PsychoPy

Réglages initiaux

Expermient Settings :

- Basic :
  - Experiment name : changer le nom de l'expérience
  - Show info dialog : permet d'adapter la boîte de métadata et d'adapter les informations demandées.
    Attention à toujours garder le champ « participant » sinon le script plantera.
  - Enable escape key : permet de sortir du script avec la touche Esc.
  - Use PsychoPy version : permet de choisir les versions de PsychoPy.
- Screen :
  - Full-screen window et Window size (pixels) : permet de faire en sorte que l'écran de l'expérience n'occupe pas tout l'espace.
  - Color : changer la couleur du fond d'écran lorsque l'expérience tourne. L'identification de la couleur est donnée en code rgb (champ Color space). Ce code correspond à 3 chiffres compris en -1 et 1 qui codent la quantité de rouge, vert et bleu composant la couleur voulue. Il y a 3 façons de changer la couleur :
    → Utiliser les codes couleurs dans les annexes.
    Attention : il faut respecter les points, les virgules, les crochets et ne pas oublier le **$** devant le code !
    → Entrer le nom de la couleur.
    → Clic droit sur le champ Color et choisir une couleur.
  - Units : concerne les unités de mesures des éléments visuels. Il faudra cocher Pix.
    → Height : permet de tout spécifier relativement à la hauteur de la fenêtre PsychoPy.
    Attention : hauteur de la fenêtre et non de l'écran).
    → Pix : permet de spécifier les tailles et emplacements en pixels. Cela a le désavantage que la taille et la position des objets est spécifique à l'écran, mais cela a l'avantage d'être intuitif et simple d'utilisation. C'est ce qu'il faut choisir dans le cadre du cours.
    Attention : si nous spécifions Pix dans Expermient Settings, cela nous permet de ne pas avoir à le spécifier pour chaque composant !

Remarque : les décimales s'écrivent avec des points, **non** des virgules.

Choisir le format du fichier de réponse

Expermient Settings :

- Data : il faut cocher toutes les case : Save excel file, Save csv file (summaries), Save csv file (trial-by-trial), Save psydat file (automatiquement coché) et Save log file.
  Fichier **.log** : fournit un enregistrement chronologique de tout ce qui s'est passé pendant l'expérience. Il ne nous est pas très utile, excepté dans un but de debuggage avec recherche des messages d'alerte (warnings).
  Fichier **.psydata** : plus riche que les trois suivants et peut être traité par des utilisateurs avancés en Python.

Fichier **.xlsx** : tableur présentant des données dont chaque ligne correspond à un item de la liste externe. Les premières colonnes représentent les colonnes de votre liste externe et les suivantes donnent des informations critiques :

- *n* : nombre de répétitions de chaque item.
- *resp2.keys_raw* : touche de réponse appuyée par le participant.
  'resp2' = nom donné à l'objet Clavier dans notre script
- *resp2.rt_mean* : temps moyen de réponse (en secs) pour cet item. Si l'item n'a été présenté qu'une fois, ce champ est identique à resp2.rt_raw. Si l'item a été répété x fois, la valeur correspond à la moyenne des x répétitions.
- *resp2.rt_raw* : temps de réponse pour l'item lors d'une présentation donnée.
- *resp2.rt_std* : s'il y a eu plusieurs répétitions de l'essai, ce champ donne une indication de la variabilité des temps de réponse (déviation standard).
- *order* : ordre d'apparition des items. Notez que cela commence à 0 et non à 1.

Fichier **'boucle.csv'** : présente exactement les mêmes données que le fichier Excel, sauf qu'il est au format texte. Le mot précédant le '.csv' correspond au nom que vous avez donné pour l'objet boucle dans le Flow.

Fichier **.csv** : présente les mêmes données que le fichier 'boucle.csv' excepté qu'une ligne correspond à une réponse du participant. Ce sont ceux-ci que nous allons utiliser dans R.

