



Manuel

TESTING EYE-TRACKING

Création : Janvier 2022 / Update : Mai 2022

Fabienne Chetail

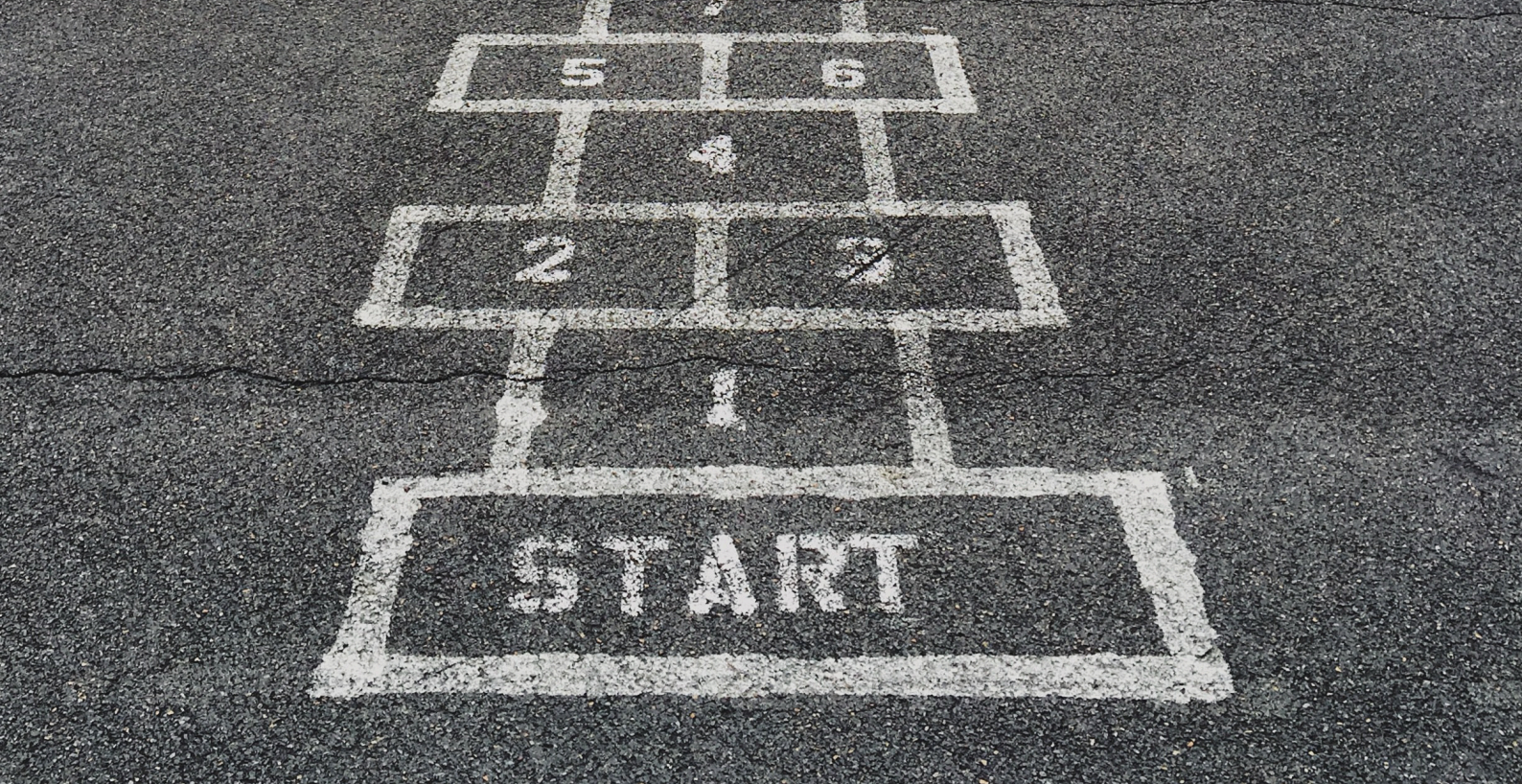


TABLE DES MATIÈRES

Préambule	6
1.1. Pourquoi ce manuel ?	6
1.2. La salle eye tracking du CRCN	7
Système Eyelink1000 au CRCN	8
2.1. Configuration générale	8
2.2. Éléments	9
2.3. Agencement précis des éléments	11
Testing	16
3.1. Critères d'exclusion des participants	16
3.2. Paramétrages en amont.....	16
3.3. Paramétrages avec le sujet.....	19
3.4. Lancement de la collecte de données	20
3.5. Paramètre optimaux études F. Chetail	24
Programmer une expérience avec Psychtoolbox.....	27
4.1. Création des fichiers préalables	27
4.2. Script expérimental.....	30
Analyser des données EyeLink	38
5.1. Logiciels et licence	38
5.2. Fichiers générés	38
5.3. Visualisation des données sous DataViewer.....	39
5.4. Analyse visuelle de la qualité des données	42
5.5. Exportation des données sous Data Viewer	47

5.6. Traitement des données sous R / RStudio.....	50
---	----





- Section 1 - **Préambule**

1.1. Pourquoi ce manuel ?

La base de ce manuel a été rédigée initialement dans le cadre d'une étude de lecture de textes avec enregistrement de mouvements oculaires en 2014 par Fabienne Chetail. En 2022, elle a décidé de mettre à jour les informations (avec l'aide de Margaux Genucchi) et surtout de les étendre, pour offrir une information plus complète de l'utilisation de l'EyeLink1000. Comme souvent, considérant que ce document comportait des informations utiles pour d'autres personnes souhaitant utiliser l'eye tracker, elle a décidé de mettre ce manuel à disposition de toutes. Gardez en tête néanmoins qu'un ensemble de paramètres expliqués dans ce document ont été ajustés dans le cadre des expériences de lecture de phrases conduites au LCLD et ne sont donc pas nécessairement adaptés à d'autres d'expériences !

De plus, les informations complètes sur le système et son fonctionnement se trouvent dans les différents manuels de l'EyeLink1000. Ces manuels sont très complets et dans les pages qui suivent nous y faisons de nombreux renvois :

- [Installation Guide](#) (installation du matériel et paramétrages de base)
- [User Manual](#) (paramétrages plus précis, utilisation de l'EyeLink lors d'une expérience)
- [Data viewer manual](#) (visualisation et traitement des données, lignes de code pour ajouter en ligne des informations dans les fichiers de données)
- [Programmer guide](#) (détail des commandes pour la programmation d'une expérience)
- [The EyeLink Toolbox](#) (Cornelissen_etal_BRMIC2002.pdf)
- [SR Research Experiment Builder User Manual.pdf](#) (manuel de programmation d'expériences via le logiciel de SR research – plutôt que Matlab par exemple donc).

Enfin, notez qu'au CRCN, bien que ce manuel puisse être transmis à des étudiant.es s'ils sont amenés à tester en eye-tracking (mémorant.es, stagiaires), seules les chercheur.es du CRCN sont autorisé.es à manipuler le matériel (e.g., changement de lentilles), à modifier les paramètres des ordinateurs (e.g., modification de physical.ini), et à utiliser des clés USB pour récupérer les données.

1.2. La salle eye tracking du CRCN

Comme expliqué dans le CRCN booklet, la responsable académique de la salle eye-tracking est Fabienne Chetail. Avant toute utilisation de la salle, vous devez l'informer de votre projet et recevoir les informations essentielles de sa part. La lecture de ce manuel est obligatoire avant d'utiliser la salle. De même, si vous projetez de modifier la configuration de la salle, de déplacer certaines parties du setting ou de modifier des fichiers sur l'un des ordinateurs, la responsable de la salle doit en être avertie.



- Section 2 -

Système Eyelink1000 au CRCN

2.1. Configuration générale

L'eye tracker que nous avons est un Eyelink 1000 (société SR Research). Notre système comporte une Desktop mount camera (voir p.23 de l'installation Guide pour une présentation détaillée du système et des autres). La Figure 1 montre la façon dont l'ensemble des éléments du système sont installés dans la salle New Eye Tracking.



Figure 1. Eléments du système d'eye tracking installés dans la salle New eye tracking du CRCN

La liste des différents éléments composants le système est ci-dessous :

- a = host computer
- b = display computer (mac mini)
- c = desktop mount camera
- d = fibre optique
- e = mentonnière (chinrest)
- f = écran du host computer
- g = écran du display computer

Cette configuration ne doit pas être changée, sauf en cas de pré-test par l'expérimentateur.trice, où iel joue le rôle de sujet en même temps. La Figure 2 présente une configuration possible dans ce cas-là.



Figure 2. Modification possible de la configuration en cas de tests par l'expérimentateur.trice (PI ou chercheur.e uniquement)

2.2. Eléments

2.2.1. Host computer

Le host computer ne doit en aucun cas être modifié par les utilisateurs/trices (pas d'installation de programmes ou autre, pas d'effacement de fichiers, excepté les fichiers de résultats). Il est connecté au secteur, à la caméra (fibre optique/câble ethernet) et à un écran. Ce host computer a un [dual boot](#) (deux démarrages possibles sur des disques durs différents). Après avoir appuyé sur le bouton de démarrage, l'ordinateur vous proposera donc de choisir entre les deux boots suivants : le boot EyeLink et le boot Windows (Figure 3).

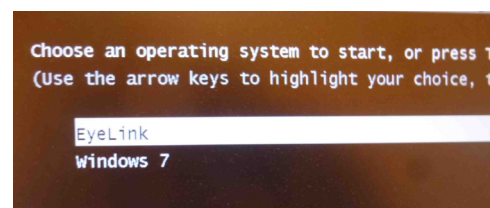


Figure 3. Dual boot sur le host computer

Le boot Eyelink permet de démarrer un enregistrement avec l'eye tracker (Figure 4). Si vous démarrez Eyelink sans avoir branché l'alimentation de la caméra, le [DOS](#) se lance et donne le prompt [ELCL\EXE>](#). Dans ce cas, branchez la caméra, puis tapez [T+Enter](#) dans la console. Eyelink se lancera. Pour quitter la partition eyelink, assurez-vous d'avoir cliqué sur 'Exit Eyelink' (après avoir éventuellement choisi 'exit set up'), puis appuyez sur le bouton de démarrage. A tout moment, vous pouvez quitter la session avec la combinaison [CTRL+ALT+Q](#). Une fois que l'ordinateur revient sous le DOS, appuyez sur le bouton de démarrage pour éteindre la machine (Figure 5).

Le boot Windows permet entre autre de modifier certains paramètres et de récupérer les données enregistrées. Pour quitter la session windows, procédez comme sur n'importe quel PC.

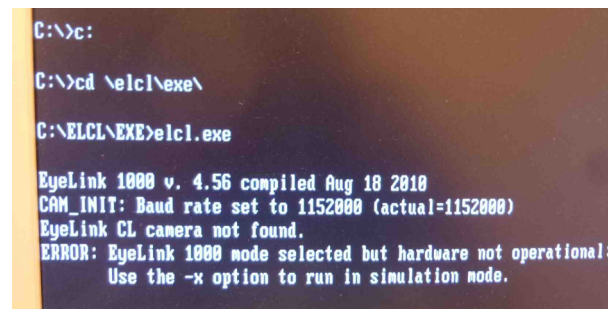


Figure 4. Boot Eye link

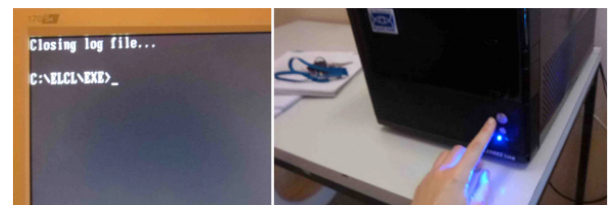


Figure 5. Quitter le boot Eyelink

2.2.2. Display computer

Le display computer peut être n'importe quel ordinateur. Jusqu'ici, une machine est principalement utilisée (mac mini qui initialement a été acheté dans le cadre d'un projet Leybaert & Colin). Sur cet ordinateur commun, les fichiers des utilisateurs doivent être rangés de façon appropriée afin que ce ne soit pas le bordel : création d'un dossier à votre nom dans ~/Documents où vous déposerez tout ce qui vous concerne (pas de fichier qui traînent sur le bureau par exemple !). Début 2022, un nouveau mac mini a été financé par F. Chetail, le précédent se révélant trop ancien pour supporter la dernière version de Matlab.

2.2.3. Liens entre le host computer et le display computer

En mode Link, l'eye tracker peut être contrôlé par le PC Display via des commandes envoyées par la liaison Ethernet. Les applications Matlab (via Psychtoolbox) et Experiment Builder notamment permettent de programmer des scripts expérimentaux pour faciliter l'interaction avec le host computer. Un scénario courant consiste à faire en sorte que l'application sur le PC Display contrôle l'eye tracker pour démarrer la configuration et la calibration du sujet, tandis que l'opérateur utilise le clavier du PC hôte EyeLink pour surveiller et contrôler à distance la collecte des données, effectuer les drift correction et gérer les problèmes éventuels (plus de détails dans la Section 4).

2.2.4. Desktop mount camera

La Figure 6 présente les différents éléments de la camera eye tracker. Pour brancher et débrancher la caméra, connectez et déconnectez le câble le plus en haut (attention, ne confondez pas avec les câbles des yeux droit et gauche, Figure 9). Une fois branchée, on entend un petit 'clac' de l'illuminateur et une LED orange s'allume derrière l'appareil. Attention, la caméra ne doit pas rester branchée quand le testing est terminé. Veillez donc à l'éteindre



Figure 6. Eléments de la caméra eye-tracker

quand vous avez terminé votre session de testing, en débranchant simplement le câble. Débranchez également le transformateur (pour cela, débranchez la prise blanche notée α sur la Figure 1). La caméra peut être utilisée pour un tracking mono- et bi-noculaire (cf. p.22 de l'Installation guide pour un montage en binoculaire). Dans le type d'expérience conduites au LCLD, c'est un tracking monoculaire qui est utilisé. Avant utilisation, vérifiez que les câbles au derrière de la caméra pour l'œil droit et gauche sont branchés.

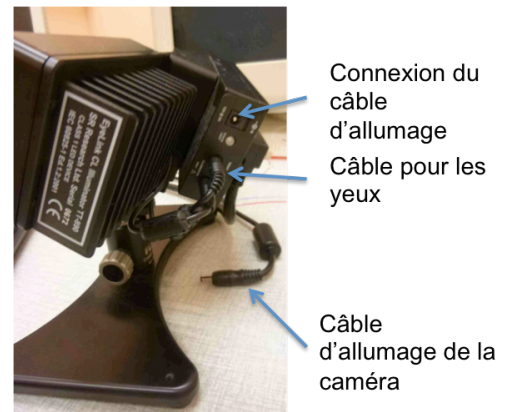


Figure 9. Câbles de la caméra

Le câble bleu connectant la caméra au host computer est une fibre optique (aussi appelée câble ethernet, Figure 7), donc fragile et très chère. Cette fibre permet de transmettre les enregistrements des mouvements oculaires du display au host computer. Veillez à ne jamais marcher dessus. Notez par ailleurs que dans les branchements, il y a un sens entre le host computer et le display computer. La partie du câble se terminant par une étiquette doit être connectée au display computer.



