinate of (last) dragged stimulus
```

**DRAGGED** contains the number of the stimulus a participant dragged. This is interesting when a participant can choose to drag one out of multiple stimuli.

DRAGGED\_ON contient le numéro du stimulus sur lequel le participant a glissé.

**DRAGGED\_CHANGED** Contient 1 si le participant a traîné un stimulus, et 0 si rien n'a été traîné.

DRAG\_OLD\_X Si un stimulus a été traîné, sa position X précédente avant le début du traînage.

DRAG\_OLD\_Y Si un stimulus a été déplacé, sa position Y précédente avant le début du déplacement.

DRAG\_X Si un stimulus a été traîné, sa position X actuelle.

DRAG\_Y Si un stimulus a été traîné, sa position actuelle en Y.

Les expériences utilisant drag ou dragging peuvent être assez complexes. Apprenez à les écrire dans la section leçons.

## 5.8.23 progress $\bigoplus$

PsyToolkit peut afficher une barre de progression. Cela peut être utile si vous voulez dire aux gens combien d'essais ont été effectués et combien de temps il faudra pour les terminer.

Les barres de progression peuvent distraire. Si vous faites une expérience sur l'attention visuelle ou la mémoire, vous devez vous demander si vous pensez que le fait de regarder la barre de progression peut interférer avec le processus cognitif que vous étudiez. En fait, il est très facile de le montrer. Il existe un certain nombre d'options pour colorer et former la barre de progression comme vous le souhaitez, mais la valeur par défaut est une simple barre de progression horizontale étroite en haut de l'écran.

La commande est simple : show progressbar done todo

Le premier chiffre correspond aux essais réalisés jusqu'à présent et le second au nombre total d'essais que le participant devra réaliser.

Dans l'exemple ci-dessous, la commande show progress affiche la barre de progression. Elle est suivie par TRIALCOUNT (essais effectués jusqu'à présent) et le nombre total d'essais (ici par exemple 30).

#### Exemple de barre de progression

```
task exampletask
show progress TRIALCOUNT 30
delay 1000
show mystimulus
```

Il existe un ensemble d'options sécifiques que vous pouvez définir dans le cadre de votre tâche (et non dans la section des options). Vous trouverez ci-dessous un exemple dans lequel vous pouvez définir les deux couleurs (une pour la couleur des essais effectués et une pour la couleur de fond, par défaut ce sont le vert et le gris, respectivement).

#### Exemple de barre de progression avec options

```
task exampletask

progress option size 0 290 600 10

progress option color1 255 255 0

progress option color2 128 128 128

progress option between 5

show progress TRIALCOUNT 30

delay 1000

show mystimulus
```

Ces options ont la signification suivante.

- size : position X (milieu de la barre de progression), position Y, largeur,
- color1 : La couleur des essais réalisés (en format Rouge/Vert/Bleu)
- color2 : La couleur des essais réalisés (en format Rouge/Vert/Bleu)
- between: L'espace entre les essais (par exemple, choisir 2 pour un petit écart entre chaque point d'avancement)

Notez qu'au lieu de color1 et de color2, vous pouvez aussi simplement écrire :

#### Exemple de barre de progression avec options

```
progress option colors 255 255 0
                                    128 128 128
```

# 5.8.24 choose #



Dans certaines circonstances, vous souhaitez demander au participant de cliquer sur plusieurs stimuli à l'écran. Idéalement, vous voudriez permettre au participant de sélectionner des objets et de les désélectionner également. Tout cela est possible avec la fonction "choisir".

En bref, la fonction choose permet au participant de cliquer sur une série de stimuli, d'afficher un symbole au-dessus des stimuli, et lorsqu'il clique à nouveau, le symbole disparaît.

Il est plus facile d'apprendre comment cela fonctionne à partir d'un exemple.

Dans l'exemple suivant, il y a trois symboles affichés à l'écran qui peuvent être sélectionnés avec un sélecteur bitmap. Les options de cette commande peuvent être définies avant que la commande choose ne soit appelée avec l'option choose. Le participant dispose de 60 secondes (60 000 millisecondes).

#### Exemple de choix

```
bitmaps
  markingsymbol
  house
  ball
  car
  exitsymbol1 ## shown when at least 2 selected
  exitsymbol2 ## shown when not enough are selected
task clickMysymbols
  show bitmap house # bitmap 1
  show bitmap ball # bitmap 2
  show bitmap car
                    # bitmap 3
  choose option select markingsymbol
```

```
choose option minselect 2
choose option exit exitsymbol1 exitsymbol2 350 250
choose 60000 1 3
save RT CHOSEN_N CHOSEN_1 CHOSEN_2 CHOSEN_3
```

Notez les éléments suivants à propos de l'exemple ci-dessus :

- Il y a trois images sur l'écran.
- Si vous cliquez sur une image, l'image "markingsymbol" sera placée sur celle-ci.
- Si vous cliquez à nouveau, ce "markingsymbol" sera supprimé
- Le RT est le temps de réponse jusqu'à ce que le symbole de sortie ait été cliqué
- CHOSEN\_N contient le nombre d'objets sélectionnés
- CHOSEN\_1, CHOSEN\_2, CHOSEN\_3, etc. contient le numéro de l'objet. Ils seront toujours dans l'ordre numérique.

Il existe plusieurs autres options:

- choose option minselect : Le nombre minimum d'objets à sélectionner
- choose option maxselect : Les objets maxumum à sélectionner
- choose option sprites : choisir parmi les sprites au lieu des stimuli statiques
- choose option keep: Garde les images du sélecteur sélectionné par le participant à l'écran (par défaut, elles sont effacées immédiatement lorsque la fonction de choix est terminée)

La commande choose est particulièrement utile pour les tâches de mémoire.

## 5.8.25 rate #

Parfois, les gens veulent une échelle de Likert dans leur expérience. Par exemple, vous pouvez montrer un article de shopping et demander aux gens quelle est la probabilité que vous l'achetiez la semaine suivante sur une échelle de un à cinq. C'est maintenant super facile à faire.

#### Exemple super simple d'utilisation de rate

```
bitmaps
  something_for_sale

task wantit
  show bitmap something_for_sale -200 0
  rate 10000 5
  save RATE RATE_RT RATE_STATUS
```

Cela vous donne le type d'échelle de Likert le plus élémentaire. Par défaut, les points de l'échelle de Likert sont des cercles jaunes et à gauche et à droite, vous voyez une petite boîte blanche. En réalité, vous voulez beaucoup plus. Pour cela, vous pouvez utiliser des options. Tout d'abord, vous pouvez sélectionner un lieu avec l'option *pos* suivie de coordonnées x et y.

Dans l'exemple ci-dessous, l'échelle est maintenant indiquée à gauche du centre.

Exemple simple de rate

```
bitmaps
   something_for_sale

task wantit
   show bitmap something_for_sale -200 0
   rate option pos -200 0
   rate 10000 5
   save RATE RATE_RT RATE_STATUS
```

Dans l'exemple ci-dessous, nous ajoutons de plus belles images (bitmaps) au lieu des symboles par défaut. Dans PsyToolkit, il y a l'option "labels" pour rate, de sorte que vous pouvez faire une image avec les textes "pas du tout" et "beaucoup" ou juste de simples flèches comme dans l'exemple ci-dessous. L'option items spécifie simplement l'image des points cliquables sur l'échelle.

De plus, dans l'exemple ci-dessous, nous avons 7 éléments et les personnes doivent faire un choix dans les 5 secondes.

#### Exemple simple de rate

```
bitmaps
something_for_sale
left_arrow
right_arrow
my_rectangle

task wantit
show bitmap something_for_sale -200 0
rate option pos -200 0
rate option labels left_arrow right_arrow
rate option items my_rectangle
rate 5000 7
save RATE RATE_RT RATE_STATUS
```

## 5.8.26 save #

Sauvegarde des variables. En général, une tâche se termine par une ligne de sauvegarde.

L'instruction de sauvegarde est d'une importance capitale, car elle garantit que les informations dont vous avez besoin pour votre analyse des données sont stockées. Par défaut, PsyToolkit ne conserve aucune information (contrairement à certains autres logiciels d'expérimentation). L'utilisateur doit indiquer au PsyToolkit quelles informations sont stockées dans le fichier de données. En général, vous voudrez enregistrer l'état actuel, le bloc actuel (si vous avez plus d'un bloc), et au moins le temps de réponse (RT), et si le participant a répondu correctement ou non (STATUS). Les exemples présentés sur ce site web peuvent vous aider à comprendre cela. Le meilleur endroit pour la commande de sauvegarde est à la fin de la description de votre tâche.

#### **Syntaxe**

save[liste de variables]

#### Exemple pour save

save BLOCKNAME RT STATUS

# 5.8.27 sound $\bigoplus$



Utilisé pour jouer des sons. Un son commence juste à jouer, le code continue. Si vous ne voulez rien faire pendant le son, vous devez le faire suivre d'un délai. Vous pouvez également arrêter le son à tout moment en utilisant l'instruction silence.

#### **Syntaxe**

sound [sound name (as defined in 'sounds']

#### Exemple pour sound

sound MySoundFile delay 200 silence MySoundFile

## 5.8.28 if #

Vous pouvez utiliser *if* pour exécuter seulement quelques commandes. Cela est souvent nécessaire pour afficher un feedback si les gens font une erreur. Le contraire de if est fi. fi sert à clore la liste de commandes conditionnelles

#### exemple avec if