Paramétrer les fichiers

Il est possible d'ajouter des caractéristiques à la liste externe. En effet, il est intéressant d'en ajouter certaines, surtout si c'est un facteur manipulé dans l'expérience, pour que la colonne soit automatiquement présente dans le fichier de résultat et ainsi faire aisément une analyse dans R sur base de ce facteur.

Le Builder permet de coder l'exactitude des réponses des participants. Cela signifie qu'un champ indiquera 1 (= bonne réponse) ou 0 (= mauvaise réponse) en fonction de la réponse du sujet. Pour cela, il est utile de préparer une colonne 'RepCorrecte' dans la liste externe.

Créer une routine

Cliquer sur Insert Routine dans le panneau Flow et sélectionner New. Il faut alors entrer le nom de la routine en sachant que :

- ✓ Il doit avoir un sens,
- ✓ Chaque nom doit être unique,
- ✓ Il doit être en alphanumérique.

Il n'est pas non plus possible de donner deux fois le même nom à un objet, ni d'utiliser le même nom pour une routine et un objet.

Ajouter un composant

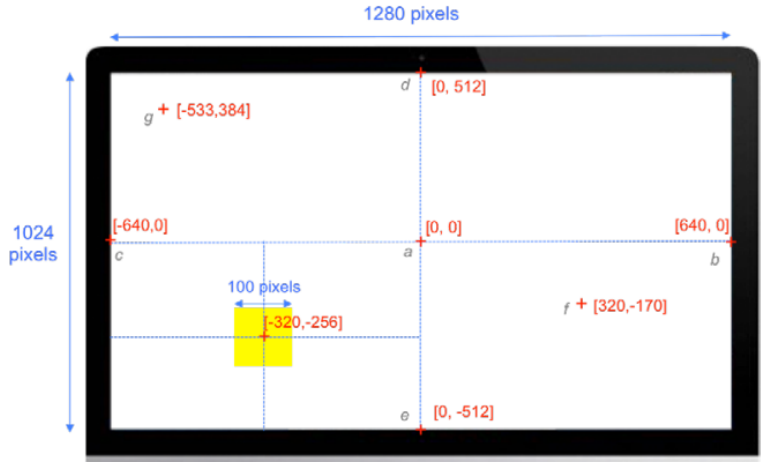
Consigne

Cela peut se faire par l'objet texte du logiciel.

Remarque : lorsque la consigne est longue, il est parfois plus facile d'importer une image que de l'écrire dans PsychoPy, car la mise en page peut être compliquée. Il est possible de créer une image consigne via PowerPoint.

Il existe aussi des composants **texte**, **image** et **forme**.

Position et taille des objets



(Résolution de mon écran : 1920 x 1080)

Utilisation d'une liste externe

Ceci servira lorsque l'on souhaite, par exemple, présenter un mot différent à chaque essai. La façon la plus élégante de le faire est de ne créer qu'une routine avec un seul objet texte et d'utiliser une liste externe. Pour cela, il faut préparer le ficher.

Le titre doit :

- Commencer par une majuscule.
- Pas de caractères spéciaux.
- Pas d'espaces.

Les colonnes contiennent les caractéristiques nécessaires à la présentation de vos stimuli et/ou des caractéristiques que vous souhaitez voir encodées dans votre fichier de résultats.

Chaque ligne correspond à un essai. Par exemple, s'il y a 3 conditions et 3 paramètres, il y aura donc 3*3 = 9 essais et donc 9 lignes, correspondant à toutes les situations possibles.

Étapes :

- *Objet texte* : il s'agira de faire référence à la colonne souhaitée dans Text et de placer le menu déroulant à Set Every Repeat. Il faudra écrire cela sou forme : $NomDeLaColonne.
- *Objet image* : il s'agira de faire référence à la colonne souhaitée dans Image et de placer le menu déroulant à Set Every Repeat. Il faudra écrire cela sou forme : $NomDeLaColonne. Dans la liste externe, il faudra la liste des mots à afficher **et** de la liste des noms de fichiers images à afficher.
- *Réponse Clavier* : il s'agira de faire référence à la colonne souhaitée dans Correct Answer et de placer le menu déroulant à Set Every Repeat. Il faudra écrire cela sou forme : $NomDeLaColonne.
- *Boucle* : il faut faire référence au fichier souhaité dans Conditions. Le nombre de conditions et de paramètres s'affichent en dessous une fois le ficher importé.
  À l'aide du champ Selected rows il est possible de choisir combien d'items de la liste seront présentés. Exemple : si nous entrons 0:3 seul les trois premiers items de la liste seront présentés. Quand il est laissé vide, PsychoPy comprend qu'il faut jouer tous les items de la liste externe.