Figure 7. Fibre optique de la salle New eye tracking

2.2.6. Chinrest

La mentonnière de la compagnie SR Research est utilisée, pour plus de confort et de précision (Figure 8). Lors de la mise en place d'une expérience, elle doit être réglée pour correspondre au positionnement des éléments (cf. Plus loin). Comme pour tous les autres éléments du système, ne faites pas n'importe quoi avec... : comme celle-ci a été correctement disposée, si votre participant.e n'est pas bien positionné.e dans la mentonnière, modifiez plutôt la hauteur de la chaise avant de toucher aux réglages de la mentonnière. Nous reviendrons plus précisément sur les dispositions physiques des éléments dans la section suivante.



Figure 8. Chinrest du CRCN

2.3. Agencement précis des éléments

2.3.1. Distances physiques entre les éléments

Durant la collecte des données, certain éléments du système doivent être placés de façon spécifique afin d'augmenter la précision des mesures récoltées, ou tout du moins, afin de ne pas la diminuer (cf. p.25 Installation guide et Figure 10). Les points suivants sont particulièrement importants :

- La caméra doit être sur une table, avec une distance de 50 à 55 cm entre le menton du/de la participant.e et la caméra (actuellement, la distance est de 52 cm).
- La caméra doit faire face au/à la participant.e et la vis de serrage doit être alignée avec le centre de l'écran du display computer.

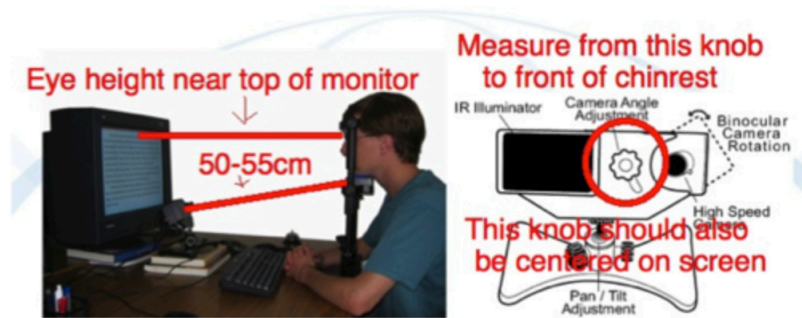


Figure 10. Distances à respecter lors de l'agencement des éléments pour les participants

- A noter que la caméra change de configuration (vis noire centrale qui se décale) pour l'enregistrement binoculaire (Figure 11).
- Le haut de l'illuminateur doit être aussi près que possible que la partie basse du moniteur (si besoin un réhausseur peut être utilisé) et aussi haut que possible sans masquer l'écran.
- La chinrest doit être centrée horizontalement avec l'écran.
- Idéalement, l'appui-front de la chinrest doit être à la même hauteur que le haut de l'écran ou à $\frac{1}{4}$ du top de l'écran.



Desktop Mount (Level)



Desktop Mount (Angled)

Figure 11. Utilisation de la caméra en mode binoculaire (à droite)



Figure 12. Distances à respecter lors de l'agencement des éléments pour les participant.es

Pour aider le positionnement des éléments, des marques noires et bleues (papier collant) ont été faites sur la table et la colonne du display computer (Figure 12). Avec cette configuration, la/le participant.e se trouve à 61 cm de l'écran (mesure de l'appui-front au top screen) et à 67 cm (mesure de l'appui-front au bottom screen). La camera se trouve à 52 cm du/de la participant.e (de la vis noire au menton) et à 52,5 cm lorsqu'on mesure de l'oeil droit et 53,5 cm lorsqu'on mesure de l'oeil gauche. Notez que l'écran peut être légèrement penché vers l'arrière si besoin (voir p.25 Installation Guide).

Important ! Par «écran» ou «screen» ou «surface d'affichage», on entend ici uniquement la partie bleue du moniteur sur l'image ci-contre (c'est-à-dire sans inclure les bords noirs en plastique de l'écran).



Figure 13. Notion d'écran (ou surface d'affichage)

2.3.2. Encodage des distances et longueurs physiques

Il est important que les informations enregistrées sur le host computer soient cohérentes avec celle de la configuration physique de la salle. Pour cela, vous pourriez être amené.e à modifier les fichiers

PHYSICAL.INI et de **FINAL.INI** du host. Mieux vaut toujours modifier les données dans **FINAL.INI** que dans les autres, puisque l'eye tracker va d'office utiliser les paramètres modifiés dans **FINAL.INI**, même s'ils sont différents dans les autres paramètres indiqués dans les fichiers .INI (dans notre cas concret : si **PHYSICAL.INI** indique une configuration différente de **FINAL.INI**, ce seront les paramètres affichés dans **FINAL.INI** qui seront considérés). Attention, ces modifications ne peuvent être faites que par des chercheur.es (pas des étudiant.es) et après accord avec la/le responsable de la salle.

Chaque fois que vous modifiez votre configuration physique (par ex., si vous utilisez un nouveau moniteur, si vous modifiez la distance entre les yeux et l'écran, etc.), vous devez vérifier que le fichier **PHYSICAL.INI** reflète toujours votre configuration. Notamment, les fichiers doivent être modifiés si :

- vous changez d'écran de testing
- vous changez la résolution de l'écran
- vous modifiez les distances entre les éléments (chinrest, caméra, display screen)

Pour accéder aux fichiers, démarrez la partition Windows sur le host computer. Allez ensuite dans le **disque dur eyelink**, puis dans le dossier **elcl/EXE**. Les fichiers s'ouvrent dans un éditeur de texte. Les parties à éventuellement modifier sont les chiffres dans les trois lignes ci-après :

- *screen_phys_coords* : spécifie la distance physique entre les quatre bords de la surface de présentation et le centre de l'écran. L'ordre de ces mesures est le suivant : gauche, haut, droite, bas, et elles sont spécifiées en millimètres (ici : -187.0 150.0 187.0 -150.0)
- *screen_pixel_coords* : spécifie la résolution de l'écran. Normalement, au début d'une expérience, l'API de EyeLink 1000 ajuste ce paramètre de manière automatiquement, de sorte que vous n'avez pas besoin de le modifier manuellement (cf. Installation manuel p.54). Si l'API n'est pas utilisé, la valeur par défaut de ce paramètre s'applique. Pour un écran avec une résolution de 1280 x 1024 pixels, il faut entrer (0.0 0.0 1280.0 1024.0)
- *screen_distance* : spécifie la distance depuis le haut et le bas de l'écran jusqu'à l'œil du/de la participant.e, respectivement (ici : 610 670)

Le détail des procédures pour déterminer ces chiffres est donné p. 54 de l'Installation manual.

Si vous êtes plusieurs à tester en même temps, en utilisant des configurations différentes, je vous conseille de vous prévenir les un.es les autres de tout changement éventuel du fichier. Si vous êtes seul.e à tester durant une longue période, avant toute première utilisation, vérifiez les données stockées dans ce fichier.

Si vous envisagez de modifier les paramètres par défaut des fichiers .INI, veuillez copier et coller les commandes cibles dans le fichier **FINAL.INI** et effectuer la modification dans ce fichier pour faciliter toute manipulation ultérieure. (traduction p. 44 du USER MANUAL). En effet, pour une configuration donnée, il est recommandé de copier toutes les commandes ou tous les paramètres que vous souhaitez modifier dans le fichier **FINAL.INI**, qui sera le dernier fichier de configuration à être traité par l'eye tracker et qui remplacera donc les paramètres figurant dans les autres fichiers .INI. Cette conception permet d'éditer facilement un seul fichier, de garder la trace des modifications apportées, de faciliter la mise à jour du logiciel (il suffit de conserver les paramètres dans le fichier **FINAL.INI**) et de faciliter le dépannage.

Si vous lisez le manuel d'installation fourni par EyeLink, on vous recommande d'utiliser un fichier nommé **TRACK.EXE** pour mesurer les distances et longueurs physiques. Cependant, notez qu'il n'est pas disponible sur Mac et donc inutilisable sur le display computer installé dans le laboratoire du CRCN. A la place, faites tourner le script MATLAB **exemple.m** situé dans le dossier MATLAB accessible via le raccourci sur le bureau du display computer. C'est un script qui permet de lancer une expérience de lecture de textes (voir détails à la Section 4). Il vous permet de tester rapidement l'EyeLink avant de vous

lancer dans votre propre expérience. Pour pouvoir mesurer `screen_phys_coords`, lancez le script `exemple.m` (après avoir indiqué le numéro du ou de la participant.e dans MATLAB). L'expérience va alors débiter par une explication adressée à un.e participant.e fictif.ve. Ensuite, les procédures de calibration et de validation se lancent. Une fois que vous vous trouvez sur l'écran de calibration, un point central s'affichera. Vous pouvez alors à partir de celui-ci réaliser vos mesures.

2.3.3. Notion de degrés d'angle

Très souvent avec un eye-tracker, on ne parle pas de mesures juste en centimètres, mais aussi en degrés d'angle. Si vous aviez besoin de paramétrer votre set up en considérant des degrés d'angle (parce que par exemple ce sont ces mesures qui sont rapportées dans un article sur lequel vous vous basez), voici comment les calculer à partir des distances métriques.

Si votre écran a une résolution de 1280 x 960 pixels, pour une taille physique réelle de 406 x 304 mm, alors il est possible de calculer la taille physique d'un stimulus à l'écran dont on connaît la taille en pixel. Une image de largeur 100 pixels fait 31.72 mm sur l'écran. Si l'on veut calculer le degré d'angle concernant la largeur de cette image, voici la situation (vue du haut sur la Figure 14) :

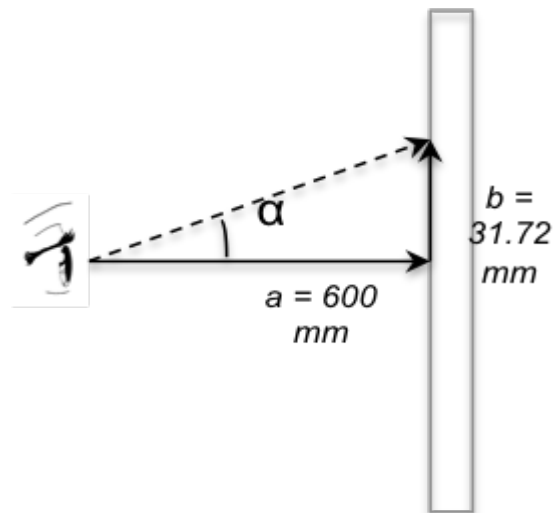


Figure 14. Situation trigonométrique pour le calcul de degrés d'angle

Comme $\tan \alpha = b / a$, on a $\tan \alpha = 31.72 / 600 = 0.053$. En prenant l'inverse de la tangente, on obtient 3.034° d'angle (faites attention à ce que votre calculatrice utilise des angles en degrés et non en radians !). Voici un convertisseur conseillé par SR Research : <http://www.oocities.org/robertellis600/va.html>.

2.3.4. Choix et changement de lentille de la caméra

Avec le système EyeLink 1000, nous disposons de 3 lentilles de taille différente : 16, 25 et 35 mm (Figure 15). Elles sont placées dans l'une des deux caisses sous la table du display computer. Ces lentilles sont à manipuler avec précaution et les caches doivent être conservés avec soin dans leur pochette. Seuls les chercheur.es du CRCN peuvent effectuer un changement de lentilles, pas les mémorant.es ou stagiaires ! Dans les manuels de l'eyelink, il est recommandé d'utiliser la lentille 35 mm avec notre système (cf. p.14 Installation guide). Plusieurs chercheur.es du CRCN ont néanmoins noté qu'ils obtenaient de meilleurs résultats avec la lentille 25 mm. Pour ma part, c'est avec 35 mm que j'ai obtenu les meilleurs résultats. Selon votre étude, il peut donc être nécessaire de faire vous-même des tests. Si vous souhaitez changer la lentille pour la première fois, demandez à l'un.e des responsables de la salle de vous montrer comment faire.

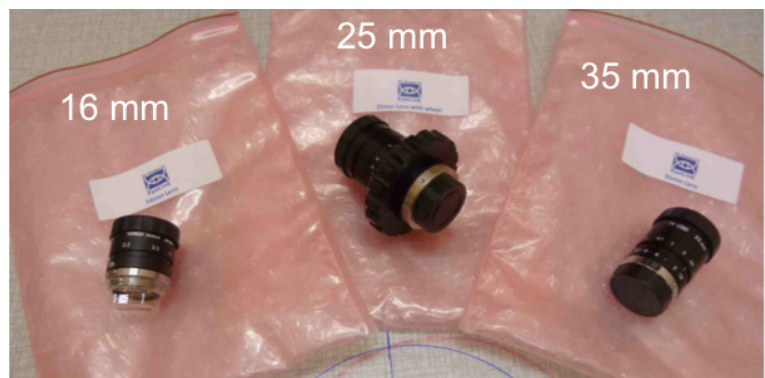


Figure 15. Les trois lentilles de la caméra disponible au CRCN



- Section 3 -

Testing

3.1. Critères d'exclusion des participant.es

Outre les critères d'inclusion/d'exclusion propres à votre expérience, sachez qu'il est déconseillé (voir impossible) de tester des participant.es qui ont :

- des lunettes => la calibration est parfois difficilement possible et/ou très peu précise
- des lentilles => elles peuvent produire des reflets sur la lentille de la caméra, bien que généralement le phénomène soit moins fort que pour des lunettes
- du maquillage autour des yeux => le maquillage contient des composants métalliques qui eux aussi brouillent la calibration. Demandez à vos sujets de venir sans maquillage ! (si besoin, il peut y avoir une bouteille de démaquillant dans la salle).

3.2. Paramétrages en amont

Le logiciel du host computer pour régler l'acquisition de données contient divers écrans, par lesquels on doit passer et on peut facilement naviguer. La Figure 17 représente de façon schématique les passages entre les différents écrans.

3.2.1. Allumage du système

Avant toute chose, vérifiez que les éléments sont bien en place (cf. Section 2), notamment que la chinrest et la caméra sont bien positionnées sur les marques. Branchez alors le câble de la caméra (et l'alimentation des ordinateurs si nécessaire) et retirez le cache de la lentille.