```
task MyTask
  show bitmap stimulus
  readkey 1 1000
  if STATUS == CORRECT
    show bitmap WellDone
    delay 1000
    clear -1
  fi
  if STATUS != CORRECT
    show bitmap Mistake
    delay 2000
    clear -1
  fi
  if $x == &y
    show bitmap SmileyFace $x $y
  fi
```

#### Exemples de if avec else

```
task MyTask
  show bitmap stimulus
  readkey 1 1000
  if STATUS == CORRECT
    show bitmap WellDone
    delay 1000
    clear -1
  else
    show bitmap Mistake
    delay 2000
    clear -1
  fi
  if $x == &y
    show bitmap SmileyFace $x $y
  fi
```

L'instruction else doit être sur sa propre ligne. Avec if, vous pouvez également comparer deux listes

Exemples de if avec des listes

```
task MyTask
  set &a 1 2 3 4
 set &b 1 2 3 4
  if &&a == &&b
    show text "Arrays a and b are exactly the same"
    show text "Arrays a and b are different"
  fi
```

#### while # 5.8.29



Comme pour if, vous pouvez mettre le code dans une boucle while. La boucle while se termine par la déclaration "while-end".

Tout ce qui se trouve entre while et while-end sera exécuté jusqu'à ce que la condition de la ligne while soit satisfaite.

#### Exemple de boucle while

```
task MyTask
  set $mycounter 0
 while $mycounter < 10
    set $x random -200 200
    set $y random -200 200
    show bitmap SmileyFace $x $y
    set $mycounter increase
  while-end
```

Si vous faites une erreur, l'ordinateur peut être bloqué dans la boucle while. Par exemple, si vous n'augmentez pas la valeur de \$mycounter dans l'exemple ci-dessus, le code sera bloqué et continuera à dessiner de nouveaux Smileys sur des localisations aléatoires.

#### Instructions moins souvent utilisées {#s5-9} 5.9



Certaines fonctions sont rarement utilisées ou nécessaires, mais elles ajoutent au potentiel des scripts du PsyToolkit.

# 5.9.1 font #

Vous pouvez définir la police de l'instruction show text.

#### Exemple d'usage de font

```
fonts
  MySmallFont arial.ttf 20
  MyBigFont arial.ttf 50

task Mytask
  font MySmallFont
  show text "small" 0 0
  font MyBigFont
  show text "small" 0 100
```

#### 5.10 Fonctions avancées

## 5.10.1 end $\blacksquare$

Vous pouvez terminer la tâche à n'importe quel moment de votre liste de d'actions dans une tâche. Vous pouvez même mettre fin à la liste de tâches en cours ou à l'ensemble de l'expérience. Il vous suffit de d'ajouter end à la tâche, à la liste de tâches ou à l'expérience. Cela peut être utile, par exemple, lorsque les gens sont en formation et que vous voulez arrêter le blocage des procès lorsqu'ils ont fait une erreur. Il s'agit toutefois d'une déclaration dont vous aurez rarement besoin.

## 5.10.2 timestamp

L'horodatage (timestamp) peut être utilisé pour saisir l'heure actuelle. Cela peut être pratique si vous vérifiez l'heure passée dans une boucle while. Cela peut également être pratique si vous essayez de déboguer du code et de vérifier si le timing est conforme à ce qui est attendu.

Vous pouvez obtenir le timestamp-diff entre deux variables de timestamp en utilisant la commande set comme dans l'exemple ci-dessous. Les horodatages ne sont pas directement accessibles ou utilisables autrement que par la commande set timestamp-diff

#### Exemples de timestamp

task MyTask

```
timestamp MyFirstTimestamp
delay 1000
timestamp MySecondTimestamp
set $x timestamp-diff MyFirstTimestamp MySecondTimestamp
save $x
```

## 5.10.3 sprite #

Dans PsyToolkit, les sprites sont un type de stimulus distinct (en plus de l'image bitmap, du rectangle, du cercle et du texte. Les sprites peuvent se déplacer et tourner. C'est idéal pour les expériences telles que le suivi d'objets multiples. Ils ont leurs propres commandes.

#### Exemple basique montrant un sprite en mouvement

```
bitmaps
mystimulus

task MyTask
sprite create mystimulus
sprite 1 display
sprite move direction 45 5
delay 2000

block test
tasklust
MyTask 1
end
```

La vitesse des sprites en mouvement est basée sur différents facteurs. La vitesse dépend de la fréquence de mise à jour de l'écran. Cela signifie simplement que si vous utilisez des expériences sur différents systèmes, vous pourriez avoir des vitesses des sprites en mouvement légèrement différentes. Consultez les exemples du package Linux pour voir comment les commandes ci-dessous fonctionnent.

Nom de la fonction	Ce que cela fait	Exemple
	Ce que ceia iait	Exemple
**Créer un sprite* *		
sprite create [bitmap]	Crée un stimulus	sprite create
	au centre l'écran	MyStimulus
sprite create [bitmap]	Crée un stimulus et	sprite create
[x][y]	le positionne	MyStimulus 100 50
sprite create [bitmap]	Crée un stimulus,	sprite create
[x][y][angle][speed]	le positionne et le	MyStimulus 100 50
, ,	bouge	$45^{\circ}5$

Nom de la fonction	Ce que cela fait	Exemple
Afficher les sprites		
sprite [sprite] display	Rend le sprite 1 visible	sprite 1 display
sprite [sprite] hide	Cache le sprite 1	sprite 1 hide
sprite [sprite] freeze	Stoppe le mouvement	sprite 1 freeze
sprite [sprite] move	remet en mouvement	sprite 1 move
déplacer les sprites		
<pre>sprite [sprite] jump (xpos)(ypos)</pre>	Change de position une fois	sprite 1 jump 200 100
sprite [sprite] move	Définit une	sprite 1 move to
(to/towards)	direction	200 100 5
(xpos)(ypos)(speed)	an 0001011	_50 100 0
sprite [sprite] move	Bouge dans une	sprite 1 move
direction (angle)(speed)	direction	direction -90 4
sprite [sprite] move path	Bouge le long d'un	sprite 1 move path
(speed) x1 y1 xn yn	ensemble de	3 200 100 -200 100
J J	localisations	300 50
sprite [sprite] evade	Les sprites ne se	sprite 1 evade
1 -1 -	croiseront pas	1
sprite [sprite] borders	Le sprite se	sprite 1 borders
<pre>[left][right][top][bottom]</pre>	heurtera à ces	-300 300 -200 200
9	frontières fixes	
rotation/défilement des		
sprites		
sprite [sprite] rotate	Faire tourner un	sprite 1 rotate -7
[speed in degrees per	sprite en continu	
refresh rate]		
sprite [sprite] rotate to	Faire tourner un	sprite 1 rotate to
[angle] [speed in degrees	sprite	270
per refresh rate]	progressivement	
sprite [sprite] rotate now	Faire tourner un	sprite 1 rotate now
[angle]	sprite	45
sprite [sprite] hscroll	Faites défiler un	sprite 1 hscroll -3
[speed]	sprite	
Changer les sprites		
sprite [sprite] speed	vitesse fixée	sprite 1 speed 2
(speed)		
sprite [sprite] accelerate	ajuste la vitesse	sprite 1 accelerate 3
(speed change)	1 11	
<pre>sprite [sprite] bitmap [bitmapname]</pre>	changer l'apparence	sprite 1 bitmap smiley

Nom de la fonction	Ce que cela fait	Exemple
commandes sur tous les sprites à la fois sprites [options] all	agir sur tous les sprites en une seule ligne	sprites display all

#### 5.11Instructions pour les équipements spéciaux

Les équipements spéciaux sont principalement les cartes d'entrée/sortie et les claviers spéciaux. Je recommande les claviers Cedrus. Si vous le souhaitez, vous pouvez construire votre propre clavier. Cela permet d'économiser beaucoup d'argent. Vous pouvez le brancher sur le port parallèle (si vous en avez un). Ce clavier est appelé "ultra" (conçu par Felix Frey, Université de Leipzig), et un document expliquant comment le construire est disponible gratuitement.

#### 5.11.1cedrus readkey

Attendre une touche spécifique de votre clavier USB Cedrus. Assurez-vous que vous savez quelle touche a quelle valeur numérique (vous pourriez vouloir faire quelques expériences à ce sujet, et il existe un exemple de programmation qui vous montre le numéro de chaque touche). Sous Linux, vous pouvez appeler cela en utilisant la commande "testcedrus", qui est fournie avec PsyToolkit.

#### table # 5.12



Les tables ont des lignes et des colonnes. Chaque fois qu'une table est utilisée dans une tâche, une de ses lignes est choisie. Chaque colonne peut être référencée en utilisant le signe @. Par exemple, ? fait référence à la deuxième colonne de la ligne qui est choisie pour un essai de tâche donné. Dans le bloc, l'utilisateur peut spécifier comment les lignes du tableau sont sélectionnées. La sélection par défaut est aléatoire, mais il existe d'autres moyens (par exemple : ordre fixe, répétition sur un essai).

#### Exemple de table

table MyTable 10 2 bitmap1 -10 1 bitmap2 5.13. BLOCK 103

#### Exemple de table avec des chaines de caractères