Erreurs fréquentes :

→ NameError: name 'Item' is not defined : le nom de la colonne n'est pas le même que celui indiqué dans le composant. $Item ne sera pas considéré comme similaire à $Items.

→ 'Item' is not defined' : laissé constant au lieu de changer en set every repeat.

→ AttributeError: 'str' object has no attribute 'status'> : une entête de colonne de votre liste externe correspond à un nom déjà utilisé pour nommer un des objets dans une routine

Ajouter une réponse

Arrêt après un certain temps

Dans la fenêtre du stimulus, il faut choisir duration pour le champ stop et entrer un chiffre. Ce chiffre représente des secondes, signifiant que l'affichage du stimulus s'arrêtera au bout de x secondes.

Clavier

Allowed keys : ce champ désigne les touches de réponse sur lesquelles le sujet est autorisé à appuyer.
Store : correspond à ce que PsychoPy doit stocker comme touche de réponse appuyée :

→ First key : ne stock que le premier appui. Généralement, ceci est considéré comme étant le plus intéressant mais cela varie en fonction de ce qui est testé.
→ Last key : ne stock que le dernier appui.
→ All keys : stock tous les appuis.
→ Nothing : ne stock rien du tout.

Store correct : si cette case est cochée, alors PsychoPy va directement encoder si la réponse donnée est correcte. Pour pouvoir faire cela, il lui faut :

→ La touche de réponse appuyée par le participant (stockée grâce à Store)
→ La vraie réponse correcte qu'il fallait donner, indiquée dans la liste externe. Il faut alors compléter le champ : $RepCorrecte.

Attention : la touche enter correspond à return en anglais, il faut donc bien noter 'return'.

Clic de souris

End Routine on press : permet de choisir quel type de réponses sont valables :

→ Never : aucun clic ne sera valable
→ Any clic : tous les clics seront considérés comme valables
→ Valid clic : seulement les clics placés au bon endroit seront considérés comme une bonne réponse.

Dans le cas du valid clic, il faut nommer le stimulus valable dans Clickable stimuli $ afin que le programme sache quel stimulus considéré comme étant la bonne réponse.

Clé vocale

Cette fonction existe mais n'est pas explicitée dans le cadre de ce cours.

Cotation sur une échelle

Ratings

Slider :

- Size : permet d'indiquer la largeur et la hauteur voulues pour l'échelle.
- Position : permet d'indiquer le placement de l'échelle sur l'écran.
- Ticks : permet d'indiquer en combien de points doit être l'échelle. Exemple : (1, 2, 3, 4, 5)
- Labels : permet de donner un nom aux différents points. Soit on en donne 5, soit on ne donne que les extrêmes et le milieu et les deux intermédiaires sont laissés vides ('').
- Onglet Appearance : on indique le type d'échelle voulu (rating = échelle de Likert.
- TriangleMarker : permet d'avoir un (gros) triangle comme marqueur.

Sur l'onglet Basic du Slider, si vous cochez la case Force end of routine, il se produira la même chose qu'avec une souris : dès que le participant aura cliqué, PsychoPy passera à l'essai suivant. Pas besoin dans ce cas de demander au sujet d'appuyer sur une touche une fois son jugement fait.

Boucles et séquences temporelles

Cliquer sur Insert Loop dans le panneau Flow et sélectionner New. Il faudra pointer le début et la fin de la boucle. Une fenêtre de propriétés s'ouvrira alors.