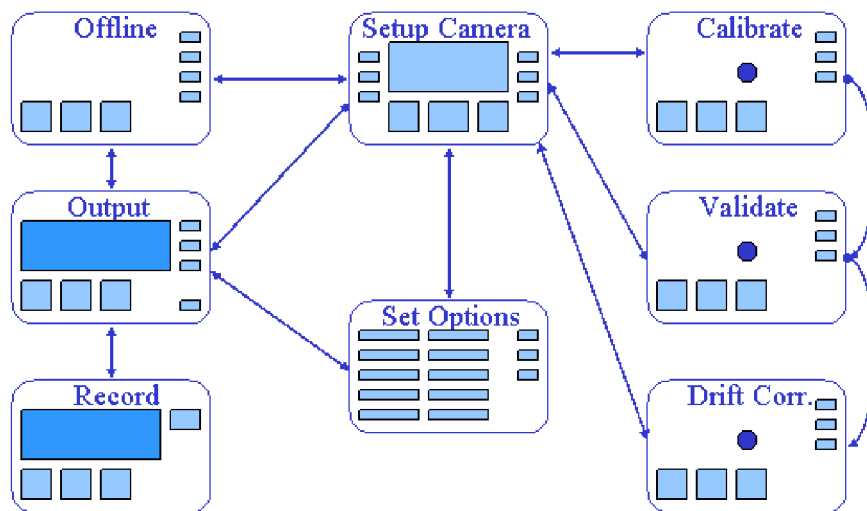


Figure 17. Relations entre les différents écrans du host computer lors de la mise en route d'une collecte de données

Allumez ensuite l'ordinateur en bootant sur Eyelink. Eyelink doit être démarré au moins 5 min avant de commencer une expérience (pour éviter des petits drifts des thresholds). Vous tombez alors sur l'écran OFFLINE ou potentiellement sur le menu CAMERA SET UP (Figure 16). Dans les deux cas, choisissez dans le menu à droite 'Set options'.

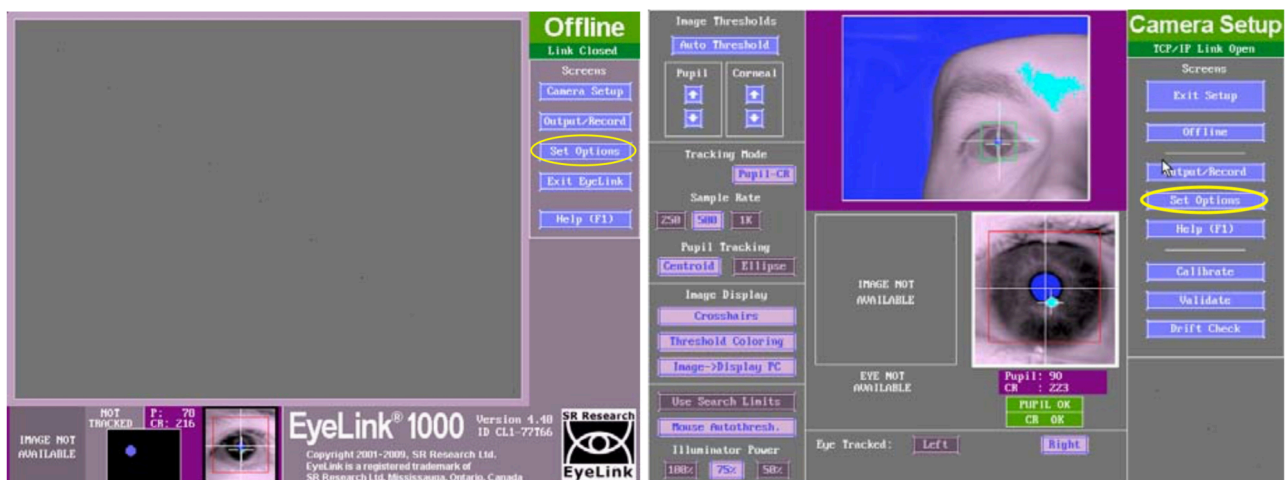


Figure 16. Ecran Offline (gauche) ou CameraSetUp (droite) sur lequel on tombe par défaut quand on démarre le boot EyeLink

Note : Si vous cliquez sur Exit Eyelink, cela permet d'éteindre l'ordinateur (une fois que l'ordinateur revient sous le DOS, appuyez sur le bouton de démarrage pour éteindre la machine). De plus, à partir de n'importe quel écran, la combinaison de touches "CTRL+ALT+Q" permet de quitter l'application Eye-Link Host.

3.2.2. Définition des options

Une fois que vous avez cliqué que «set option», vous tombez sur la page représentée à la Figure 16. C'est ici que vous allez définir des paramètres essentiels pour votre collecte de données. Quand un rectangle est coloré en rose-violet, cela veut dire que cette option a été activée.

Cet écran est divisé en différentes parties :

1. Partie calibration. Choisissez une calibration se fait sur 9 points, avec un pacing interval de



Figure 18. Panneau de configuration des différents options possibles (accessible par « Set Options »)

1000 ms, en ordre randomisé, avec le premier point répété. Trois ou cinq points sont insuffisants pour bien calibrer dans des expériences de lecture, 13 n'apporte pas grand chose.

2. Partie configuration. La configuration sélectionnée doit être **Desktop (level), Monocular, 35 mm lens, Stabilized head, Camera level, ILLUM Cam**. Cela correspond à notre système de caméra et à une lentille de 35 mm (si vous n'avez pas changé la lentille avec celle de 25 mm!). Si ce n'est pas la configuration active par défaut, allez dans 'Select config' et sélectionnez-la. Si cette configuration n'apparaît pas dans la liste, cliquez sur 'load defaults'. Si cela ne fonctionne toujours pas, redémarrer EyeLink après avoir débranché l'alimentation de la caméra. Au moment où le prompt apparaît, rebrancher la caméra. Eye link se lance. Refaites alors les actions précédentes (select config / load defaults).
3. Partie tracking. **AREA** doit être sélectionné, afin d'enregistrer l'aire de la pupille (et non juste le diamètre)
4. Partie Event and data processing. **Gaze, normal, Extra, and STD** doivent être sélectionnés. **Gaze** est la position de la pupille en coordonnées (x,y) de l'écran, alors que **HREF** correspond aux coordonnées (x,y) en référence à la tête du participant. **Normal** est adapté aux études en lecture (alors que High est plutôt réservé à des études en psychophysique et/ou de petites saccades doivent être détectées). Extra : filtre maximal, pour réduire le bruit dans le signal.
5. Partie Compress EDF files. .edf est le format des fichiers de données. Ici, sélectionnez toutes les options. C'est plus que ce dont on a besoin, mais il vaut mieux trop que pas assez.
6. Partie Recording data view. Gaze cursor et gaze doivent être sélectionnés. Cela permet de voir sur le host computer les mouvements oculaires du participant pendant l'expérience.

Une fois les options activées, vous n'avez plus besoin de les vérifier tant que vous n'éteignez pas le host computer. Cela veut dire qu'il vous suffit de faire une seule vérification/modification en début de chaque session de testing. Une fois que c'est fait, choisissez CAMERA SET UP dans le menu à droite.

3.2.3. Définition des paramètres de la caméra

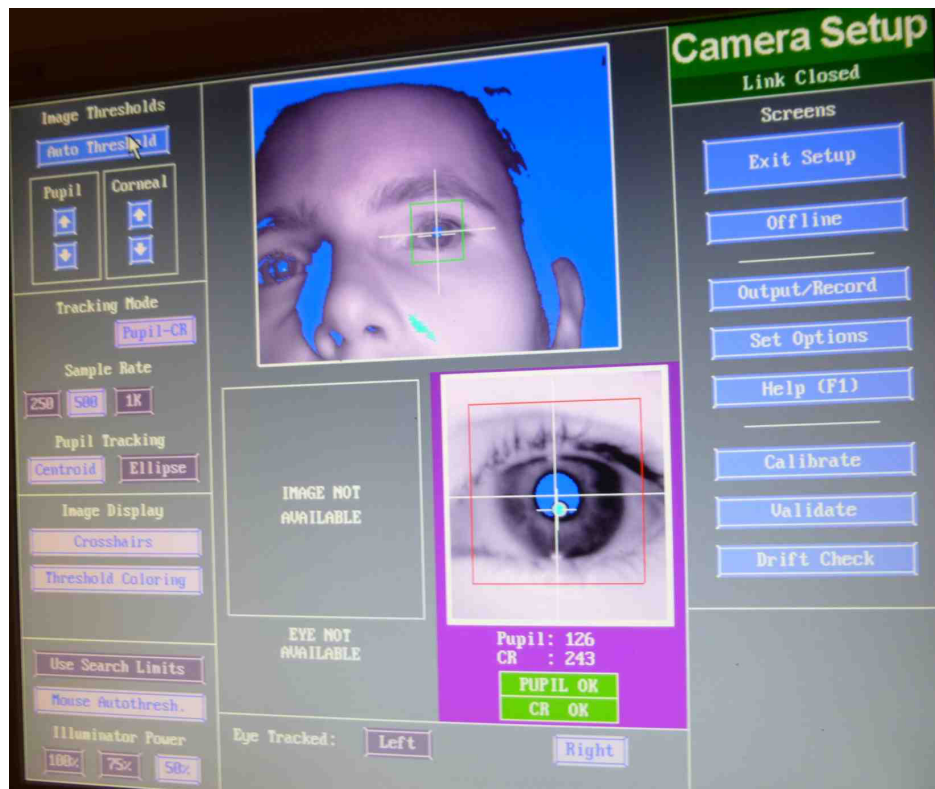


Figure 19. Ajustements généraux dans Camera Set up

Même si votre sujet n'est pas encore là, vous pouvez faire les premiers ajustements de la caméra, tels qu'ils sont représentés sur la Figure 17.

Notamment :

- Tracking mode. Il doit être sur **Pupil-CR**, avec un **pupil tracking centroid** (cela permet d'avoir le moins de bruit dans le signal). Le taux d'échantillonnage (**sample rate**) doit être sur **500**. Cela signifie que la position de l'œil sera enregistrée toutes les 2 ms (choisir 250 voudrait dire toutes les 4 ms et choisir 1000 voudrait dire toutes les 1 ms, cela divise par deux ou double la taille du fichier de données respectivement).
- Image display. **Crosshairs** and **threshold coloring** doivent être activés. Cela permet de dessiner les fixations sur les images du contenu montré à l'écran du sujet.
- Mouse autothreshold. Ce paramètre doit être activé.
- Illuminator power. Il est très important que ce paramètre soit sur **50%**. Dans le User manual, il est conseillé d'utiliser 75%, mais les travaux précédents dans le CRCN ont montré qu'avec notre configuration, 50% donne de meilleurs résultats.

De même que précédemment, une fois que ces options sont activées, vous n'avez plus besoin de les vérifier tant que vous n'éteignez pas le host computer. Cela veut dire qu'il vous suffit de le faire une seule fois en début de chaque session de testing.

Cliquez sur 'Exit Setup' si vous voulez revenir à l'écran visité avant l'écran "camera Setup".

3.3. Paramétrages avec le sujet

Lorsque le sujet est là, faites les vérifications d'usage (par ex. carte étudiant dans le cadre de SONA), collectez les metadata, assurez-vous qu'il/elle n'est pas maquillée, ne porte pas de lunettes ou de lentilles (il vaut mieux avoir vérifié cela en amont). Passez ensuite à la partie eye-tracking.

La première chose à faire est de tester l'œil dominant du sujet. Pour cela, demandez-lui de fixer une croix d'un seul œil. Si l'œil qui reste ouvert est le droit, alors on considère que c'est lui qui est dominant.

1. Faites asseoir le participant de façon à ce qu'il soit positionné de façon confortable sur la chinrest (ni trop haut, ni trop bas). Ne modifiez pas l'emplacement des éléments ou l'ajustement de la chinrest pour cela, mais dites au sujet de régler la hauteur de la chaise.
2. Dans Camera Setup, vérifiez que l'œil sélectionné est le bon (le droit, si c'est l'œil dominant).
3. Demandez à votre participant de se tenir immobile en regardant le centre de l'écran. Cela va vous permettre d'ajuster plusieurs paramètres:

- Position. L'œil tracké dans la fenêtre du haut (cf. Figure 17) doit être plus ou moins au centre de la fenêtre (l'autre œil doit être en dehors de la fenêtre ou près du bord). Si ce n'est pas le cas, faites quelques ajustement de la camera (possibilité d'ajuster l'inclinaison de la caméra). Les yeux doivent être plus ou moins alignés avec la ligne médiatrice horizontale de l'écran (cf. Figure 18).
- Focus. Vérifiez le focus de la caméra, en tournant directement le support de la lentille (Figure 19). En théorie, cela nécessite d'être ajusté uniquement en début de session.

- Détection de la pupille. Si la pupille est détectée, un cadre vert et deux croix (une grande et une petite) doivent apparaître sur l'image de l'œil (cf. Figures 18 et 19). La zone de la pupille détectée est en bleue. Si aucune pupille n'est détectée, les cadres **pupils** et **CR** (sous l'œil zoomé, Figure 17) sont en rouge (vous pouvez voir cela en demandant à votre sujet de fermer les yeux). Le petit point jaune montre la réflexion de la cornée (CR). Demandez au sujet de regarder à différents coins de l'écran pour s'assurer que la détection de la pupille reste correcte et que le CR n'est pas perdue.