```
table MyTable
  10 2 bitmap1 "condition one"
-10 1 bitmap2 "condition two"
```

Dans une tâche qui utilise un tableau (avec l'instruction table), chaque colonne peut être référencée avec le signe @. Ainsi, @2 renvoie à la deuxième colonne de la ligne du tableau qui a été sélectionnée.

Chaque fois qu'une tâche est exécutée, une seule ligne du tableau associé est sélectionnée. Par défaut, une ligne est choisie au hasard. Il existe d'autres façons de choisir des lignes de tableau (en utilisant l'instruction tasklist dans les blocs).

Dans chaque essai de tâche avec un tableau, vous avez non seulement accès aux colonnes, mais vous pouvez également obtenir le numéro de la ligne de la table. La variable est appelée TABLEROW. Cela peut être pratique pour sauvegarder des données. S'il y a beaucoup de données importantes dans chaque tableau que vous voulez les utiliser dans votre analyse, vous pouvez ainsi vous assurer quel ligne a été utilisée.

### 5.13 block **#**

Dans les expériences, les essais se font par blocs. Un bloc d'essais signifie que le participant effectue, par exemple, 100 essais de la tâche Stroop. Vous pouvez aussi faire une pause au milieu, et créer deux blocs de 50 essais à la place. Les blocs appellent des tâches, donc vous devez vraiment avoir au moins un bloc dans votre script PsyToolkit. Les blocs peuvent être complexes, mais ils peuvent aussi être très simples, comme cet exemple ci-dessous.

#### Exemple de bloc simple, appelant la tâche stroop 100 fois

```
block MyBlock
tasklist
strooptask 100
end
```

### 5.13.1 bitmap 📅

L'instruction bitmap est simple. Vous voulez souvent afficher une image avec une instruction et attendre qu'on appuie sur une touche.

#### Example de bitmap dans un bloc

```
block Myblock
  bitmap MyInstruction
  wait_for_key
  tasklist
    strooptask 100
  end
```

L'affichage d'un bitmap dans une tâche et dans un bloc sont différents. Ici, dans les blocs, il ne faut pas mettre le mot "show" avant le bitmap. Dans les tâches, vous devez le faire. Il y a de bonnes raisons à cette distinction (dans les tâches, les stimuli sont rapides, alors que dans les blocs, ils sont considérés comme des instructions).

## 5.13.2 sound $\clubsuit$



Comme dans la tâche, vous pouvez démarrer un son avec sound et l'arrêter avec silence

Exemple avec sound dans un bloc

```
block Myblock
  sound welcometune
  bitmap MyInstruction
  wait_for_key
  silence welcometune
  tasklist
    strooptask 100
```

## 5.13.3 delay #



Comme pour les tâches, vous pouvez introduire une pause delay. Parfois, c'est bien pour les comptes à rebours.

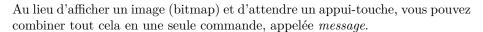
#### Exemple avec delay dans un bloc

```
block Myblock
  bitmap number3
  delay 500
  bitmap number2
  delay 500
  bitmap number1
  delay 500
```

5.13. BLOCK 105

```
tasklist
mytask 100
end
```

# 5.13.4 message **\$\Pi\$**



Exemple de message dans un bloc (en attente de la barre d'espacement, qui est la touche par défaut)

```
block Myblock
message MyInstruction
tasklist
strooptask 100
end
message ThankYouBitmap
```

Exemple de message en bloc (exemple en attente de la touche "b")

```
block Myblock
message MyInstruction b
tasklist
strooptask 100
end
message ThankYouBitmap
```

Vous pouvez utiliser la touche de la souris au lieu du clavier. Veillez à ce que cela soit clairement indiqué dans le message. Le participant poursuivra ensuite en cliquant (ou en touchant) n'importe où sur l'image bitmap de l'instruction.

Exemple de message en bloc (exemple d'attente d'une pression sur le bouton de la souris au lieu du clavier)

```
block Myblock
message MyInstruction mouse
tasklist
strooptask 100
end
message ThankYouBitmap
```

## 5.13.5 pager #

Parfois, vous voulez une série d'instructions et vous laissez l'utilisateur les parcourir. En d'autres termes, vous avez une série d'images (ou "pages", d'où le mot "pager") que vous voulez montrer les unes après les autres. La commande pager vous permet de faire exactement cela.

- L'utilisateur peut utiliser les touches fléchées et utiliser la touche Q pour s'en sortir. La barre d'espacement permet de passer à l'image/page suivante. Assurez-vous que toutes ces images se trouvent dans la section bitmap.
- Vous pouvez également utiliser la souris au lieu du clavier (cidessous). Veillez à indiquer dans les instructions la façon dont les utilisateurs naviguent dans l'image et comment ils en sortent.

#### Exemple de pager dans un bloc

```
block Myblock
  pager MyInstruction1 MyInstruction2 MyInstruction3 MyInstruction4
  tasklist
    strooptask 100
  end
```

Vous pouvez également utiliser la souris (ou le toucher) pour faire défiler les pages d'instructions. Pour cela, vous devez donner quelques informations supplémentaires. Si vous êtes débutant, essayez d'abord les pagers de base comme expliqué ci-dessus.

Le pager souris/tactile est parfois plus utile (par exemple, les expériences sur l'écran tactile). Dans l'exemple ci-dessous, nous utilisons l'option "pager" pour dire à l'ordinateur ce qui suit :

utilisez la souris avec pager L'image nommée "back" devra être présentée à la position -200 200 L'image nommée "next" devra être présentée à la position 0,200 L'image nommée "start" devra être présentée à la position 200 200

Ainsi, outre les instructions, chaque écran comporte 3 images, une pour revenir en arrière, une pour avancer et une pour quitter les instructions. Cette dernière n'est affichée qu'une fois que le participant les a toutes parcourues. Notez que pour cette expérience, vous devez disposer de bitmaps pour les images : back, next, start, MonInstruction1, MonInstruction2, MonInstruction3.

#### Exemple avec pager utilisant la souris dans un bloc

```
block Myblock
  pager option mouse back -200 200 next 0 200 start 200 200
```

5.13. BLOCK 107

```
pager MyInstruction1 MyInstruction2 MyInstruction3
tasklist
  strooptask 100
end
```

# 5.13.6 Paramétrer les variables #



Parfois, vous souhaitez définir une variable globale. Vous pouvez le faire tout comme vous le feriez dans les tâches. Par exemple, vous pouvez fixer le temps de réponse maximum à une valeur plus élevée pendant la formation.

#### Exemple de paramétrage d'une variable dans un bloc

```
block MyTraining
  pager MyInstruction1 MyInstruction2 MyInstruction3 MyInstruction4
  set &MaxResponseTime 2000
  tasklist
    strooptask 20
  end
block MyRealDatablock
  set &MaxResponseTime 1000
  pager MyInstruction1 MyInstruction2 MyInstruction3 MyInstruction4
  tasklist
    strooptask 100
  end
```

## 5.13.7 tasklist #



Chaque bloc comporte une liste de tâches. Celle-ci décrit les tâches qui sont appelées, combien de fois et dans quel ordre. Vous pouvez avoir une ou plusieurs tâches. Il est rare d'avoir plusieurs tâches, mais cela peut être le cas dans les paradigmes de commutation de tâches. Sur chaque ligne de la liste des tâches, vous devez au moins indiquer le nom de la tâche et le nombre d'essais.

#### Exemple simple de tasklist

```
block MyTraining
  tasklist
    strooptask 20
  end
```

Vous pouvez spécifier que vous voulez que chaque fois que les gens font une erreur, le même processus doit être répété, en utilisant repeat\_on\_error. Si vous utilisez cette méthode, vous devez préciser ce qui est considéré comme une erreur en utilisant la déclaration d'erreur (voir l'exemple ci-dessous).

Repeat\_on\_error ne fonctionne que lorsque vous utilisez des tableaux, les variables aléatoires ne sont pas fixées aux mêmes valeurs.

#### Exemple simple de tasklist