Très souvent, il est bon de laisser un petit temps entre 2 essais (typiquement entre 500 et 1000 msecs) afin que les essais ne défilent pas trop vite : c'est ce qu'on appelle un **intervalle inter essai** (**ISI**). Il y a plusieurs façons d'introduire cela avec le Builder.
Ainsi, on peut, au lieu de spécifier le timing des premiers objets au temps t = 0 sec, de commencer à t = 0.5 sec (ce qui amène à ajuster le timing pour l'ensemble des objets).

Randomisation
Différents ordres aléatoires :

- *Ordre aléatoire* : hasard complet. Mais dans cela peut amener un sujet à donner plusieurs mêmes réponses à la suite, ce qui n'est pas nécessairement optimal.
- *Ordre aléatoire contraint* : contraindre le hasard et faire en sorte qu'il génère une suite aléatoire des stimuli, mais avec la contrainte qu'il n'y en ait pas plus de x essais à la suite d'une même catégorie.

Un ordre (aléatoire ou aléatoire contraint) peut également être :

- *Fixe* : même ordre pour tous les participants.
- *Variable* : varie pour chaque participant ou chaque groupe de participant.

Trials properties :

- LoopType : permet de choisir le type de randomisation voulu.

  Boucles avec méthode des constantes
  - Boucle sequential : un seul ordre est possible, les items sont présentés séquentiellement [a,b,c]. Si cela est répété 3 fois, le seul ordre possible est [a,b,c,a,b,c,a,b,c].
  - Boucle random : la randomisation n'est effectuée qu'à l'intérieur de chaque set. Si il y a trois set d'essais (abc), (abc) et (abc), il y aura donc (3!) x (3!) x (3!), donc 216 ordres possibles. Dans ce cas, il y aura donc toujours au moins deux essais 'a' avant l'apparition du troisième essai 'b'. Un ordre possible serait [b,a,c,c,b,a,c,a,b].
  - Boucle full random : Dans ce cas, il n'y a qu'un seul (abcabcabc), menant à 3^9 = 19683 ordres possibles. Cela signifie qu'une séquence de ce type est possible : [a,a,a,b,b,c,c,c,b]. **Remarque** : si nRep = 1, choisir random ou full random revient au même.

  Boucles avec méthode 'en escalier'
  - Staircase et interleaved staircase : ces boucles sont adaptatives, l'ordre et la nature des stimuli présentés varient en fonction des réponses des sujets.

Si nous avons besoin d'un ordre aléatoire fixe ou un aléatoire contraint fixe, il est conseillé de créer cet ordre en dehors de PsychoPy puis de l'intégrer à la liste externe. Dans ce cas, il faut demander au logiciel de présenter les essais dans un ordre séquentiel, c'est-à-dire les uns à la suite des autres tels qu'ils sont définis dans la liste externe, selon l'ordre fixe.

Si vous voulez utiliser un ordre aléatoire variable, c'est très facile de le faire dans PsychoPy.
Enfin, si vous voulez incorporer un ordre aléatoire contraint variable, le Builder propose quelques possibilités, mais assez restreintes.

Utilisation d'une liste externe dans une boucle

1. Choisir le Name, le LoopType et le nombre de répétition (nRep $).
   Attention : pour le nombre de répétition, il faut prendre en compte le nombre de conditions et de paramètres compris dans la liste externe. Par exemple, une boucle répétée 3 fois avec 4 paramètres et 4 conditions donnera 4 + 4 + 4 = 12 essais au total.
2. Choisir le fichier correspondant à la liste externe à placer dans Condition.
3. Faire référence aux colonnes de la liste externe pour chaque stimulus pertinent et pour chaque caractéristique pertinente.
   Attention : ne pas oublier de cocher Set Every Repeat dans le menu déroulant.
   Cas particuliers :
   → Le clavier : Store correct : si cette case est cochée, alors PsychoPy va directement encoder si la réponse donnée est correcte. Pour pouvoir faire cela, il lui faut :
   - Les touches que le participant doit appuyer inscrites dans Allowed Keys $.
   - La touche de réponse appuyée par le participant (stockée grâce à Store)
   - La vraie réponse correcte qu'il fallait donner, indiquée dans la liste externe. Il faut alors compléter le champ : $RepCorrecte.