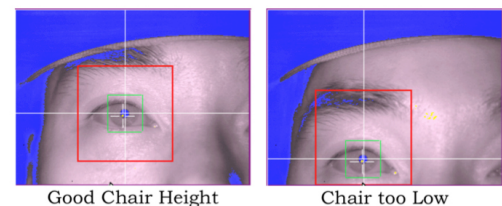


Figure 20. Exemple de bon et mauvais centrage de l'œil

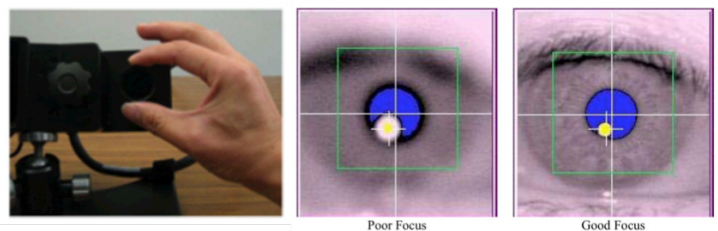


Figure 21. Réglage du focus de la caméra

- Pupil thresholds. Quand la détection de l'œil est stabilisée, cliquez sur '**Auto Threshold**' (en haut à gauche). Les valeurs sous l'image zoomée de l'œil sont automatiquement ajustées pour la pupille (**pupil**) et le reflet de la cornée (**CR**). La qualité du threshold s'évalue en regardant l'image de l'œil zoomé (Figure 20). Si le threshold est trop haut, le bleu ne recouvre pas toute la pupille (Figure 20, gauche). Si le threshold est trop bas, il y a des ombres au bord de l'œil et de la cornée (Figure 20, droite). Si c'est nécessaire, vous pouvez ajuster manuellement le threshold (à utiliser avec parcimonie !), tout en demandant au sujet de regarder à différents coins de

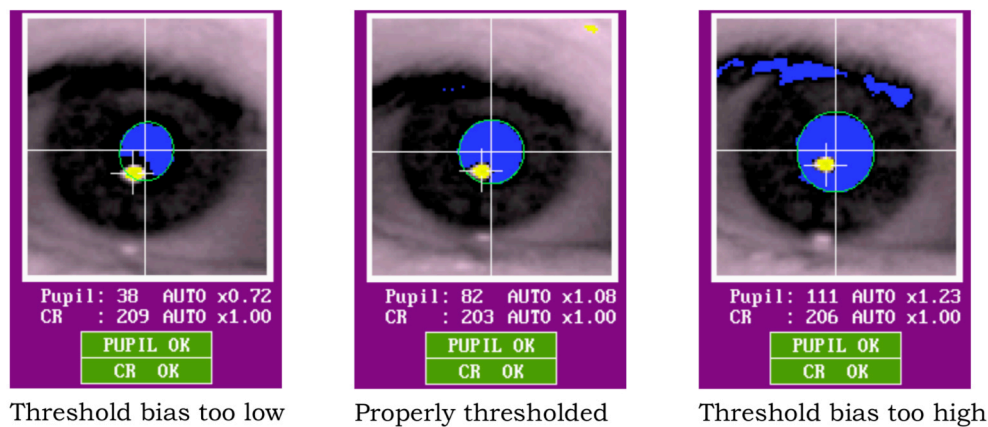


Figure 22. Trois configurations différentes de seuils pour la détection de la pupille et de la cornée

l'écran pour s'assurer que la détection de la pupille reste correcte. Selon le [User manual \(p.69\)](#), le pupil threshold devrait être entre 75 et 110, et le CR threshold ne devrait pas excéder 230. Si le pupil threshold est trop bas, le manuel suggère d'augmenter l'illuminator output. Si le pupil ou CR threshold est trop haut, le manuel suggère de diminuer l'illuminator output.

Une fois ces réglages effectués, l'étape suivante est la calibration suivie de la validation. Elles peuvent être initiées à partir de l'écran Camera Set up, mais il est plus aisé de l'inclure dans le script Matlab de l'expérience.

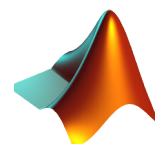
3.4. Lancement de la collecte de données

Une fois les paramètres correctement ajustés, ouvrez le logiciel Matlab sur le display computer, en cliquant une fois sur l'icône dans la barre du bas. A noter que l'EyeLink est également interfaçable avec l'application PsychoPy, bien que ça n'ait pas encore été fait dans la salle New Eye Tracking.

Dans le menu déroulant de Matlab (gauche), cliquez sur le dossier à votre nom, puis le cas échéant dans le sous-dossier qui contient votre expérience. Dans ce manuel, un script d'exemple sera utilisé, accessible dans le dossier «EXEMPLE». Double-cliquez sur le script Matlab correspondant (fichier qui se termine par .m). L'expérience se lance et vous devez rentrer les metadata concernant le participant (ici, simplement un chiffre pour indiquer le numéro de participant, Figure 21).

3.4.1. Calibration, validation, drift correction

Quand vous validez l'entrée du numéro de participant (en appuyant sur «entrée»), Psychtoolbox se lance (voir Section 4). L'écran de calibration/validation apparaît alors sur le display computer (Figure 22). A partir de là, votre sujet doit garder sa tête immobile, potentiellement jusqu'à la fin de l'expérience.



a) Calibration

La calibration (p. 72 User manual) sert à déterminer la correspondance entre la position de la pupille sur l'image de la caméra et la position du regard sur l'écran. C'est une calibration en 9 points qui est utilisée ici. En voici les différentes étapes :

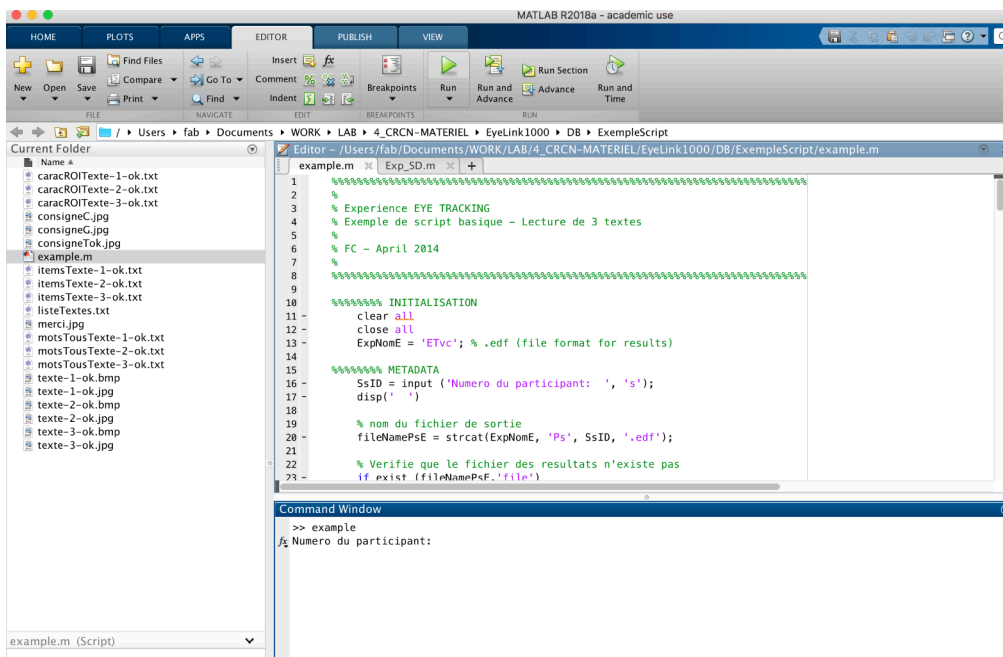


Figure 23. Fenêtres Matlab à l'ouverture du script

- Expliquez la consigne à votre participant : *Vous allez voir un gros point blanc au centre de l'écran avec un petit point noir à l'intérieur. Fixez bien le petit point noir. Ce point va bouger à différents endroits de l'écran. Vous devez le suivre du regard et quand il est immobile, fixez-le, sans anticiper sa position ultérieure.*
- Appuyer sur C sur votre clavier (host computer) pour lancer la calibration.
- Un point apparaît sur le display computer, accompagné d'un signal sonore. Le participant le fixe. Quand la détection de la pupille est stable, cliquez sur **Accept fixation** sur le host computer. Le point se déplace alors aux différents endroits. A ce moment-là, vous n'avez rien à faire car le programme accepte lui-même les autres points de fixation. Pendant ce temps-là, regardez bien sur le host computer si la pupille et la réflexion de la cornée continuent d'être correctement détectées.
- La calibration est réussie si l'ensemble des points forme un carré et que les points sont correctement alignés (Figure 23).
- Important ! Tant que le résultat n'est pas optimal, cliquez sur **Restart** pour que le sujet recommence. Cette étape est cruciale, car cela va déterminer la précision des mesures ! Il faut généralement faire minimum deux calibrations par participant.
- Si le résultat est bon, cliquez sur **Validate** depuis le host computer.

b) Améliorer la calibration ?

Comme dit précédemment, la qualité des calibrations détermine l'utilité des données enregistrées et la précision du calcul du regard. Voici des procédures simples pour améliorer cette étape:

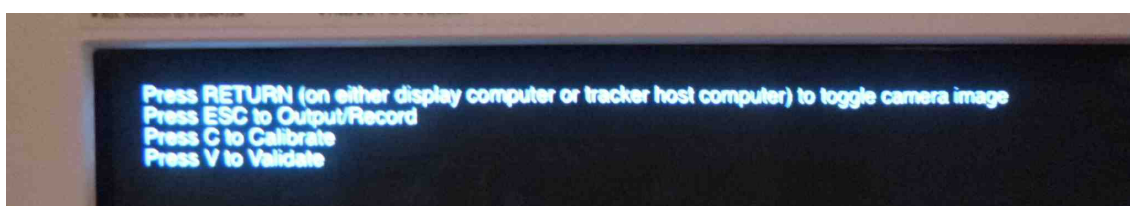


Figure 24. Fenêtre permettant de lancer la calibration / validation

- La zone pupillaire seuil doit se trouver à l'intérieur de la boîte rouge autour de la pupille (cf. Figure 17) lorsque le sujet regarde n'importe quelle zone de l'écran. Si une partie de la pupille sort de cette boîte, la pupille sera perdue.
- Le reflet cornéen ne doit jamais être perdu ou mal identifié lorsque le sujet regarde autour de la zone calibrée.
- Demandez toujours au sujet de regarder aux quatre coins de l'écran après avoir effectué le réglage de la caméra. Assurez-vous que le signal de la pupille et du CR n'est pas perdu lorsque le sujet le fait.
- Les sujets qui n'ont jamais été calibrés auparavant doivent s'entraîner à fixer de manière stable les cibles d'étalonnage. Essayez d'effectuer au moins deux calibrations par sujet avant de commencer à collecter les données.

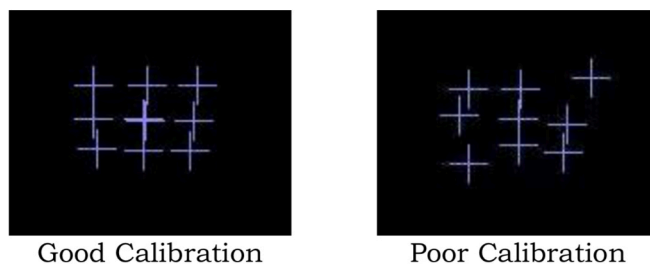


Figure 25. Résultats potentiels de la calibration

- Lorsque vous écrivez vos propres applications, essayez de faire correspondre la couleur de fond de l'écran pendant l'étalonnage et la validation à celle des écrans de test.
- Les changements de taille de la pupille causés par de grandes différences de luminosité peuvent dégrader la précision du système.

c) Validation

Le validation (p.76 User manual) intervient après la calibration et permet de valider la calibration en donnant une mesure précise de l'exactitude du système à prévoir la position du regard sur l'écran.

- La procédure est la même que pour la calibration : le participant a neuf points à fixer et c'est vous qui déclenchez la séquence en cliquant sur **Accept fixation**.
- Après cela, les neuf points sont présentés sur l'écran du host computer avec l'erreur de l'estimation de la position du regard (en ° d'angle) (Figure 24). Dans le carré vert en bas à droite de l'écran, le système rapporte l'erreur maximale et l'erreur moyenne, avec une appréciation (GOOD ou BAD).
- La recommandation de SR Research est d'accepter les calibrations avec un maximum de déviance de 1.00° d'angle et une moyenne maximale de 0.50°. Pour les études en lecture avec F. Chetail, le maximum doit être de 0.80° d'angle (avec une moyenne inférieure à 0.50°). Cela signifie que même si la conclusion de l'eye tracker est 'GOOD', cela peut ne pas être valide pour l'expérience. Faites notamment attention à ce qu'il n'y ait pas un biais systématique dans une direction sur les 9 points.
- Si la performance est mauvaise, sur juste un ou deux points, le participant a pu être inattentif. Dans ce cas, refaites la validation (**Restart**). Si les résultats ne sont pas meilleurs après cela, c'est que la calibration n'était pas suffisamment bonne, et il faut donc refaire cette étape. Pour cela cliquez sur 'Accept'. Vous revenez à l'écran de choix sur le display computer. Sélectionnez **Calibrate** en tapant sur la touche C.
- Si la performance est bonne, cliquez sur **Accept**. Vous revenez à l'écran de choix (Figure 22). Appuyez alors sur **ESCAPE** pour commencer l'expérience.

d) Drift correction

Le drift correction affiche une seule cible au participant et mesure ensuite la différence entre la position de fixation calculée pendant l'étalonnage et la cible actuelle. Cette étape permet simplement de signaler l'erreur de fixation calculée. Si vous trouvez que l'erreur est grande, vous pourriez vouloir refaire l'étape de calibration/validation. Vous pouvez aussi incorporer un drift correction au début de chaque essai de votre expérience (voir section 4), pour s'assurer que la précision est toujours bonne. C'est particulièrement utile si vous avez l'impression que le sujet a bougé la tête et que vous voulez tester si les mesures collectées par l'eye tracker sont toujours correctes.

3.4.2. Poursuite du script d'expérience

Une fois l'étape de calibration/validation terminée, le script Matlab continue d'être lu. La consigne apparaît sur l'écran. Laissez le participant la lire, puis récapitulez avec lui des points importants de la consigne. Si vous avez inclus une étape de Drift checking, prévenez votre participant qu'avant chaque essai, il devra de nouveau fixer un point central (1 seul point). Vous pouvez également inclure dans votre script de refaire une calibration/validation à n'importe quel moment de l'expérience (voir section 4). En effet, bien que vous donniez à votre sujet la consigne de ne plus bouger du tout la tête une fois la calibration terminée, il se peut par exemple qu'il éternue en plein milieu de l'expérience et que donc il retire sa tête de la chinrest. Dans ce cas, il faut refaire une calibration/validation avant l'essai suivant.

Lorsque l'expérience est terminée, le script sur termine et l'on revient sur l'interface de base de Matlab sur le display computer. Vous n'avez rien à faire sur le host computer (enregistrement automatique du fichier .edf).

Si la session de testing est terminée, quittez Matlab et éteignez l'ordinateur. Sur le host computer, un bref message indiquera que les données ont été enregistrées. Choisissez ensuite **Exit set up**, puis **Exit eyelink**, puis appuyez sur le bouton de démarrage.



Figure 26. Exemple de résultat d'une validation. Les points représentent les lieux à fixer sur l'écran, les croix représentent la position de la pupille quand ces lieux étaient fixés, les chiffres représentent l'écart (en degrés d'angle) entre les coordonnées des lieux sur l'écran et les coordonnées de la pupille détectés lorsqu'elle fixait ces lieux

3.5. Paramètre optimaux études F. Chetail

En Février 2022, dans le cadre d'une étude de lecture de texte, l'ensemble des paramètres ont été testés pour définir la configuration permettant la meilleure précision des données. En voici le résumé :

OPTIMAL PARAMETERS

Distances : [à indiquer]

Position chinrest : [marques table + réhausseur]

Position écran : [marques table]

Lentille : [25/35]

Background: [essayer avec noir sur fond blanc]

Illuminator power : [50/75]

Eye : [left / right]



- Section 4 -

Programmer une expérience avec Psychtoolbox

Cette section est à destination des chercheurs en priorité. Tout est présenté sur base d'expériences de lecture de textes / de phrases.