```
table MyTable
  "condition1" 255 0 0 1 ## respond with "r" to red rectangle
  "condition2" 0 255 0 2 ## respond with "g" to green rectangle
task Mytask
  table MyTable
 keys r g
  show rectangle 0 0 50 50
                             @2 @3 @4
 readkey @5 2000
  if STATUS == WRONG
    error ## this makes this trial an error for the repeat_on_error process
  fi
  save @1 STATUS RT
block MyTraining
  tasklist
   Mytask 10 repeat_on_error
  end
```

Dans l'exemple ci-dessus, vous voyez un rectangle rouge ou vert (présenté à la position 0,0, qui est le centre de l'écran, et 50px de large et 50px de haut). Vous devez répondre avec la touche "r" ou "g". Vous vous demandez peut-être pourquoi vous devez spécifier explicitement ce qu'est une erreur si vous utilisez repeat\_on\_error. C'est une bonne question, car dans l'exemple ci-dessus, c'est effectivement une erreur si la personne répond avec la mauvaise touche. Mais il y a des expériences dans lesquelles vous voulez en fait que les gens ne répondent pas (c'est-à-dire que la valeur de STATUT est TIMEOUT). Et dans certaines expériences, il y a plus d'une réponse par essai. repeat\_on\_error vous donne un contrôle total sur ce qu'est une erreur et ce qu'elle n'est pas. Dans certaines expériences, vous pouvez souhaiter que tous les participants effectuent les essais exactement dans le même ordre. Vous pouvez simplement parcourir chaque ligne de votre tableau des tâches. Pour cela, vous utilisez l'option fixed.

#### Exemple simple de tasklist

```
block MyTraining
  tasklist
    strooptask 20 fixed
  end
```

5.13. BLOCK 109

Dans certaines expériences, vous pouvez souhaiter que les participants fassent au moins un certain nombre d'essais corrects. Pour cela, vous utilisez l'option *correct*. Dans l'exemple ci-dessous, le participant doit avoir fait 10 essais corrects, mais après 100 essais, il s'arrêtera, quel que soit le nombre d'essais corrects. Dans la variante *allcorrect*, les essais doivent être consécutivement corrects.

### Exemple simple de tasklist

```
block MyTraining
tasklist
strooptask 10 correct 100
end

block MyTraining
tasklist
strooptask 10 allcorrect 100
end
```

Enfin, vous voulez parfois que les essais ne se répètent jamais. Il y a trois façons d'obtenir cela dans l'instruction tasklist

- all\_before\_repeat : faire tous les essais comme dans le tableau, choisis au hasard, mais les faire tous avant de les choisir à nouveau
- no\_repeat : ne jamais répéter immédiatement un essai du tableau (c'està-dire que la même ligne d'un tableau ne sera pas répétée lors du prochain essai)
- fixed : faire les essais dans l'ordre où ils se trouvent dans le tableau

Dans l'exemple suivant, il n'y a que trois entrées sur la liste proposée. Elles ne seront pas répétées tant qu'elles ne seront pas toutes terminées. Vous pouvez ajouter explicitement que vous ne voulez jamais les répéter.

### Exemple de tasklist

```
table MyTable
bitmapBlue
bitmapRed
bitmapGreen

task strooptask
table MyTable
keys space
show bitmap @1
readkey 1
```

```
block MyTraining
  tasklist
    strooptask 10 all_before_repeat no_repeat
  end
```

Si vous avez fixé des blocs consécutifs, mais que vous utilisez la même tâche et le même tableau, vous continuerez à partir de la ligne du tableau d'où vous étiez parti. C'est une fonction pratique au cas où vous voudriez passer par de nombreuses essais, mais où les gens ont parfois besoin d'une pause. Assurez-vous que vous avez toujours une instruction end à la fin de votre liste de tâches

## 5.13.7.1 maxtime



Vous pouvez fixer la durée maximale d'un bloc (en millisecondes, secondes ou minutes). C'est très simple. Voici un exemple d'un bloc qui peut durer au maximum 2 minutes. Vous pouvez avoir des instructions avant la liste des tâches (par exemple, message ou pager, elles ne sont pas prises en compte dans le temps maximum).

Exemple de bloc avec maxtime de 2 minutes (notez le m)

```
block MyTraining
 maxtime 2m
 tasklist
    strooptask 10 all_before_repeat no_repeat
  end
```

Exemple de bloc avec maxtime de 100 s (notez le s)

```
block MyTraining
 maxtime 100s
 tasklist
    strooptask 10 all before repeat no repeat
  end
```

Exemple de bloc avec maxtime, avec des variables que vous devez spécifier en millisecondes

```
block MyTraining
 maxtime &MyVariable
 tasklist
    strooptask 10 all before repeat no repeat
  end
```

5.13. BLOCK 111

## 5.13.7.2 feedback \$\frac{1}{4}\$

Le feedback peut être utilisé pour créer un retour d'information au participant, par exemple sur la vitesse moyenne de réponse. La création d'un feedback fait partie de la structure "bloc". Le retour d'information est décrit dans un document distinct, en raison de son utilisation avancée : [comment utiliser le feedback (https://www.psytoolkit.org/doc3.4.0/feedback.html)

## 5.13.8 blockorder $\clubsuit$



Par défaut, les blocs sont effectués dans l'ordre où ils sont décrits dans votre code. Vous pouvez modifier cet ordre en réorganisant le code du bloc, mais il existe un moyen plus simple et plus rapide : l'instruction blockorder. En utilisant blockorder, vous pouvez lister les blocs que vous voulez faire et l'ordre dans lequel vous voulez les faire.

L'idée derrière cette fonction est simplement la rapidité du codage, et la facilité pour changer l'ordre (par exemple pour le contreblancement). Dans l'exemple suivant, imaginez qu'il existe un code pour trois blocs, appelés formation, test1 et test2. L'exemple suivant montre comment définir l'ordre de ces blocs de différentes manières.

### Exemple d'utilisation de blockorder

blockorder training test1 test2

Plus important encore, vous pouvez avoir plus d'un blockorder. Lorsque vous avez plus d'un blockorder, PsyToolkit en choisira un au hasard. Le blockorder choisi sera stocké dans la variable BLOCKORDER. Par défaut, cette variable contient la valeur 1, mais si vous avez donné, par exemple, 3 blockorder, cette valeur peut être comprise entre les valeurs 1 et 3. Vous trouverez ci-dessous un exemple de cette utilisation.

Dans l'exemple ci-dessous, l'ordinateur choisira l'un des deux blockorder au hasard. C'est ce que l'ordinateur fera lorsqu'il verra deux blockorder. C'est très utile pour faire un contrebalancement, par exemple si vous voulez que certains participants commencent avec une tâche et d'autres avec l'autre.

### Exemple d'utilisation du blockorder (notez que ces tâches ne font rien de significatif)

task color\_task

```
keys a l
  readkey 1 1000
 readkey
  save BLOCKORDER BLOCKNAME TASKNAME RT
task shape_task
 keys a l
 readkey 1 1000
 readkey
  save BLOCKORDER BLOCKNAME TASKNAME RT
blockorder
  training
  color_task
  shape_task
blockorder
  training
  shape_task
  color_task
```

## 5.13.9 include (usage avancé seulement)



Vous pouvez inclure un autre fichier dans votre script. Cela peut être utile si vous travaillez avec de très grands tableaux que vous voulez stocker dans un fichier séparé. Toute ligne commençant par "include" suivie d'un nom de fichier utilisera ce nom de fichier à cet endroit du script. Les fichiers inclus doivent se trouver dans le même répertoire que le fichier principal du script utilisant include.

## 5.13.10 part (pour utilisateur avancé)



Vous pouvez écrire des "bribes" de quelques lignes et les inclure ailleurs dans votre script. C'est pratique si vous avez plusieurs tâches ou blocs qui sont globalement les mêmes à l'exception de quelques lignes. Vous pouvez ainsi utiliser part au lieu de retaper tout le script. part remplace simplement le texte dans votre code. L'objectif principal est d'avoir un code plus court et plus efficace sans avoir à répéter les mêmes choses.

### Exemple d'utilisation de part