   → Le cercle : pour créer un cercle, il faut choisir l'objet Polygon, puis regular polygon… dans Shape et spécifier 100 dans Num.vertices afin qu'il y ait 100 côtés au polygone.
   La couleur est à spécifiée dans l'onglet Advanced > Fill color : $Couleur.

Remarque : si vous avez changé certaines choses dans la liste externe au cours de l'expérience, il ne faut pas oublier de réimporter le fichier de la liste externe pour qu'il soit mis à jour dans PsychoPy également.

Feedback

Pour cela, il faut utiliser le composant Code dans la catégorie Custom des Components. Cette procédure implique d'utiliser des petites lignes de codes.

1. Créer une routine nommée Feedback qui doit être incluse dans la boucle des essais.
2. Créer les composants de la routine :
   → Un composant Code permettant de définir le Feedback.
   Remarque : le composant Code doit toujours être **avant** le composant Text.
   Dans Begin Experiment, on crée une variable contenant le message à afficher, par exemple : *msg*. Il peut être vide (*msg = '' ''*) ou contenir du texte (*msg='fb'*).
   Dans Begin Routine, on écrit les lignes du code pour définir ce que vaut la variable *msg* à chaque essai. C'est-à-dire qu'on on définit les conditions pour mettre à jour la variable en fonction de ce qu'a répondu le participant (copier/coller page 24-25 TP2) :

```
if reps2.corr==1:
    msg="BRAVO!"
    coul="green"
else:
    msg="erreur!"
    coul="red"
```

Le resp2 correspond au nom du composant clavier donné dans la routine d'essai (mieux vaut donner des noms simples et clairs pour pouvoir s'y retrouver et faire peu de fautes).
Erreur fréquente : output: SyntaxError: invalid syntax : erreur dans la ligne de code.

→ Un composant Text pour afficher le Feedback.
On y définit la taille, la position du message de feedback.
Dans le champ Text, il faut entrer $msg. Le signe $ indique que le message est variable à chaque essai et le mot msg renvoie directement à la variable créée.
Il faut aussi modifier le paramètre du composant Color : $coul.
Attention de ne pas oublier de changer constant en set every repeat !

Feedback pour certains essais
La procédure serait la même que précédemment, avec un petit changement dans Begin routine :

```
if resp.corr == 1:
   continue
else:
   msg = "erreur..."
```

Dans ce cas, on ne veut pas qu'un feedback soit donné dans le cas de réponses correctes. On veut simplement que le programme continue à s'exécuter.

Super-boucles

Cas où l'on souhaiterait randomiser les blocs d'essais. Les super-boucles sont des boucles nichées (*nested*) dans une autre :

1. Créez un fichier externe par bloc.
   Attention : faites en sorte qu'ils comportent exactement les mêmes colonnes !
2. Créez une 'super liste externe' contenant les deux listes externes correspondant à chaque blocs.
3. Créez **une seule** routine qui contient le détail d'un essai (qu'il soit congruent ou pas).
4. Créez une troisième liste externe (SuperListe) qui va lister les noms des listes externes des deux blocs.
5. Créer boucle de base (trials) : on souhaite qu'elle fasse référence au contenu de la troisième liste externe. Dans le champ Condition, il faut donc indiquer $Liste (nom de la colonne de la troisième liste externe : SuperListe).
   Cette première boucle permet de récupérer la colonne qui contient le nom des fichiers qui contiennent vos blocs.
6. Introduire une deuxième boucle (trials2) : une boucle par-dessus la première.
   Attention : n'oubliez pas de vous assurer que le champ de LoopType est bien sur Random pour que les deux listes (congruente, incongruente) soient randomisées ! Cela peut évidemment varier en fonction des expériences et il est possible de faire varier les types de randomisation dans chacune des deux boucles.
   Cette deuxième boucle permet de préciser la liste qui contient la colonne avec les noms des blocs.

Les super-boucles permettent également de ne pas répéter les routines qui auraient la même structure mais où seuls 1 ou plusieurs paramètres des composants changent.