4.1. Création des fichiers préalables

Lors d'une expérience de lecture de textes ou de phrases, les essais sont affichés sur l'ordinateur expérimental sous forme d'image. Il faut donc générer ces images. De plus, pour chaque zone d'intérêt potentielle (typiquement, chaque mot), il faut définir les coordonnées de la région d'intérêt (ce que appelle ROI pour *region of interest*).

Un exemple de fichiers complets pour expliquer la génération des fichiers est donné sur base du dossier de l'expérience de référence «22-02_THE-M_manipET» (sous dossier *Matériel*). Attention, le but ici n'est pas de détailler chaque fichier (ou de déboguer les scripts en fonction des besoins de chacun), mais bien de partager les grandes lignes de ma façon de faire pour celles et ceux que ça pourrait intéresser :-). Aussi, si vous ouvrez tel quel les fichiers, il y a toutes les chances pour que ça ne fonctionne pas directement (problème de chemins d'accès, de fichiers présents, de packages absents,...). A vous de simplement vous inspirer du code et de la procédure si cela vous tente...

4.1.1. Préambule important sur le type d'écran

Pour des raisons pas tout à fait éclaircies, cela fait une grande différence de préparer les fichiers images à présenter sur un **écran rétina** ou non : le calcul de la taille des caractères sous Matlab ne donne pas la même chose (sous retina, des décimales sont produites et cela fait bugger l'eyelink), le calcul des coordonnées (notamment des rectangles ROIs) n'est pas exactement le même (cf. Script3 ci-dessous), et enfin, les images produites ne semblent pas tout à fait identiques (non seulement, il faut multiplier par deux les

dimensions avant la génération sous écran retina cf. ligne 62 du script 5 – pour ensuite réduire les dimensions– mais en plus, elles semblent avoir un « quelque chose » de différent, qui fait bugger le script expérimental).

Pour la manip citée ici, tout avait été généré sous Retina initialement, puis à force de bugs, tout a été re-généré sous l'ordi et l'écran de la salle eye-tracking.

4.1.2. Grandes étapes de génération des fichiers

1. **Copier-coller du ou des textes dans "exTexte.txt"**. Soit c'est un texte continu, auquel cas copiez tout d'un coup (éventuellement en insérant des espaces insécables), soit c'est une suite de paragraphe ou de phrases que vous allez afficher (un par écran) et dans ce cas chaque phrase/paragraphe occupe une ligne dans le fichier texte.
2. **Extraction de tous les caractères présents dans le(s) texte(s) ou phrases à présenter.** Cela va permettre de calculer ensuite la largeur (width) exacte de chaque stimuli, et donc de calculer ses coordonnées x/y pour le positionner sur l'image ainsi que les coordonnées du rectangle définissant la ROI qui lui est associée. Pour cela, il s'agit d'utiliser le script R : `script1_extractLetters.R`. L'output sera `Oscr1_signTexts.txt`.
3. **Calcul de la 'pixel width' pour chaque caractère, en fonction d'une certaine police et d'une certaine taille d'écriture.** La flexibilité apportée ici permet de présenter n'importe quel texte en n'importe quelle police et en n'importe quelle taille. C'est un script Matlab qu'il faut utiliser ici (malgré mes efforts et ceux d'Alain, nous n'avons pas réussi à faire ces calculs sous R) et à cette étape il faut être sur un écran non Retina : `script2_computeLetterSize.m`. L'output est `Oscr2_signSize.txt`.
4. **Calcul du découpage du texte en lignes et écrans avec les coordonnées de chaque mot et les coordonnées de chaque ROI.** Le script `script3_fittingTexts.R` est assez complexe mais fait très bien le job (et il est très commenté). Les outputs sont `Oscr3_textCuttingCoordinates.txt` (que l'on va utiliser tout de suite après) et aussi `Oscr3M_coordinates.txt` et `Oscr3M_words.txt` qui peuvent être utiles pour le lancement d'une expérience.
5. **Marquage des items cibles dans les textes.** Il s'agit ici de coder la présence de mots cibles dans l'entière des textes. Dans la façon light de le faire (indispensable, mais généralement liée au fait qu'il n'y a justement pas de mots cibles spécifiques), il faut ouvrir le fichier `Oscr3_textCuttingCoordinates.txt` dans Excel, ajouter la colonne « target ». S'il y a des mots cibles par screen, il faut mettre manuellement une croix dans la colonne « target » pour ces mots cibles (laisser vide sinon) et enregistrer le tout sous `Oscr3_textCuttingCoordinates_targets.txt`. S'il y a des mots cibles à identifier et qu'il serait trop fastidieux de les marquer à la main, il faut juste créer la colonne « target » et enregistrer le doc comme dit précédemment. Quel que soit le cas de figure, il faut ensuite faire tourner le script `script4_markingTargets.R` (et des lignes sont indiquées si besoin pour inclure le marquage des mots cibles, à partir du fichier « targets.txt » par exemple – attention, ce dernier ajout provient de l'expérience 22-03_THE_A_manipET et non 22-02_THE-M_manipET). Les outputs sont de trois ordres, multipliés par le nombre d'écrans (par ex. si l'expérience contient 100 essais, avec 1 essai = 1 écran, il y a aura à chaque fois 100 outputs de chaque catégories) : `caracROI_screen_x.txt` (coordonnées des ROIs de chaque mot), `caracTargets_screen_x.txt` (codage des cibles éventuelles : f – pour filler– si ce n'est pas un mot cible), `words_screen_x.txt` (mots à afficher).

6. **Génération de toutes les versions d'images (jpg, bmp, avec et sans ROI).** Il faut pour cela utiliser le script `script5_creatingPictures.m`, sur un écran non retina (et idéalement de la même résolution que l'écran expérimental). Ces images sont générées sous Matlab car il m'a été impossible de clairement gérer les tailles de police sous R. Les images de chaque écran sont générées telles que `screen_1.bmp`, `screen_1.jpg` et `screen_1_roi.jpg`.
7. **Génération d'images .jpg avec masquage des items cibles (optionnel).** Cette procédure permet de potentiellement supprimer un biais de l'expérimentateur lors de l'inspection visuelle avant les analyses (cf. Section 5). Le script à utiliser est XXXX et les outputs sont XXXX.

STEP 1 : Sous MATLAB, génération des images de texte avec mots cibles cachés

- créer un dossier contenant :
 - o Les textes used pour l'expérience (noms de fichiers exacts)
 - o e.g., `caracROITexte-1.txt` etc..
 - o le script `extractionItemCibles.m`
 - o le script `TEXTEok_bis.m`
- Run `extractionItemCibles.m`
 - o Cela crée des fichiers textes du style : '`caracROITexte-1-ok-CRITIQUE.txt`' => extrait les coordonnées des ROI des mots cibles
- Run `TEXTEok_bis.m`
 - o Cela crée les images des textes avec un masque blanc sur les mots cibles => nommées e.g., '`texte-1-e1.jpg`' (le nom doit être exactement le même que celui que portaient les images lorsqu'elles étaient chargées pendant l'expérience)

4.1.3. Fichiers finaux utiles

Une fois tous vos fichiers générés, voici ceux dont vous pourriez avoir besoin pour faire tourner votre expérience :

- Dans un sous-dossier «`caracTexte`», tous les fichiers générés au point 5 précédent.
- Dans un sous-dossier «`pictureTexte`», tous les fichiers générés au point 6 précédent.
- Dans le dossier racine (où se trouve votre script expérimental), la liste de vos écrans à affichés (par ex. `texte_listeScreens.txt`), votre fichier de questions et de réponses aux questions le cas échéant, vos fichiers de consignes.

Dans la partie suivante, nous allons décrire un script expérimental simplifié (dossier `ExempleS-`

[cript](#)), qui n'a pas été conçu nécessairement selon ces étapes au niveau des fichiers préalables. Néanmoins, à des visées de compréhension, je rends disponible aussi le script expérimental correspondant à l'expérience «22-02_THE-M_manipET» afin que la manière d'organiser les fichiers d'input et d'output soit transparente

4.2. Script expérimental

Dans le script exemple, l'expérience consiste à présenter trois textes, les uns à la suite des autres. Les participants doivent simplement les lire et appuyer sur «espace» pour passer de l'un à l'autre. Ce type d'expérience est ici programmé avec Matlab, même si l'Eye link dispose d'une interface pour programmer des expériences.

Avec Matlab, un scénario courant consiste à faire en sorte que l'application sur le PC Display contrôle l'eye tracker pour démarrer la configuration et la calibration du sujet, tandis que l'opérateur utilise le clavier du PC hôte EyeLink pour surveiller et contrôler à distance la collecte des données, effectuer les drift correction et gérer les problèmes éventuels. Une toolbox sous Matlab a été développée pour permettre la communication entre la psychtoolbox (PTB) et l'EyeLink (voir article de Cornelissen, Peters, & Palmer, 2002).

Ci-dessous, les éléments principaux à ajouter dans un script Matlab-PTB sont présentés, afin que le display computer communique avec le host computer et la caméra. Ici, les commandes présentées fonctionnent sous Mac. Pour les commandes sous Windows et pour de nombreux autres détails, consultez le *Programmer guide*. Pour le script complet, voir le script nommé [exemple.m](#).

4.2.1. Informations préliminaires

Dans les exemples ci-après utilisant les commandes Psychtoolbox, [w](#) fait référence à la fenêtre psychtoolbox. DE plus, pour faciliter la visualisation des données sous Data Viewer, il faut que les stimuli soient présentés sous forme d'image (y compris lorsqu'il s'agit de texte). Dans le cas de texte, si vous souhaitez contrôler exactement la position des mots, générez les images comme expliqué dans la section 4.1.

4.2.2. Code à intégrer avant le lancement des essais

- **Enregistrer les détails de l'eye tracker.** Le but ici est de fournir à EyeLink des détails sur l'environnement graphique et d'effectuer certaines initialisations. Les informations sont renvoyées dans une structure qui contient également des valeurs par défaut et des codes de contrôle utiles (par exemple, le bit d'état du tracker et les valeurs des clés EyeLink).

```
el = EyeLinkInitDefaults(w);  
el.backgroundColor = black;  
el.foregroundcolour = white;  
el.msgfontcolour = white;  
el.calibrationtargetcolour = white;  
EyeLinkUpdateDefaults(el); % pass the values back to the EyeLink
```


- **Initialiser l'eye tracker.** Cela permet l'initialisation de la connexion avec le Eyelink Gazetracker. Il y a un exit du programme si ça échoue.

```
if EyelinkInit() ~= 1
    fprintf('Eyelink Init aborted.\n');
    cleanup; % cleanup function
    return;
end;
```

- **Ouvrir un fichier pour enregistrer les données eyelink**

```
Eyelink('OpenFile', 'test.edf');
```

- **Ajuster la résolution correcte d'enregistrement et le spécifier le fichier de données.** Il s'agit de définir le contenu du fichier EDF à l'aide des commandes `file_sample_data` et `file_event_filter`, et de définir les données de liaison par le biais des commandes `link_sample_data` et `link_event_filter`

```
Eyelink('Command','screen_pixel_coords = %ld %ld %ld %ld', 0, 0, width-1, height-1);
Eyelink('Message', 'DISPLAY_COORDS %ld %ld %ld %ld', 0, 0, width-1, height-1);
Eyelink('command','link_sample_data = LEFT,RIGHT,GAZE,AREA,GAZERES,HREF,PUPIL,STATUS,INPUT')
Eyelink('command','file_sample_data = LEFT,RIGHT,GAZE,HREF,AREA,HTARGET,GAZERES,STATUS,INPUT');
Eyelink('command','file_event_filter = LEFT,RIGHT,FIXATION,SACCADE,BLINK,MESSAGE,BUTTON,INPUT');
Eyelink('command','link_event_filter = LEFT,RIGHT,FIXATION,SACCADE,BLINK,MESSAGE,BUTTON,INPUT');
```

- **Inclure les étapes de calibration et validation.** Ce sont les commandes qui lancent les deux étapes à partir du script (et non du host computer).

```
EyelinkDoTrackerSetup(el)
```

4.2.3. Code à intégrer pendant les essais (dans la boucle)

Voici quelques commandes à ajouter à l'intérieur d'une boucle d'essais (avec `i` correspondant à l'identification d'un essai). Chaque essai doit avoir une paire de commandes "StartRecording" et "StopRecording" ainsi que des messages d'intégration au fichier de données (message pour marquer l'heure des événements critiques et l'information sur l'image/la zone d'intérêt/la condition de l'essai).

- **Inclure un drift correction avant chaque.** C'est un moyen facile de s'assurer de la qualité de la calibration en cours de route. Si le résultat est mauvais (parce que par exemple le sujet aurait bougé la tête), alors on peut décider de relancer une calibration/validation.

```
EyelinkDoDriftCorrection(el)
```

- **Marquer le début d'un essai**

```
Eyelink('Message', 'TRIALID %d', i);
```

- **Charger l'image correspondant à un essai.** Cela consiste à Envoyez un message d'intégra-

tion pour qu'une image puisse être chargée comme arrière-plan de superposition lors de l'analyse du Data Viewer. Ce message peut être placé n'importe durant l'essai (c'est-à-dire après le message 'TRIALID' et avant 'TRIAL_RESULT'). Plus d'informations sont données dans la section "Protocole d'intégration EyeLink Data to Viewer -> Commandes d'image" du manuel d'utilisation d'EyeLink Data Viewer.

```
Eyelink('Message', '!V IMGLOAD FILL %s', imageNameTestDV{i})
```

- **Transférer l'image de l'essai.** Cela permet de voir l'image vue par le participant sur le host computer. L'image doit être au format .bmp

```
Eyelink('Command', 'set_idle_mode'); % mode off line
Eyelink('Command', 'clear_screen 0');
Eyelink('command', 'draw_box %d %d %d %d 15', 0,0,width-1,height-1);
transferimginfo=imfinfo('image.bmp');
Eyelink('ImageTransfer',transferimginfo.FileName,0,0,0,0,0,0,1);
WaitSecs(0.1);
```

- **Démarrer l'enregistrement des mouvements oculaires.** Cela correspond au début de l'enregistrement de la position des yeux (précédé d'une courte pause pour que l'eye tracker puisse terminer la transition de mode). Les paramètres de la commande 'StartRecording' contrôlent la disponibilité des [file_samples](#), [file_events](#), [link_samples](#), [link_events](#).

```
Eyelink('Command', 'set_idle_mode');
WaitSecs(0.05);
Eyelink('StartRecording');
WaitSecs(0.1);
Eyelink('Message','SYNCTIME'); % mark zero-plot time in data file
```

- **Arrêter l'enregistrement**

```
Eyelink('Message', 'BLANK_SCREEN');
Eyelink('StopRecording'); % stop recording of eye-movements for the current
trial
Screen('FillRect', w, el.backgroundcolour); % Clear the display
Screen('Flip', w);
WaitSecs(0.1); % adds 100 msec of data to catch final events
```

- **Envoyer les ROIs.** Il s'agit ici d'envoyer des messages d'intégration nécessaires à l'analyse des données, notamment des informations sur les zones d'intérêt pour l'essai (cf. section "Protocole pour l'intégration des données EyeLink au visualiseur-> Commandes de zone d'intérêt" du manuel d'utilisation du visualiseur de données EyeLink). Important : il ne faut pas envoyer trop de messages dans un laps de temps très court ou le tracker EyeLink pourrait ne pas être en mesure de les écrire tous dans le fichier EDF. Pensez à ajouter un court délai tous les quelques messages.

```
Eyelink('Message', '!V IAREA RECTANGLE %d %d %d %d %d %s', i, 100,200,500,
220,'cible') % ici la ROI correspond à un rectangle de 400 pixels de long
et 20 de haut
```

- **Enregistrer le nom des variables et des conditions (pour DataViewer).** Il s'agit ici d'en-

voyer des messages pour inclure des informations sur les conditions de l'essai. Chaque message peut être une paire de variable de condition d'essai et sa valeur correspondante suivant le message à jeton '**!V TRIAL_VAR**' (voir la section "Protocole pour l'intégration EyeLink Data to Viewer-> Commandes de messages d'essai" du manuel d'utilisation de EyeLink Data Viewer). J'ai tendance à minimiser l'utilisation de cette commande, et à ajouter toutes les conditions et autres lors du traitement dans R.

```
WaitSecs(0.001);  
Eyelink('Message', '!V TRIAL_VAR type %d', 2) ;  
Eyelink('Message', '!V TRIAL_VAR imgfile %s', 'pingouin.jpg') ;
```

- **Marquer la fin d'un essai.** Cette commande envoie le message "TRIAL_RESULT" pour marquer la fin d'un essai dans Data Viewer. Ce message est différent du message END de fin d'enregistrement qui est enregistré lorsque l'enregistrement de l'essai se termine. Data viewer n'analysera pas les messages, événements ou échantillons qui existent dans le fichier de données après ce message.