```
part showAnimatedSquare
  show rectangle 0 0 50 50 255 255 0
  delay 100
```

```
show rectangle 0 0 100 100 255 255 0
  delay 100
  show rectangle 0 0 150 150 255 255 0
  delay 100
part removeSquare
  clear -1 -2 -3
task some_task
 keys space
 part showAnimatedSquare
 readkey 1 1000
  part removeSquare
  delay 100
task another_task
  keys a
  part showAnimatedSquare
  readkey 1 5000
  part removeSquare
  delay 200
block test
  tasklist
   some_task 10
  end
block test
  tasklist
   another_task 10
  end
```

## 5.14 Vue d'ensemble de toutes les variables du système #

Vous trouverez ci-dessous un aperçu rapide de presque toutes les variables "système" que vous pouvez utiliser dans votre script d'expérience. Elles sont décrites en détail dans le texte, mais ici elles sont toutes ensemble. Elles contiennent généralement toutes une valeur numérique, sauf pour TASKNAME et BLOCKNAME. Ces deux derniers ne sont utiles que dans l'instruction de sauvegarde. Les variables liées au choix ne sont pas énumérées. Elles doivent toutes être écrites en majuscules.

• RT : dernier temps de réponse provenant de readkey ou readmouse

- STATUS: de readkey(s)/readmouse, il peut être CORRECT, WRONG, TIMEOUT (or 1, 2, 3)
- KEY : la touche qui a été préssée dansle dernier readkey/keystatus (un nombre)
- UNDER MOUSE : le numéro du stimulus sur lesquel le participant à cliqué
- MOUSE\_X : la position x de la souris dans le dernier état de readmouse
- MOUSE\_Y : la position x de la souris dans le dernier état de readmouse
- TABLEROW : ligne du talbeau, choisie au hasard une fois par essai
- TASKNAME : le noom de la tâche à laquelle appartient l'essai
- TRIALCOUNT : le nombre de d'essais réalisés jusqu'à présent dans le cadre de n'importe quelle tâche
- BLOCKNUMBER : Le numéro du bloc courant. Il démarre à 1.
- BLOCKNAME : le nom du bloc courant auquel appartient l'essai
- BLOCKORDER : Est normalement de 1, mais si plusieurs blockorder sont définis, donne le numéro d'e l'ordre de blocage'un blockorder choisi au hasard
- RATE : La valeur choisie de l'échelle de d'évaluation
- RATE RT : Le RT du processus d'évaluation
- RATE STATUS : sera de 1 s'il est noté, ou de 3 s'il y a un délai d'attente (il n'v a pas de réponse correcte, bien sûr)
- SHOW COUNTER

#### Exemple de show counter

```
task my_task
   show rectangle 0 0 100 100 0 255 0 # green rectangle
   show rectangle -200 0 100 100
                                  255 0 0 # red rectangle
   show rectangle 200 0 100 100
                                 0 0 255 # blue rectangle
   show text SHOW_COUNTER ## this will be 3
   delay 1000
   clear SHOW COUNTER ## delete last shown stimulus
```

SHOW COUNTER remplace l'option show-counter set. C'est rarement nécessaire, mais dans un code avancé avec beaucoup de stimuli, il peut être utile d'effacer plus tard les stimuli de l'écran (parce que vous pourriez perdre la trace des numéros de compteur de stimuli).

### Liens vers toutes les instructions 5.15



(cette liste n'est pas encore complète)

Un court document sr la syntaxe est disponible ici (en anglais)

- Options basiques : origin | bitmapdir | sounddir | fontdir | fullscreen | resolution | version | mouse| variable | window | screensize | screendistance | vsync | egi| escape| parallelport| pcidio24| cedrus | iolab| eye-tracker| executable
- Instructions de base : key| show | bitmap | text| clear| sound| keys| readkey | delay | save | table
- Instructions moins souvent utilisées : readmouse | if | while | move | relocate | hide | unhide | font drag | dragging
- Instructions avancées : readkeys | readmouse | keystatus | choose | set | progress | timestamp | sprite | end
- Instruction de bloc : bitmap | sound | delay | message | pager | set | tasklist | feedback | block-maxtime
- Instructions spéciales : include | part | blockorder
- vue d'ensemble des variables systèmes

## Chapter 6

# Un projet complet de recherche en ligne #

Voici la description complète d'un projet de recherche psychologique en ligne avec des tutoriels vidéo. Cette enquête comprend une expérience en ligne. Si la programmation d'une expérience en ligne ne vous intéresse pas, et si vous souhaitez seulement faire un questionnaire, vous pouvez consulter la leçon à ce sujet\*. Cette leçon vise en fait à comprendre les expériences psychologiques cognitives et la manière de les intégrer dans les études en ligne. Elle peut être utile à tout chercheur, y compris aux étudiants en psychologie qui réalisent leur projet de recherche.

Cette leçon vous apprendra à utiliser le Psytoolkit pour faire ce qui suit :

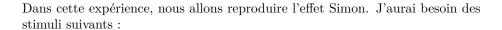
- Programmer une expérience à l'aide PsyToolkit
- Tester l'expérience
- Analyser les données des tests
- Intégrer l'expérience dans un questionnaire en ligne
- Obtenir des participants en ligne
- Réaliser l'étude en ligne et collecter les données en ligne
- Analyser les données du groupe avec SPSS ou R

## 6.1 Étape 1. Réfléchissez à votre expérience et créez des stimuli

Avant de commencer à faire quoi que ce soit, il vous faut réfléchir à votre tâche et d'utiliser du papier et un crayon pour dessiner ce à quoi vous voulez que tout cela ressemble. Voici les choses auxquelles il faut penser en général.

- 1. Qu'est-ce que je veux mesurer exactement, et est-ce que cela peut être implémenté dans PsyToolkit (il y a des limitations avec tout type de logiciel)
- 2. Combien de temps l'étude doit-elle durer?
- 3. Combien d'essais
- 4. À quoi doivent ressembler les stimuli (vous pouvez aussi avoir des sons, mais ils ne sont pas utilisés dans cet exemple de projet)

### 6.2 Étape 2. Créez les stimuli 🟶



- 1. Une flèche pointant vers la gauche
- 2. Une flèche pointant vers la droite
- 3. Une croix de fixation
- 4. Instructions au début de l'expérience
- 5. Message d'erreur en cas d'erreur
- 6. Informations pour dire aux gens qu'ils vont s'entraîner (bloc 1)
- 7. Informations pour dire aux gens qu'ils vont faire une vraie collecte de données (bloc 2)
- 8. Un message de remerciement lorsqu'elles sont terminées.

Nous savons que la résolution par défaut des expériences est de 800 par 600 pixels, il faut donc faire en sorte que les stimuli soient adaptés à cette résolution.

- Vous devez apprendre à utiliser un peu Inkscape (ou tout autre logiciel de dessin capable d'exporter des fichiers image)
- Open Inkscape
- Dessiner les stimuli, exporter les stimuli, enregistrer le fichier inkscape (SVG)

Regardez la vidéo (15 minutes) sur la façon de dessiner les stimuli (plein écran recommandé). C'est de loin la vidéo la plus longue ici, mais elle vous permet vraiment de vous lancer dans l'une des parties les plus importantes de la conception d'une expérience, à savoir la création de bons stimuli! Le son est un peu faible sur les premières vidéos, il faut donc augmenter le volume!

Dans la vidéo, les stimuli sont enregistrés au format PNG. Mais PsyToolkit peut fonctionner avec n'importe quel type de bitmap, comme JPEG, BMP, GIF, etc. Le PNG est choisi ici parce que c'est ce qu'Inkscape exporte. Consultez la documentation pour plus de détails.

### Étape 3. Créer une nouvelle expérience 🖶 6.3



- Connectez-vous et créez une nouvelle expérience
- Appelez-la "simon".
- Commencez à coder dans la case blanche du haut
- Utilisez l'onglet "scripting" pour vous aider

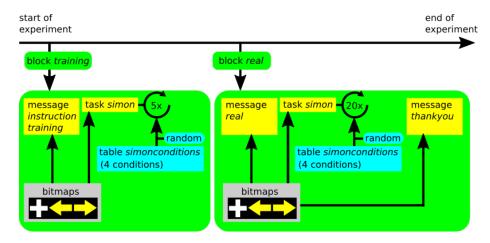
Regardez la vidéo (2 minutes) pour voir comment créer une nouvelle expérience et démarrer le code :

## Étape 4. Coder l'expérience \$\frac{\pi}{2}\$

Avant de commencer ici, il est important de comprendre la structure d'une expérience codée dans le langage de script PsyToolkit. Ceci est expliqué dans la documentation en ligne. En voici un bref résumé :

- Une expérience est décrite entièrement sous forme de texte.
- Vous devez saisir le texte manuellement dans la zone de texte de votre navigateur
- Une expérience comporte différentes sections (séparées par des lignes vides), et ce sont celles qui sont utilisées ici : \*\* bitmaps pour indiquer à l'ordinateur quelles images sont utilisées pour les stimuli visuels \*\* task consistant à indiquer à l'ordinateur la séquence des événements, le moment, les réponses, etc. dans la tâche Simon \*\* table pour décrire les quatre conditions différentes utilisées dans la section des tâches \*\* block pour décrire un bloc d'essais et présenter les principaux messages, tels que l'instruction et le message de remerciement

La description de ces sections n'est pas exactement l'ordre des événements de l'expérience. L'expérience commence réellement avec le premier bloc, même si celui-ci est placé à mi-chemin dans le code. L'algorithme du script et son déroulement est présenté dans le diagramme ci-dessous :



Le code de l'expérience se trouve ci-dessous, juste à titre de référence. Chaque ligne de code est décrite plus en détail dans les vidéos ci-dessous. Mais il est bon d'y jeter un coup d'œil rapide pour que vous sachiez de quoi il s'agit.