Pauses

1. Insérer un routine nommée 'Break' ou 'Pause'.
   Attention : pour ne pas avoir de problèmes, insérez toujours cette routine **avant** la routine d'essai.
2. Compléter la routine avec les éléments souhaités (une consigne, un message, etc).
3. Ajouter un composant Code (à placer avant les autres composants créés !) et entrer le code suivant dans Each Frame :
   if trials_2.thisTrialN not in [2]:
   continueRoutine=False
   Ceci nous permet de dire au logiciel de ne pas lancer la pause tant que l'on n'est pas au deuxième essai. Avec 'trials_2' le nom de la boucle !

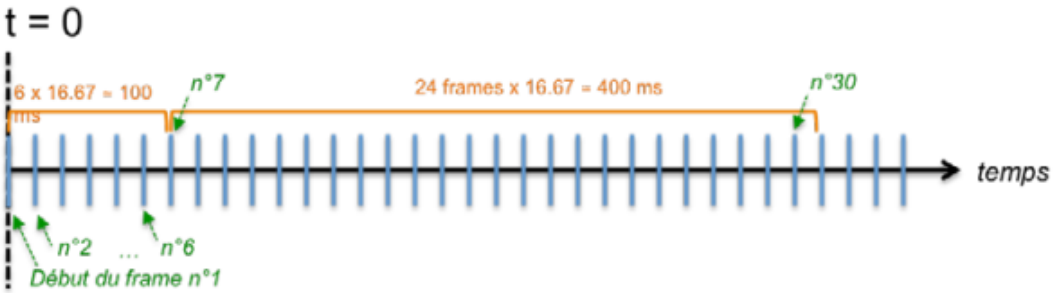
Principe de Frames

Les informations sur un écran d'ordinateur sont affichées selon une certaine fréquence de rafraichissement (taux de rafraichissement). La fréquence de rafraichissement (exprimée en hertz, Hz) d'un écran d'ordinateur ou d'un moniteur vidéo est définie par le nombre d'images s'affichant sur l'écran par seconde. Cette valeur varie généralement entre 60 et 200 Hz.
Résolution de mon écran : 60 Hz.

60 Hz : l'écran affiche une nouvelle image toutes les 0.01667 sec (1/60), soit encore 1 image toutes les 16.67 msecs (1000 msecs / 60Hz). Ainsi, 16.67 msecs est ce qu'on appelle la durée d'un cycle de balayage ou encore la durée d'un frame.
Cela veut dire que l'écran est en mesure d'afficher 60 images par secondes et que chaque images dure 16.67 msec.
Le nombre de frames est obtenu en divisant la durée voulue (100 msec) par la durée d'un frame (16.67 msec) : 100 / 16.67 = 6, cela correspond à 6 frames.

Petite particularité : l'ordinateur ne peut afficher quelque chose à l'écran que pendant un nombre entier de frames. Ainsi, afin que le timing soit le plus valide et précis possible, il est conseillé de prendre l'habitude de parler en nombre de frames.

Start
Soit nous démarrons au temps t=0 et PsychoPy se calera sur le début d'un frame, soit il faut indiquer le numéro du frame. Si nous souhaitons laisser un blanc de 500 ms avant (soit 30 frames), le numéro de frame (frame N) serait 31.



Stop

Il est possible de parler en nombre de frames (duration frames), soit en numéro de frame (frame N). Le nombre de frames spécifie la durée d'un stimulus en frames et le numéro de frame se base sur le schéma ci-dessus et spécifie le numéro de frame auquel il faut s'arrêter.

Spécifier la durée par une condition

Il est aussi possible de spécifier une condition pour préciser la temporalité d'un évènement. De cette façon, on est sûr que l'événement n+1 commencera bien après un nombre complet de frames écoulés pour l'affichage de l'événement n.

Condition doit être complété avec une ligne de code très simple : **$amorce.status==FINISHED** qui veut dire lorsque le statut de l'amorce est : 'fini'. Avec 'amorce' = le nom de l'évènement précédent.