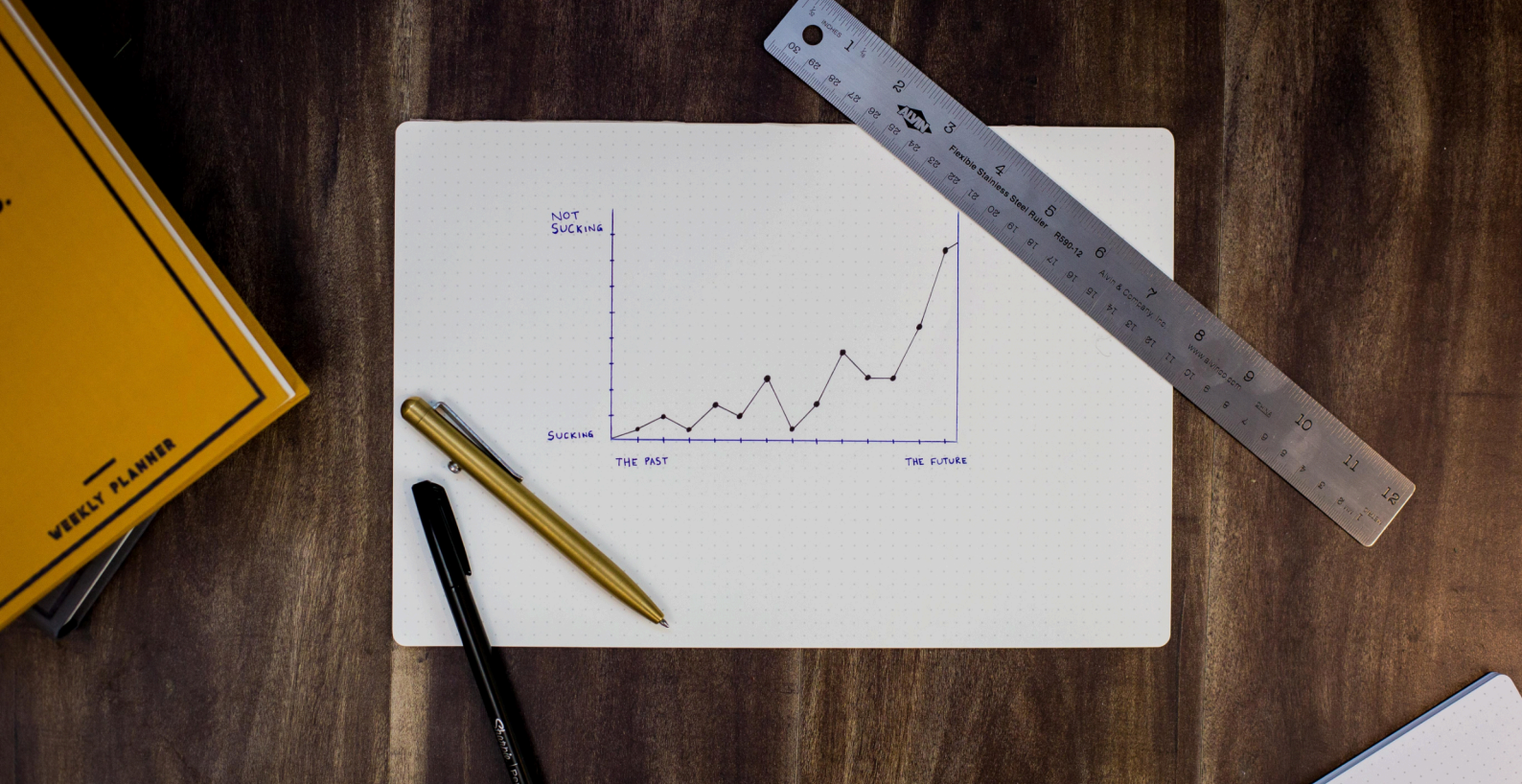
```
Eyelink('Message', 'TRIAL_RESULT 0')
```

4.2.4. Code à intégrer après les essais

- **Fermer la connexion Eyelink et recevoir le fichier de données sur le display computer.** Attention, le nom du fichier ne doit pas être trop long ! (pas plus de 8 caractères)

```
Eyelink('Command', 'set_idle_mode');  
WaitSecs(0.5);  
Eyelink('Closefile');  
Eyelink('ReceiveFile', 'test.edf', 'test.edf');  
Eyelink('Shutdown');
```

Voici pour les commandes essentielles. Pour plus de détails, il s'agit de se référer aux manuels de programmation d'EyeLink. Pour un exemple simple reprenant l'ensemble des commandes mentionnées, faites tourner le script [exemple.m](#). Pour un exemple plus complexe, avec quelques variations et l'intégration des fichiers préalables présentés dans la section 4.1, vous pouvez consulter le fichier [exp_M_1_2022_03_v6.m](#).



- Section 5 -

Analyser des données EyeLink

5.1. Logiciels et licence

Toutes les données enregistrées par l'eye tracker sont stockées sous forme d'un fichier .edf. La lecture de ce fichier nécessite l'utilisation du logiciel DataView, qui permet la visualisation et le traitement des données. Il permet de plus d'extraire les données pour des analyses ultérieures sous R par exemple. L'application DataView peut être téléchargée ici : www.sr-support.com/thread-7.html.

Pour pouvoir utiliser les fonctions complètes du logiciel, vous devez avoir une licence. Au CRCN, cette licence est associée à une clé USB ('dongle', Figure 25), stockée dans le bureau de F. Chetail. Lors de la première utilisation du dongle, il faut activer la licence sur votre ordinateur (www.sr-support.com/thread-332.html). Si vous n'avez pas de licence, vous pouvez aussi ouvrir vos fichiers avec le logiciel sans le dongle, mais dans ce cas, vous êtes averti.e que vous n'avez qu'une permission d'utilisation restreinte de logiciel (mais suffisante pour visualiser les données des cinq premiers essais).



Figure 27. Dongle contenant la licence pour DataView

5.2. Fichiers générés

A la fin d'une expérience, plusieurs fichiers sont générés et vont vous permettre de traiter les données :

- **Fichier Eyelink .edf** : c'est LE fichier qui contient les données d'eye tracking facilement exploitables. Ce sont des données qui sont déjà pré-traitées par l'Eyelink en fonction par exemple des ROIs, des paramètres de fixation, etc. Durant une expérience, les fichiers .edf s'enregistrent automatiquement sur le host computer, dans la partition Windows. Pour les récupérer, démarrez l'ordinateur host sur Windows, allez dans le disque dur Eyelink, puis dans le dossier [elcl/data](#). Vous retrouverez ici vos fichiers. Vérifiez régulièrement que le disque dur interne du host computer n'est pas saturé et supprimez les fichiers que vous avez récupérés, par exemple à la clôture d'une expérience (= vous avez copié tous les fichiers .edf et en avez fait un ou plusieurs back-up, donc vous pouvez les supprimer définitivement du host computer). Vous pouvez également inclure des commandes dans votre script Matlab pour faire en sorte que les données s'enregistrent aussi sur le display computer (voir Section 4).
- **Fichier Eyelink .asc** : En plus du fichier .edf, Eyelink enregistre aussi les data brutes (donc sans aucun pré-traitement). Ces données sont facilement lisibles dans un éditeur de texte (.asc). S'il peut toujours être utile de consulter ce fichier (qui s'ouvre comme un fichier texte), il est relativement rare de le faire et c'est toujours pour des besoins très spécifiques car l'avantage d'utiliser DataViewer est qu'il pré-traite adéquatement les données (e.g., calcul des coordonnées moyennes d'une fixation sur une durée donnée). Si vous désirez néanmoins récupérer les données brutes au format ascii, une façon de faire est de convertir les fichiers .edf en utilisant l'exécutable [EDF2ASC.EXE](#) sur le host computer. Voici les étapes à suivre, pour un fichier nommé «test.edf» : (1) démarrez EyeLink sans brancher le câble de la caméra, (2) quand le prompt apparaît, tapez `cd ..`, (3) tapez `cd data`, (4) tapez `EDF2ASC.EXE test.edf`, (5) éteignez Eyelink et démarrez Windows : le fichier `test.asc` se trouve dans [EyeLink/elcl/data](#).
- **Fichiers Matlab .txt** : En fonction de la façon dont vous avez programmé votre expérience, plusieurs fichiers .txt de données peuvent être générés par Matlab. Dans l'exemple présent, il y en a deux : *Mt1s1p6.txt* (stockage des metadata, de l'identité et de l'ordre des écrans présentés, du temps passé sur chaque écran) et *M_quest_t1s1p6.txt* (stockage de l'identité et de l'ordre des questions de compréhension, des réponses données et du temps mis).

5.3. Visualisation des données sous DataViewer

Pour accéder aux données, il faut ouvrir le fichier .edf dans DataViewer (plusieurs fichiers peuvent être chargés en même temps). Dans le même dossier où se trouvent les fichiers .edf doivent se trouver les fichiers images .jpg : cela va permettre de voir où tombent exactement les fixations. La Figure 26 donne une image de la vue qui s'ouvre sous DataViewer. Attention, la description précise des sous-fenêtres, champs et fonctionnalités se trouvent dans le *DataViewer manual* : ne sont repris ici que les éléments de base et/ou utiles à l'exportation des données.

La fenêtre est divisée en deux grandes sections.

1. Le trial view (partie droite de la Figure 27) montre les données spatialement. On voit effectivement le texte avec le fond d'écran utilisé pendant le testing (qui vient des images .jpg), les ROIs (*regions of interest*, en rouge, générées par l'Eyelink sur base des informations transmises pendant le testing) et les fixations (en bleu clair). D'autres informations peuvent être ajoutées (par ex. Les saccades) en utilisant les boutons de la barre d'outils générale (Figure 26), mais par défaut, les fixations et les ROIs sont tou-



Figure 28. Boutons d'ajout des fixations, saccades, clignements,... dans la fenêtre TrialView

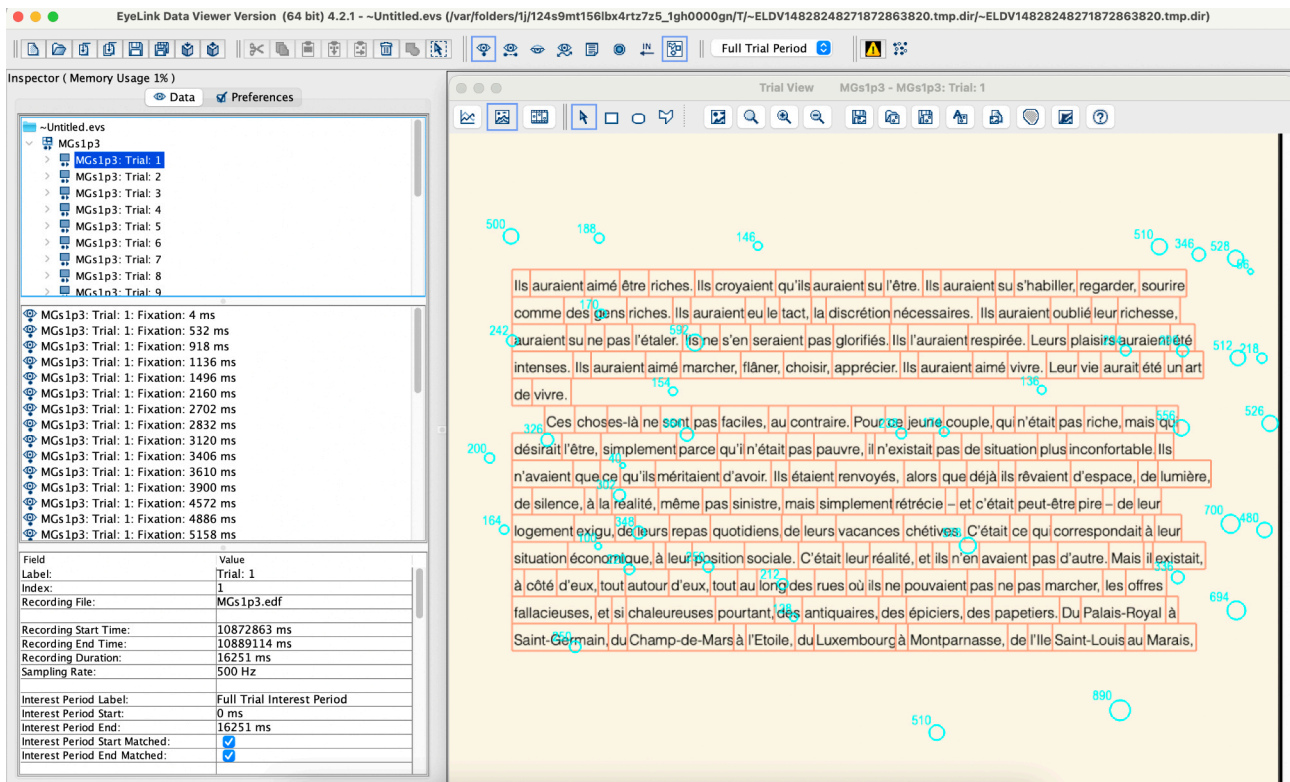


Figure 29. Vue de DataView après le chargement d'un fichier .edf

jours montrées. Si les couleurs de marquage rendent le tout difficilement lisible, il est possible de les modifier dans les préférences (Figure 28). Plus généralement, le panel «préférences» permet de modifier de nombreux paramètres, non seulement concernant l'affichage des données dans DataView mais également concernant la façon dont DataView pré-traite les données. Le manuel EyeLink DataView détaille cela précisément, mais la plupart des paramètres par défaut pourraient être tout à fait adéquats pour le genre d'analyses que vous voudriez faire (voir plus bas).