```
bitmaps
  leftarrow
  rightarrow
  training
  real
  instruction
  thankyou
  errorfeedback
  fixpoint
table simonconditions
  "leftpos leftA com"
                        leftarrow -200 1
  "rightpos rightA com" rightarrow 200 2
  "leftpos rightA inc"
                        rightarrow -200 2
  "rightpos leftA inc"
                        leftarrow
                                    200 1
task simon
  table simonconditions
  keys a 1
  show bitmap fixpoint
  delay 200
  clear 1
  show bitmap @2 @3 0
 readkey @4 2000
  clear 2
  if STATUS != CORRECT
    show bitmap errorfeedback
```

```
delay 500
    clear -1
  fi
  delay 500
  save @1 STATUS RT BLOCKNAME
block training
 message instruction
  message training
  tasklist
    simon 5
  end
block real
  message real
  tasklist
    simon 20
  end
  message thankyou
```

### Définition et mise en ligne des stimuli 6.4.1



- d'abord informer l'ordinateur des stimuli et les télécharger
- Afin de dessiner des stimuli à l'écran, vous devez fournir une coordonnée X et une coordonnée Y. Les coordonnées XY 0 0 se rapportent au centre de l'écran. Les coordonnées XY -200 0 se réfèrent à une position à mi-chemin à gauche du centre de l'écran, etc.

Regardez la vidéo (3 minutes) comment donner ces instructions à l'ordinateur

### Définir les conditions 6.4.2



créer un tableau PsyToolkit qui contient des informations sur les 4 conditions :

- 1. Stimulus présenté à gauche + Flèche pointant vers la gauche = compatible
- 2. Stimulus présenté à droite + flèche pointant vers la droite = compatible
- 3. Stimulus présenté à gauche + flèche pointant vers la droite = incompatible
- 4. Stimulus présenté à droite + flèche pointant vers la gauche = incompatible

Regardez la vidéo (5 minutes) sur la façon de programmer les 4 conditions.

## 6.4.3 Décrire les événements 🕀



Chaque événement d'un procès est décrit dans la section des tâches

Regardez la vidéo (7 minutes) sur la manière de décrire la tâche avec les événements d'un essai.

## 6.4.4 Blocs #

Nous allons maintenant entrer les informations des blocs

- Message de bienvenue
- Bloc d'entrainement de 5 essais
- Bloc de données réel de 20 essais
- Message de remerciement

Regardez la vidéo (4 minutes) sur la façon de mettre en place des blocs d'essais et des messages généraux de bienvenue/instructions/remerciements.

### Étape 5. Compiler et exécuter # 6.5



Maintenant que votre code est tapé et que vos stimuli sont téléchargés, essayez de compiler. Le compilateur a probablement dit qu'il y avait une erreur, simplement parce qu'il est très facile de faire une quelque part. Ne vous inquiétez pas, l'ordinateur vous aidera à trouver où l'erreur s'est produite.

Regardez la vidéo (5 minutes) qui explique comment compiler et exécuter votre expérience dans votre navigateur, et comment enregistrer les données automatiquement.

Le codage est souvent un processus d'essais et d'erreurs. Le retour d'information sur les erreurs dans l'encadré jaune devrait vous donner un indice sur ce qu'il faut améliorer. Pouvez-vous déterminer ce que vous avez fait de mal?

Une fois que la compilation ne renvoie pas de messages d'erreur, vous pouvez tester votre expérience en cliquant sur le bouton "Exécuter".

À la fin de l'expérience, l'ordinateur peut vous montrer vos données et même les enregistrer dans votre compte. Assurez-vous de bien choisir cette option, car nous avons besoin des données pour l'étape suivante.

### Étape 6. Analysez votre dossier de test # 6.6



- Dans l'expérience, naviguez vers le bas jusque "analyze", et sélectionnez
- Vous pouvez sélectionner le fichier de données que vous venez de créer. C'est maintenant le fichier sur lequel vous travaillez (vous pouvez avoir plusieurs fichiers)
- Dans les cases, vous pouvez saisir les éléments les plus importants pour l'analyse des données.
- Chaque ligne de votre fichier de données est un essai dans la tâche.
- Sélectionnez la colonne qui correspond au RT, à l'erreur, au bloc
- Sélectionnez également que vous souhaitez exclure les données relatives à la formation de votre analyse de données
- Sélectionnez également que vous souhaitez exclure les procès immédiatement après une erreur
- Cliquez sur "analyser" et regardez vos réponses moyennes.

En principe, si vous menez une expérience en laboratoire, vous pouvez simplement effectuer votre expérience et enregistrer les fichiers de données recueillis ici. Si vous le faites, assurez-vous de tenir un journal de bord de votre étude.

En psychologie cognitive, nous recueillons généralement des données auprès d'un certain nombre de participants. Par exemple, dans la tâche de Simon, vous pouvez vous contenter de 10 à 20 participants. Ensuite, dans l'analyse de groupe, nous comparons les scores moyens de chaque participant dans une analyse de groupe, qui sera expliquée plus loin.

Regardez la vidéo (5 minutes) sur la façon d'analyser votre fichier de données de test.

À ce stade, vous avez appris à coder une expérience et à faire une analyse de base. Dans la suite de la description de ce projet, nous allons intégrer l'expérience Simon dans un questionnaire/enquête en ligne et collecter des données en ligne. Ensuite, nous téléchargerons et analyserons les données.

#### Étape 7. Mise en place du projet en ligne # 6.7



## Mettre en place une enquête en ligne 🚏



- Maintenant que votre expérience fonctionne, vous êtes prêt à mettre en place une étude en ligne dans laquelle vous intégrerez l'expérience.
- Regardez les fonctions sur la gauche, et sélectionnez "Créer une étude".
- Appelez-le "simon\_survey".

Regardez la vidéo (3 minutes) sur la façon de mettre en place les premières parties d'une étude en ligne.

#### 6.7.2Commencez par des questions et une enquête test #

- Dans la case du haut se trouve le code de l'enquête.
- Chaque question se compose de quelques lignes et il y a beaucoup de documentation
- Pensez aux choses que vous voulez savoir de vos participants
- Au minimum, vous voulez connaître l'âge
- Vous devez également configurer l'écran d'introduction/de bienvenue
- Vous devez rendre l'étude "en ligne".
- Vous devez compiler et tester l'enquête

Le bouton save de la page de codage d'enquête devient jaune lorsque le code n'a pas été enregistré. Enregistrez souvent si vous ne voulez pas perdre

Regardez la vidéo (5 minutes) sur la façon de commencer à saisir les questions du questionnaire en ligne dans lequel l'expérience sera intégrée.

### Terminer l'enquête 🚏 6.7.3



- Nous ajoutons quelques questions provenant de la bibliothèque des enquêtes
- Intégrer l'expérience
- Compilation et test
- Passer à un mode de collecte de données réel
- Prêts à demander à vos participants!

Regardez la vidéo (5 minutes) pour voir comment l'expérience est intégrée dans le questionnaire en ligne.

Comme le code de l'expérience, vous pouvez voir le code réel de l'enquête, juste pour vous guider. Chaque ligne de code est décrite plus en détail dans les vidéos.

- 1: age
- t: textline
- q: How old are you?
- {min=18,max=100} Enter your age

scale: side

- {score=-100} Always left
- {score=-50} Usually left
- {score=0} Both equally
- {score=50} Usually right
- {score=100} Always right
- 1: questions
- t: scale side
- o: width 20%
- q: Please indicate your preferences in the use of hands in the following activities or objects
- Writing
- Throwing
- Toothbrush
- Spoon
- 1: score
- t: set
- mean \$questions
- 1: simonexperiment
- t: experiment
- simon

## 6.8 Étape 8. Analyser les données 🟶



## 6.8.1 Assurez-vous que l'analyse de votre expérience est paramétrée

Le système doit savoir quel bloc de données vous voulez inclure (par exemple, vous n'êtes probablement pas intéressé par le bloc de formation), etc.

Regardez la vidéo (1,5 minute) sur la façon de vous préparer à l'analyse des données de tous les participants à votre étude.

## 6.8.2 Téléchargez vos données 🚏

Une fois que vos participants ont fait l'expérience, vous êtes prêt à récupérer vos données sur le serveur.

- Regardez la vidéo (3 minutes) pour voir comment télécharger les données, et quels sont les fichiers dans les fichiers data.zip
- Il est possible d'inclure des fichiers de données dans un format compatible avec SPSS, ce qui permet d'introduire très facilement vos données dans  $SPSS^1$
- Regardez la vidéo (3 minutes) sur la compréhension du processus de téléchargement et la signification des différents fichiers dans le téléchargement.

## Ouvrez votre fichier de données dans SPSS #



Dans cette section, nous allons analyser les données avec SPSS. Il est supposé que vous avez installé SPSS, qui est populaire dans l'enseignement de la psychologie. Bien sûr, les données sont dans un format générique et peuvent être analysées avec n'importe quel programme de statistiques. La procédure d'importation des données est analogue.

Regardez la vidéo (8 minutes) sur la façon d'importer et d'analyser votre ensemble de données PsyToolkit dans SPSS.

### 6.8.4 Et c'est tout

Nous espérons que cela vous aidera à mettre en place votre propre étude. Si vous avez encore des questions, n'hésitez pas à m'en faire part par e-mail.

<sup>&</sup>lt;sup>1</sup>D'autres formats de données sont possible, notamment csv qui permet d'importer les données dans presque tous les logiciels de statistiques : R, Jmaovi, jmp.

## Chapter 7

## Les questionnnaires #

Ceci est une introduction, pour une présentation complète, regardez ici. Avec PsyToolkit, vous pouvez réaliser différents types d'enquêtes en ligne :

- De simples enquêtes en ligne sans expérimentation sur le temps de réaction (elles peuvent être réalisées sur des plateformes mobiles, comme les téléphones). Par exemple, une enquête sur la satisfaction des étudiants ou sur les attitudes et opinions des gens. Consultez également la bibliothèque qui propose des questionnaires prêts à l'emploi.
- Enquêtes en ligne **avec** expérimentation des temps de réaction. C'est l'idéal pour les études de psychologie cognitive ; vous pouvez utiliser l'enquête pour obtenir des informations générales sur les participants et ensuite lancer l'expérience dans le navigateur.
- Vous pouvez poser un nombre illimité de questions.
- Prise en charge des instructions aux participants en 5 langues (anglais américain et britannique, chinois simplifié, néerlandais et allemand). Par exemple, si vous choisissez le chinois simplifié, le texte (par exemple, "continue") des boutons est en chinois.
- Outils d'analyse en ligne de base des enquêtes complétées.
- Saut conditionnel aux questions basées sur les réponses données précédemment
- Randomisation de l'ordre des questions de sections spécifiques d'un questionnaire.
- Tous les types de questions courantes, telles que les questions à choix unique, les questions à choix multiples, les échelles de Likert, les curseurs, les lignes de texte, les zones de texte, etc.

Le support multilingue fait référence à l'interface utilisateur, comme le texte des boutons, etc. Bien entendu, vous devez toujours rédiger vos questions dans

la langue dans laquelle vous voulez qu'elles soient présentées. Jetez un coup d'œil aux enquêtes/questionnaires prêts à être téléchargés dans la bibliothèque d'enquêtes. \*\*\*\*\*

### Introduction # 7.1



Internet est idéal pour réaliser des enquêtes en ligne, et il existe divers très bons outils web que vous pouvez utiliser gratuitement (il suffit de faire une recherche sur Google pour trouver "enquête en ligne gratuite"). Il est possible qu'il existe d'autres outils plus adaptés à vos besoins (PsyToolkit est gratuit et il n'y a aucun intérêt à vous le "vendre" - le but est simplement de vous donner la meilleure information possible sur ce qu'il est et vous décidez vous-même).

PsyToolkit a son propre interface de conception de questionnaires, mais il n'a pas d'interface utilisateur sophistiquée (comme certains autres outils en ligne). Toutes les questions doivent être tapées dans une zone de texte au format texte, en suivant un protocole de formatage spécial. cependant, Ce n'est pas difficile. PsyToolkit vous aide à créer des enquêtes très simples avec des types de questions standard, telles que des questions dans lesquelles vous pouvez choisir un élément parmi plusieurs, choisir plusieurs éléments parmi plusieurs, saisir une ligne de texte, etc. C'est similaire à ce que proposent d'autres outils d'enquête, et si c'est tout ce que vous voulez faire, vous pouvez utiliser d'autres outild (par exemple, [zoho.com] qui est assez bon).

Vous pouvez également avoir des conditions, c'est-à-dire que certaines questions ne seront posées qu'en fonction des réponses à d'autres questions.

Lorsque vous avez créé et compilé votre enquête, vous obtenez un lien URL. Vous pouvez envoyer ce lien aux personnes que vous souhaitez faire participer à l'enquête. Les données collectées seront enregistrées sur le serveur du PsyToolkit et peuvent être téléchargées sous forme de données brutes (avec quelques informations supplémentaires, comme l'heure à laquelle la réponse a été donnée) et d'un fichier tableur des réponses/scores par participant.

Une possibilité unique des enquêtes en ligne dans PsyToolkit est que vous pouvez y intégrer des expériences sur les temps de réaction typiques de la psychologie cognitive. En fait, c'est la seule raison pour laquelle j'ai développé les enquêtes en ligne avec PsyToolkit.

Les données des enquêtes et les données sur les temps de réaction peuvent être téléchargées et analysées sur votre propre ordinateur. Ainsi, les enquêtes en ligne avec PsyToolkit sont principalement destinées aux psychologues qui souhaitent mener des expériences en ligne sur les temps de réaction. Bien entendu, si vous souhaitez simplement faire un questionnaire, vous pouvez également le faire. En ce sens, les questionnaires PsyToolkit en ligne peuvent être utiles à toute personne souhaitant réaliser une enquête en ligne.

Dans de nombreuses expérience, vous souhaitez obtenir des informations de base sur les participants, comme leur sexe et leur âge. c'est la raison qui nous a conduit à intégrer une expérience sur le temps de réaction dans une enquête. Si vous voulez savoir comment programmer une expérience sur le temps de réaction dans le PsyToolkit, vous devez lire l'introduction au script d'expérience du PsyToolkit.

### Comment créer une enquête 🟶 7.2

Les enquêtes sont créées sous forme de texte dans une zone de texte du navigateur. Cette méthode est différente des autres sites d'enquêtes en ligne. Psy-Toolkit ne dispose pas d'une interface graphique de type "pointer-cliquer" permettant de créer chaque question (qui sera peut-être créée à l'avenir). Néanmoins, la création d'une enquête n'est pas très difficile!

La première chose à faire est de créer votre propre compte sur le site web de PsyToolkit. Ensuite, explorez-le pour vous familiariser avec l'interface. Si vous sélectionnez "créer une enquête", vous obtiendrez un exemple de question. Vous pouvez supprimer l'exemple et entrer un nouveau questionnaire. La programmation d'un questionnaire est expliquée ci-dessous. La façon de l'exécuter réellement est expliquée ci-dessous dans la section "Exécution des enquêtes".

Ce qu'il faut vraiment savoir, c'est qu'il existe différents types de questions. Chaque question doit être décrite en quelques lignes, et la façon de le faire suit un format très spécifique (je l'appelle la syntaxe de l'enquête). Ce qu'il faut comprendre, c'est que chaque question comporte un certain nombre de champs, qui sont marqués d'un symbole au début de la ligne. Le premier champ de chaque question est le "label", qui s'écrit simplement "l :". Ce qui suit est l'étiquette proprement dite. Les étiquettes ne doivent pas comporter d'espaces

Les enquêtes sont simplement constituées de texte et sont très simples à écrire. Chaque question est constituée de quelques lignes de texte. Entre les questions, il y a une ligne vide. Il est préférable de regarder l'exemple à la fin de ce document pour le comprendre.

### Réalisation des enquêtes 🟶 7.3



Maintenant que vous savez comment rédiger les bases d'un questionnaire, vous pourriez en faire un. Vous devez "compiler" les informations que vous avez saisies rendre l'enquête exécutable.

## 7.4 Exemple complet \$\Pi\$

Exemple sans expérience intégrée. Notez que dans l'exemple suivant, il n'y a que deux questions (avec une ligne vide entre les deux questions). La deuxième question utilise également des balises html, dans ce cas, le texte entre et est affiché en gras.

### Exemple