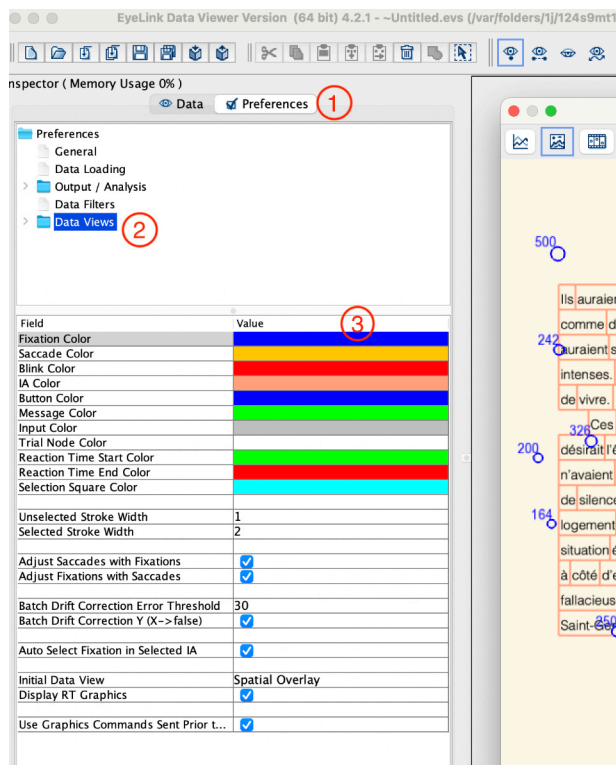


Figure 30. Panel «Préférences» pour modifier des paramètres de visualisation et de pré-traitement des données. Ici, on modifie la couleur des fixations sur le TrialView (3)

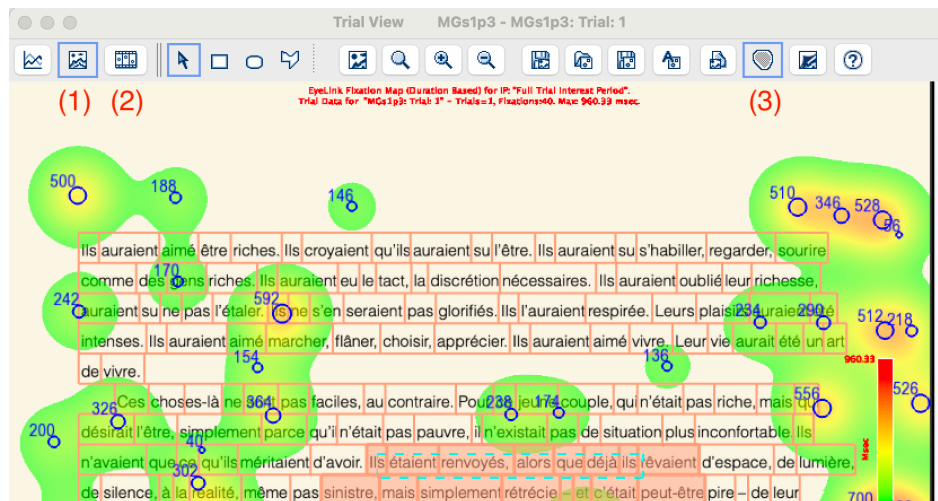


Figure 31. Différentes présentation des données spatialement (1: statique, 2 : temporel sous forme de film, 3 : heat map)

A noter que le trial view permet de générer de nombreuses vues différentes de vos données, comme notamment un film de votre essai ou une *heat map* (voir Figure 29).

2. L'inspector est la deuxième fenêtre centrale de DataViewer, elle-même constituée de deux volets (Figure 30). Comme dit précédemment, le volet 'Preferences' permet de modifier des paramètres d'affichage ou de pré-traitement. Le volet 'Data' permet d'accéder au détail des données, organisées en fonction des ROIs définies préalablement.

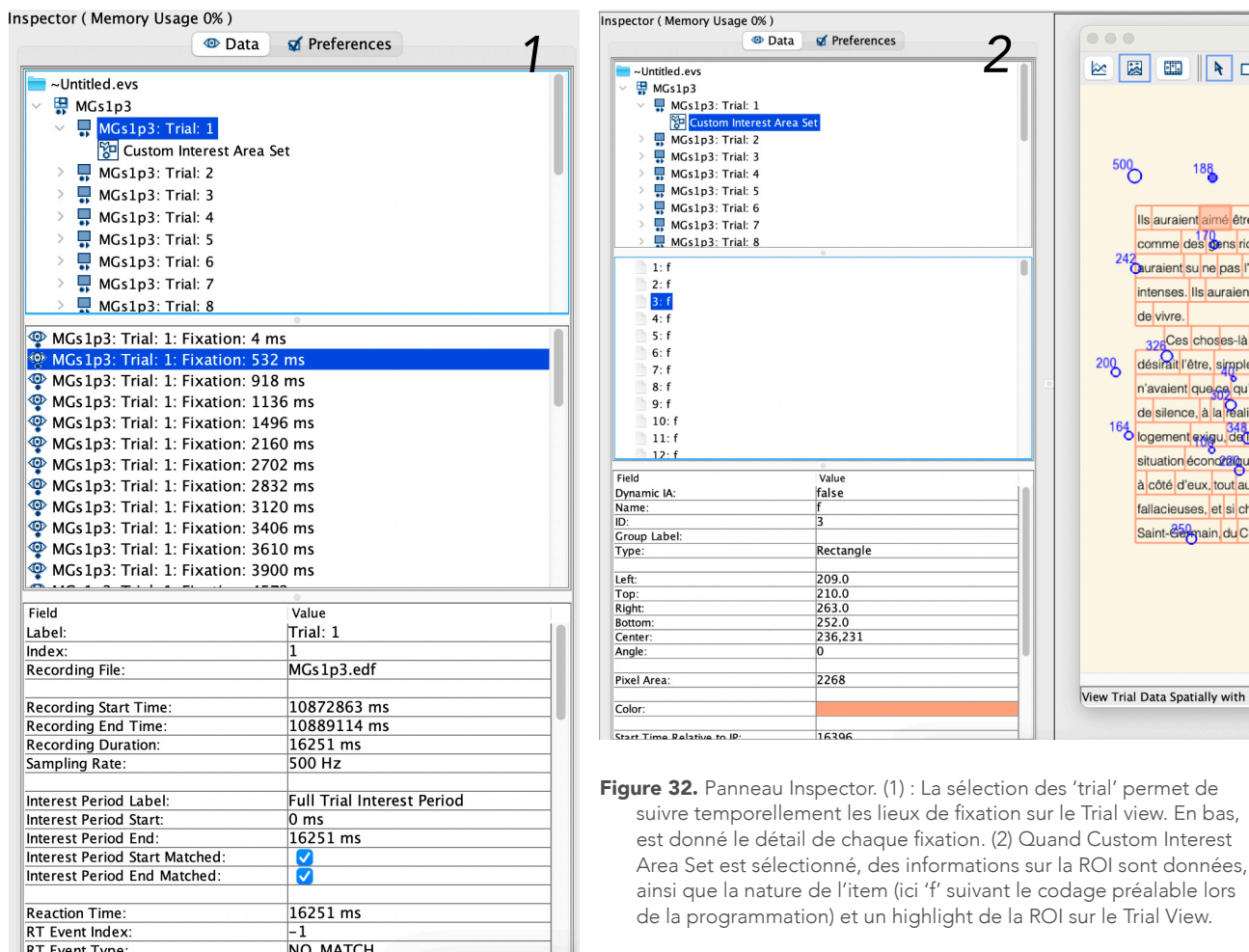


Figure 32. Panneau Inspector. (1) : La sélection des 'trial' permet de suivre temporellement les lieux de fixation sur le Trial view. En bas, est donné le détail de chaque fixation. (2) Quand Custom Interest Area Set est sélectionné, des informations sur la ROI sont données, ainsi que la nature de l'item (ici 'f' suivant le codage préalable lors de la programmation) et un highlight de la ROI sur le Trial View.

Pour la visualisation des données, et avant l'exportation, un certain nombre de paramètres doivent être modifiés dans le panel Preferences (cf. Figure 28).

- Dans *General*, vous pouvez changer le default directory (c'est là que les images des stimuli sont chargées), mais DataViewer ne semble pas accepter les chemins trop longs... Changez la résolution si ce n'est pas la bonne (par défaut, elle est de 1024 x 768).
- Dans *Data Loading*, assurez-vous que fixations, saccades, blinks, et messages sont loaded. Décochez 'Collapse Identical Interest Areas with different time stamps' (ce qui est fait typiquement quand l'analyse est faite sur des périodes d'intérêt individuelles, cf. p.117 DataViewer manual).
- Dans *Output/Analysis*, cochez UTF8 encoding (plus facile pour manipuler les données sous R) et notez que le variable delimiter doit être la tabulation (\t).

5.4. Analyse visuelle de la qualité des données

Avant toute exportation des données et traitement, la première étape est l'inspection visuelle des données. C'est la plus longue, mais cela permet deux choses : repérer les essais problématiques (écrans pour lesquels les mouvements oculaires sont erratiques) et repérer les sujets problématiques (sujets pour lesquels la majorité des données par écran sont inutilisables). Cela va donc vous permettre d'exclure a priori les essais et participants inutilisables.

Pour faire cela correctement, il va falloir passer en revue chaque écran de chaque sujet (par exemple pour une expérience avec 50 sujets qui sont exposés à 100 écrans chacun, cela fait 5000 vérifications). Ça peut paraître long, mais en fait, une fois le coup pris, ce n'est pas si fastidieux. Comme toujours en cas d'inspection visuelle, un point critique est d'adopter les mêmes critères tout au long de la vérification. Comme ces critères (et le 'feeling' de 'bonnes données') est en partie construit empiriquement (comme en EEG : chaque chercheur à son appréciation d'un bon signal en fonction de toute son expérience), cela peut être une bonne idée de jeter un coup d'oeil aléatoirement aux données de plusieurs sujets en amont, pour que vous vous fassiez une idée du signal.

5.4.1. Éléments clés de l'inspection visuelle

Après vous être assuré.e que les fichiers .edf et fichiers .jpg sont dans le même dossier, chargez les fichiers .edf dans Data Viewer. Prenez tous les fichiers des sujets. En fonction du nombre d'essais et du nombre de sujets, cela peut prendre beaucoup de temps. Passez alors chaque essai de chaque sujet en revue. Il y a deux choses à prendre en compte : l'**allure générale des fixations** (est-ce qu'elles tombent bien ? Est-ce qu'elles suivent les ROIs ?) et la **séquence des fixations** (tout particulièrement quand elles sont très denses). Avec l'expérience, un simple coup d'oeil suffit à savoir si les données sont bonnes pour un essai. Voici des exemples :

1. *Données de bonne qualité* (Figure 31) : on voit qu'à première vue les fixations suivent correctement les ROIs. Ajouter la visualisation des saccades confirme cela. De plus, on peut naviguer dans le panel du milieu de l'Inspector pour voir la séquence des fixations : on clique sur la première fixation (dont le rond se remplit sur le Trial view) et on fait défiler les suivantes en passant de l'une à l'autre avec la flèche du bas. Cela permet de recréer le décours temporel des fixations (on peut aussi jouer la vidéo pour cela).
2. *Données décalées* (Figure 32) : on voit que les fixations suivent correctement le texte et les ROIs, mais il y a un **décalage vertical** (ici, elles sont toutes décalées vers le haut). C'est un pro-

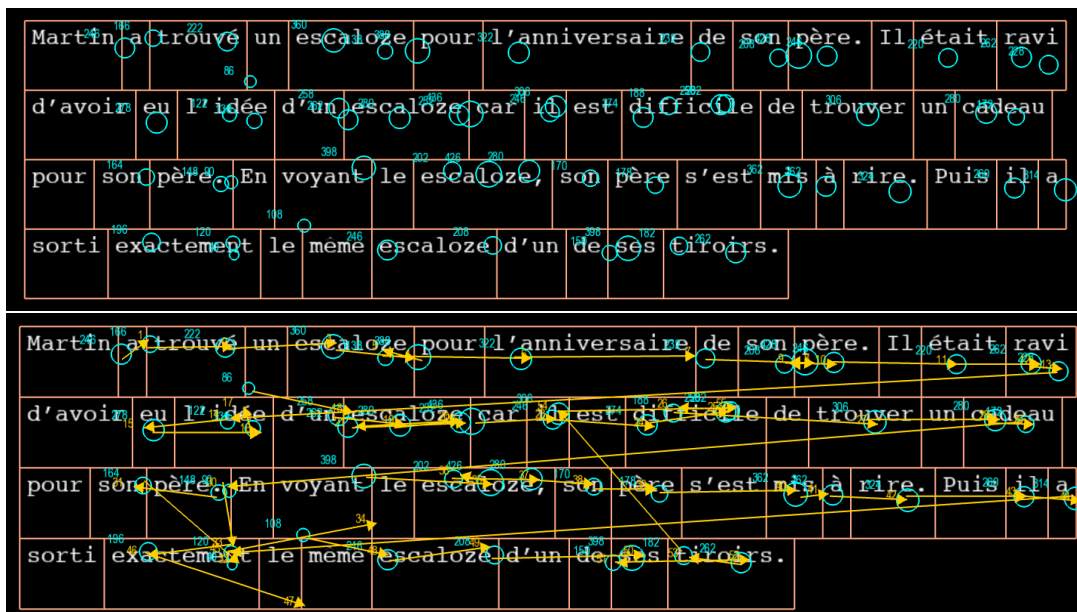


Figure 33. Données de bonne qualité dans le Trial Viewer. En haut, juste les fixations, en bas les saccades sont ajoutées.

blème, car lors de l'extraction des données, des fixations pourraient être erronément comptabilisées pour des mots cibles. Par exemple ici, les fixations rouges vont être comptabilisées pour les mots *semaine*, *souvenait* et *Claire*, alors qu'elles correspondent à des fixations du mot *joudance*. De même, les fixations bleues seraient comptabilisées pour le mot *joudance* alors qu'elles correspondent au mot en dessous. Enfin, en l'état la conclusion serait que la dernière ligne n'a reçu aucune fixation (ce qui n'est bien sûr pas le cas). Il y a deux solutions ici : soit vous décidez de supprimer l'essai, soit vous décidez de corriger l'essai. La deuxième solution est tout à fait appropriée, surtout si cela se répète souvent pour un sujet donné. Cela suggère simplement que la calibration n'a pas été optimale et comportait déjà le décalage. La section 5.4.3. présente comment corriger un tel décalage vertical. Il y a parfois un **détailage horizontal**, mais de mon expérience, la correction est plus difficile à apporter (vous devez faire des inférences fortes sur les lieux de fixation) et pour ma part, j'ai tendance à ne pas les corriger et à exclure tout simplement l'essai. Enfin, parfois le **détailage est parabolique** (en forme de U / V), ou il est vertical mais pas homogène (Figure 33) ou encore il combine à la fois un décalage vertical et horizontal : autant un décalage vertical est facile à corriger, autant tout autre forme de décalage, même si on en comprend la répartition spatiale, peut s'avérer hasardeux à corriger. Dans ces cas, il faut mieux exclure l'essai (et le participant, si beaucoup d'essais sont touchés).