```
option: survey-name: Childhood memory
1: gender
t: radio
q: What is your gender?
- Female
- Male
1: age
t: range
q: What is your age?
- {min=60,max=120} Select your age
1: earliest
t: textbox
q: Please write down your <b>earliest 3 memories of personal event</b>. For each even,
- event1
- event2
- event3
```

Notez que l'exemple suivant comporte une expérience et une question de l'échelle intégrée Big 5 sur la personnalité. Notez deux choses importantes : Chaque étiquette n'a pas d'espace, c'est toujours un seul mot. Et deuxièmement, entre les questions, il y a une ligne vide.

### Exemple

Agree a littleAgree moderately

option: survey-name: Psychological Studies cognitive demonstration
scale: agree
- Disagree strongly
- Disagree moderately
- Disagree a little
- Neither agree nor disagree

Very awake and fitNeither fit nor tiredI feel a bit tiredI feel very tired

- Agree strongly 1: firsttime t: radio q: Is this the first time that you do this specific study? - Yes, first try - No. I already tried and finished it before and wanted to try it again - No. I tried it before, but it did technically not work, and therefore try again 1: where t: radio q: Where are you right now? - Alone in a room, and it is quiet - Alone in a room, but it is not quiet at all (for example because of listening to music/tv) - Outside university, in a room with other people, but it is pretty quiet - Outside university, in a room with other people, and it is not quiet - In a university room or library - Somewhere else 1: gender q: Are you male or female? t: radio - Male - Female - Do not want to answer 1: student q: Are you a psychology student? t: radio - Yes - No 1: computer q: Do you regularly play computer games? t: radio - Yes, at least once a week - No, less than once a week or never 1: tired q: How fit do you feel? t: radio

```
1: tipi
q: I see myself as ...
t: scale agree
- I see myself as Extraverted, enthusiastic
- I see myself as Critical, quarrelsome
```

- I see myself as Dependable, self-disciplined
- I see myself as Anxious, easily upset
- I see myself as Open to new experiences, complex
- I see myself as Reserved, quiet
- I see myself as Sympathetic, warm
- I see myself as Disorganized, careless
- I see myself as Calm, emotionally stable
- I see myself as Conventional, uncreative
- 1: simontask
- q: Please carry out the experiment now, it takes around 10 minutes
- t: experiment
- classroom\_simon
- 1: comments
- t: textbox
- q: Do you have any comments?
- Type comments in the box below:

## Chapter 8

## Les questionnaire en ligne

Pour utiliser les questionnaires en ligne, voir l'introduction aux questionnaires en ligne. Le langage de script d'enquête est complètement différent du langage de script d'expérience. Si vous souhaitez intégrer une expérience dans une enquête en ligne, vous devez suivre ces instructions.

### Structure d'un questionnaire # 8.1



Les questionnaires sont de simples fichiers texte. Chaque ligne a une signification. L'ordinateur lit le questionnaire ligne par ligne. Le premier caractère de chaque ligne a généralement une signification particulière.

- Chaque questionnaire se compose de plusieurs questions, codées sous forme de texte simple (que vous pouvez éditer dans votre navigateur)
- Le questionnaire doit être saisi manuellement, ligne par ligne (exemples ci-dessous)
- Chaque question se compose de quelques lignes
- Les questions sont séparées les unes des autres par une (ou plusieurs) ligne
- Chaque question, quel qu'en soit le type, a une structure de base simple (pour en savoir plus sur la structure des questions)
- Chaque questionnaire peut avoir des sections qui randomisent l'ordre (en savoir plus sur l'ordre randomisé)
- Chaque questionnaire peut comporter des commentaires sur les lignes commençant par le signe hashtag (#comment)

### Structure générale d'une question



- Chaque question est composée d'un certain nombre de lignes de texte.
- Les questions sont séparées les unes des autres par une ligne vide
- Chaque ligne commence (généralement) par une lettre suivie d'un deuxpoints. La lettre décrit une caractéristique importante, comme le nom de la question ou le type de question.
- Chaque question comporte un ou plusieurs éléments, chacun commençant par un signe moins
- La première ligne d'une question commence par l:. La lettre l signifie "étiquette (label)". Chaque question a une étiquette, ou un "nom". Et ce, pour deux raisons: 1) L'ordinateur a besoin d'un moyen d'identifier une question, et il utilise le label. 2) L'utilisateur a besoin d'un moyen d'identifier les réponses aux questions, et label est le moyen le plus simple pour y parvenir.
- Un label ne peut comporter que des lettres (majuscules ou minuscules) et des chiffres, ainsi que le caractère de soulignement (underscore).
- Chaque question doit commencer par l:, mais il n'est pas nécessaire de donner une étiquette. Dans ce cas, le logiciel appellera la question "auto suivie d'un chiffre. C'est pratique si vous n'êtes pas disposé à étiqueter vos questions. Vous pouvez toujours les identifier par l'ensemble de la question. Et pour les sauts de question, le label n'est généralement pas nécessaire pour les utilisateurs.

### Exemple de structure de la question. La flèche <- sert uniquement pour introduire une explication

```
1: MyQuestion1
                                  <-- this is the label
                                  <-- this is the type of question
t: radio
q: What is your favourite movie? <-- this is text of the question the participant sees
- Toy Story
                                  <-- answer 1
- Spirited Away
                                  <-- answer 2
- Babe
                                  <-- answer 3
- Ratatouille
                                  <-- answer 4
                                  <-- the empty line separating this question from the :
```

- Chaque question peut faire apparaître une image en haut. Vous devez télécharger le fichier image (png, bmp, gif, jpg, etc.). Voir les images.
- Vous pouvez utiliser la ligne d'option o: random pour rendre aléatoire l'ordre des options de réponses.

### Commentaires pour les programmateurs {#comment}



Lorsque vous programmez votre enquête, vous pouvez parfois vouloir faire un commentaire qui vous est destiné et que l'ordinateur doit ignorer. Vous pouvez utiliser le hashtag pour cela.

Contrairement aux scripts d'expérimentation du PsyToolkit, le symbole de commentaire "#" doit se trouver au début d'une ligne. En d'autres termes, vous ne pouvez pas avoir de commentaires commençant au milieu d'une ligne.

### Formatage du texte \$\Pi\$ 8.2



Vous pouvez formater le texte en utilisant les balises HTML standard. Les plus courantes sont celles pour le gras, l'italique et la couleur du texte. La couleur générale de l'écran ne peut pas être modifiée.

- Tout ce qui se trouve entre et sera présenté en caractères gras.
- Tout ce qui se trouve entre et sera présenté en italique.
- Tout ce qui se trouve entre et sera présenté en rouge, etc.

### Exemple de formatage du texte

- 1: MyQuestion
- t: radio
- q: What is your <i>favourite</i> movie?
- <b>Toy Story</b>
- <font color='blue'>Spirited</font> Away
- Babe
- Ratatouille

## Types de questions #



Il y a les types de questions suivants :

- radio Un choix parmi de nombreuses options
- check les choix multiples parmi les nombreuses options (cases à cocher)
- experiment Intégrer une expérience
- scale Utiliser des échelles de type Likert
- range Permet à l'utilisateur de saisir une plage de nombres entre deux valeurs

- textline Le participant peut saisir seulement une ligne de texte
- textbox Le participant peut saisir plusieurs lignes de texte
- multiradio Plusieurs éléments à choix unique dans une question

Il existe les types suivants qui ne présentent que des informations et un bouton "Continuer

- info Montrez juste quelques informations, un message, etc.
- youtube Montrer une vidéo YouTube

Il existe différents items de contrôle :

- jump saut vers un label de question (conditionnel ou inconditionnel)
- set Fixer une étiquette de question (pour la notation et les nombres aléa-
- random aléatoriser des questions ou contrebalancer les versions d'un ques-

Vous pouvez insérer dans une question du son ou une vidéo :

- Play a sound Présenter un son dans une question
- Play a video Présenter une vidéo dans une question
- youtube Jouer une vidéo youtube dans une question

### Radio: Sélectionner un parmi plusieurs 🕏 8.3.1



Utilisez cette option lorsque vous souhaitez qu'une seule réponse possible soit sélectionnée. Le nom "radio" vient des radios classiques, qui avaient un ensemble de boutons dont au maximum un pouvait être enfoncé.

### Exemple

- 1: MyQuestion1
- t: radio
- q: What is your favourite movie?
- Toy Story
- Spirited Away
- Babe
- Ratatouille

Le paramètre "q:" est le seule qui peut s'étendre sur plusieurs lignes. Cela peut parfois être pratique pour poser des questions plus longues, comme dans l'exemple ci-dessous, qui utilise le format HTML.

### Exemple de question sur plusieurs lignes

This is item 1 out of 1 items.	
What is your favourite movie? Toy Story Spirited Away Babe Ratatouille	
Click this button to continue	

Figure 8.1: exemple de bouton radio

- 1: MyQuestion1
- t: radio
- q: In the following question, we would like to which movie you like best. Make sure you understand the following:
- <u1>
- This is only about movies
- You are allowed to say that you are not sure.
- Toy Story
- Spirited Away
- Babe
- Ratatouille
- I am not sure

Au lieu d'un texte, vous pouvez afficher une image. Vous devez télécharger les images (comme pour topimage) et ensuite utiliser l'option {image=nom de fichier}. Dans les fichiers d'affichage, vous pouvez voir les noms de fichiers que vous avez déjà. N'utilisez pas les fichiers comportant des espaces. Vous devez saisir le nom de fichier complet, y compris l'extension, comme jpg, png, bmp ou gif. Cela vous permet même de télécharger de petites animations en utilisant des gifs animés.

### Exemple avec des images de type radio téléchargées

- 1: MyQuestion1
- t: radio
- q: In the following question, we would like to which movie you like best. Make sure you understand the following: <ul>