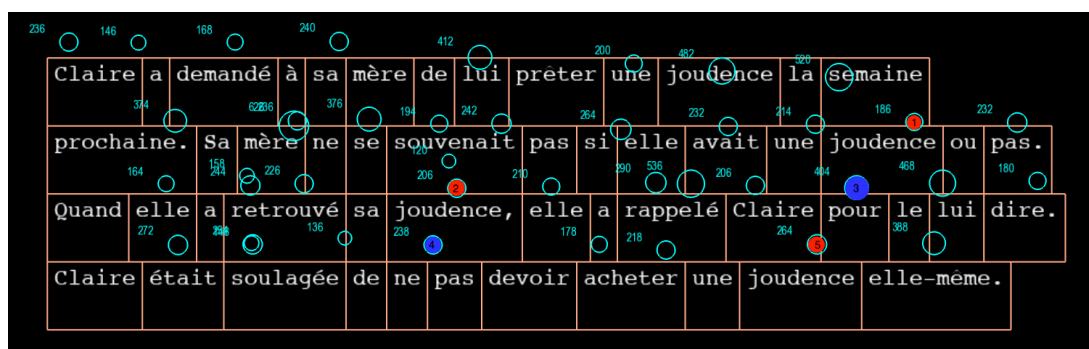


Figure 34. Données présentant un détailage vertical

3. *Données moins transparentes* (Figure 34). Ce ne sont pas forcément des données à jeter, mais des données qui nécessitent que vous passiez plus de temps dessus pour décider si oui ou non

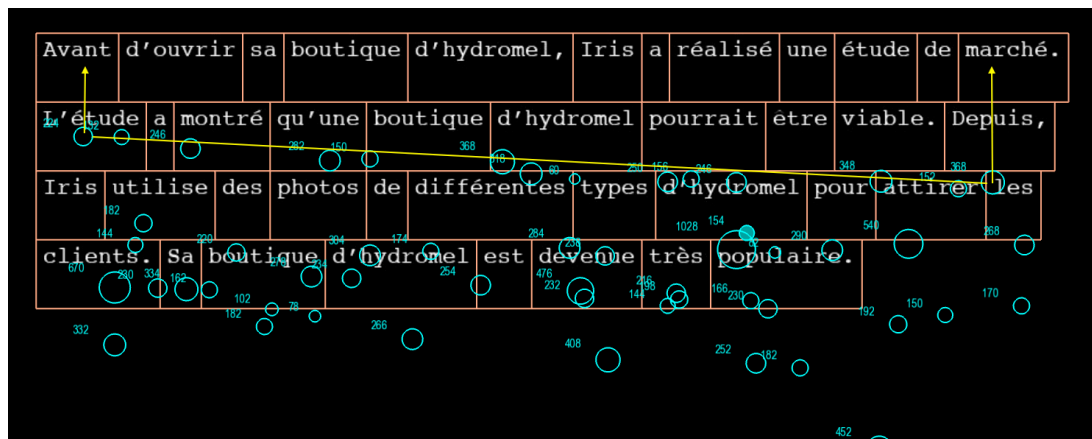


Figure 35. Données présentant un décalage vertical inégal

elles sont cohérentes. Dans l'exemple ici, les fixations sont denses au début, puis « normales » vers la fin. Il faut comprendre ce qui s'est passé au début donc. En repassant sur les fixations, on se rend compte que la première ligne a été lue plusieurs fois. A nous alors de décider si oui ou non on garde les données de l'essai.

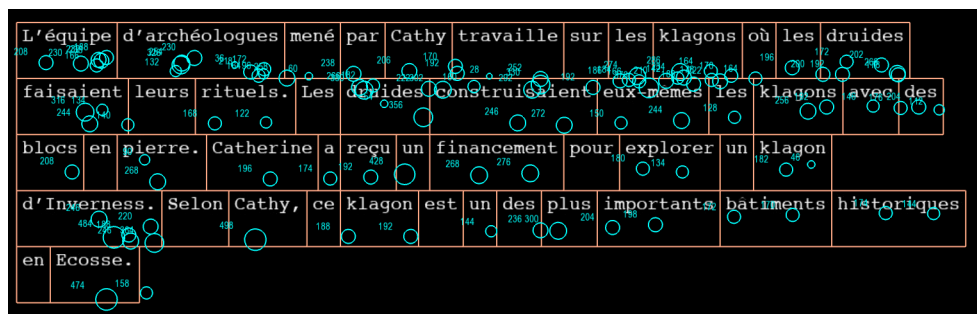


Figure 36. Données méritant qu'on s'y arrête

4. *Données inutilisables* (Figure 35). C'est le cas de données où plusieurs décalages sont présents ou bien où le sujet a fait quelque chose de difficilement compréhensible.

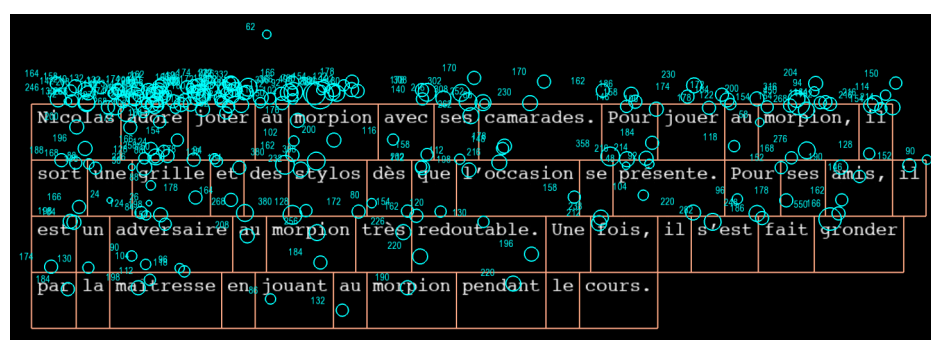


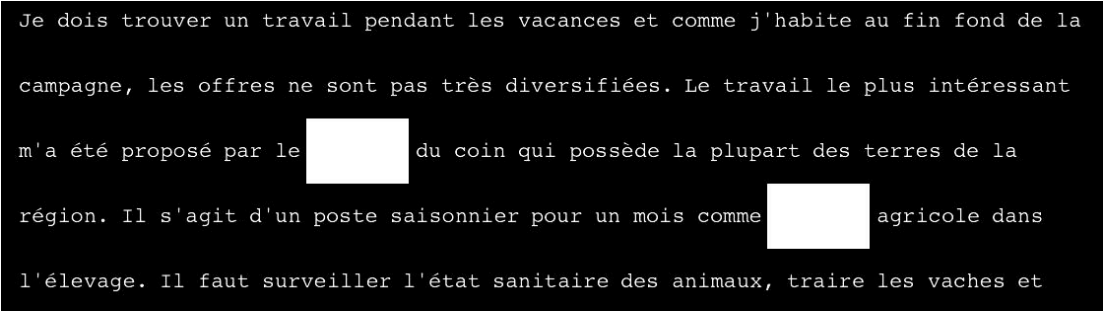
Figure 37. En plus d'un décalage vertical, ces données semblent présenter beaucoup de bruit : il y a fort à parier que cet essai soit à exclusion (sauf si en regardant le décours temporel le patron devient plus clair).

Durant l'étape d'inspection visuelle, il est utile de consigner dans un tableau excel à double entrée (sujet, écran/essai) vos commentaires. A minima, vous devriez noter les essais qu'il va ensuite falloir exclure lors des analyses sous R.

5.4.2. Prise en compte du biais de l'expérimentateur

Dans certains cas, vous pourriez vouloir vous prémunir de tout biais inconscient qui vous amèneriez à prendre des décisions en faveur de vos hypothèses. C'est tout particulièrement le cas dans les expériences où des ROIs d'items cibles sont définies et où un facteur intra-sujet est manipulé. Par exemple, si vous testez l'effet de fréquence des mots, pour les mêmes phrases par écran, le mot cible est parfois un mot fréquent, parfois un mot rare. Or, quand vous faites votre inspection, non seulement vous voyez les mots (et donc vous pouvez déduire la condition de l'essai), mais en plus vous avez une indication très nette des données (plus le rond est gros, plus l'item a été fixé longtemps). Lors de mes toutes premières analyses eye-tracking, je me suis dit que je pourrais être malgré moi soumise à un biais de sélection des items à exclure (par ex. plus de tendance à rejeter des essais pas clairs quand la durée de fixation ne semble pas aller dans le sens de mes hypothèses).

Pour me prémunir de cela, j'ai décidé de réaliser l'inspection visuelle non pas avec l'image du texte des essais «brute», mais avec une image où les mots cibles ont été masqués au préalable : je vois le texte, je vois les fixations, mais je ne vois pas l'identité des mots cibles (Figure 36). Pour cela, il faut re-générer des images .jpg (cf. Section 4). Une autre façon de faire est d'importer les fichiers .edf sous dataviewer sans que les fichiers .jpg ne soient présents dans le dossier : le data trial affichera les ROIs et les fixations, mais aucun mot. Un bémol avec cette méthode : vous ne savez pas précisément où étai(en)t le(s) mot(s) cible(s), or parfois les données ne sont bruitées que sur une partie de l'essai : vous pourriez être tenté.e de jeter l'essai, mais si vous vous apercevez que les données sont correctes autours des mots cibles, vous pourriez garder l'essai. Pour finir, gardez en tête que cette procédure supplémentaire n'est pas toujours nécessaire : si ce qui vous intéresse sont des indices généraux de lecture d'un texte (pas de «vraies» ROIs), pas besoin de faire cette étape.



Je dois trouver un travail pendant les vacances et comme j'habite au fin fond de la campagne, les offres ne sont pas très diversifiées. Le travail le plus intéressant m'a été proposé par le [] du coin qui possède la plupart des terres de la région. Il s'agit d'un poste saisonnier pour un mois comme [] agricole dans l'élevage. Il faut surveiller l'état sanitaire des animaux, traire les vaches et

Figure 38. Exemple de texte dans le Trial View avec les mots cibles masqués

5.4.3. Corriger des données

«Corriger les données» se fait essai par essai, semi-manuellement, en cas de **décalage vertical uniforme**. Reprenons l'exemple de la Figure 32 : il semblerait ici que si on remontait toutes les ROIs de quelques pixels, alors les points de fixation tomberaient correctement.

Dans l'expérience utilisée pour l'exemple, cet essai est l'essai 5 du sujet 105 et correspond à l'écran 62 (les numéros d'essai et d'écran ne sont pas identiques ici car les essais ont été randomisés). Voici la marche à suivre pour corriger les données. Après avoir ouvert le fichier .edf et repéré les essais qui posent problème pour un sujet donné (inspection visuelle), ouvrez le script [ROIchanges.r](#) (l'entièreté du code est disponible en Annexe). Dans la section «Paramètres», changez ce qui est nécessaire (les chemins d'accès, le numéro de sujet, d'essai et de texte, le lag). Le lag correspond au nombre de pixels que vous voulez ajouter ou retirer pour décaler les ROIs (-10 = toutes les lignes des cadres jaunes sont remontées de 10 pixels). Après quelques essais, vous arriverez facilement à déterminer s'il faut que vous ajoutiez 10 pixels ou plutôt 30 pour 'réparer' tel ou tel essai. Lorsque vous lancez le script, cela génère un fichier, intitulé ici « newROI_Ss_105_trial_5.ias ».

Retournez ensuite sous DataView. Dans Inspector, à 'Trial 5, cliquer sur la flèche : cela fait apparaître «Custom Interest Area Set» (Figure 37). Faites un clic droit sur Custom Interest Area Set et choisissez 'Delete'. Cela permet de supprimer toutes les ROIs (Figure 38), ce qui nous laisse alors le champ libre pour en ajouter de nouvelles. Pour ajouter les nouvelles ROIs, allez dans Trial view (fenêtre du texte), cliquez sur l'icône 'Import Area of interest file' (Figure 39). Par défaut, il s'ouvre sur le bureau, c'est donc en partie pour cela que les fichiers .ias générés par le script R précédent doivent être enregistrés sur le bureau dans un premier temps (cf. ligne 25 du script). Comme le montre le Figure 40, les nouvelles délimitations apparaissent et seront traitées comme telles lors de l'exportation des données. A ce stade, regardez si la nouvelle délimitation est correcte, sinon recommencer les étapes précédentes (génération dans R d'un nouveau fichier .ias avec une translation verticale un peu différente et ré-importation dans Data viewer).

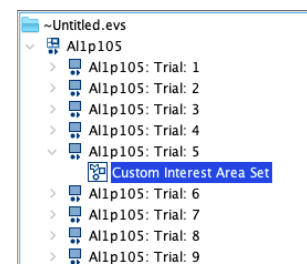


Figure 39. Espace de stockage des coordonnées des ROIs

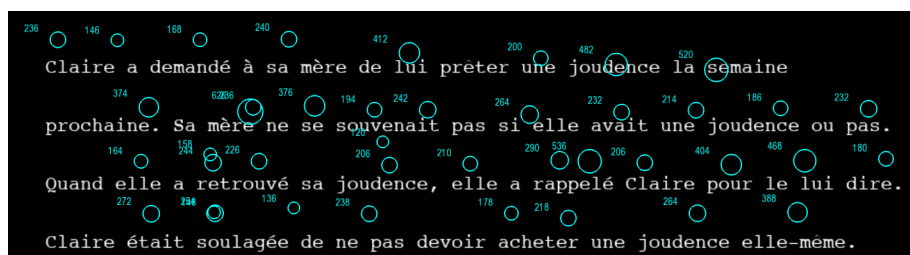


Figure 40. Suppression des ROIs pour l'essai problématique

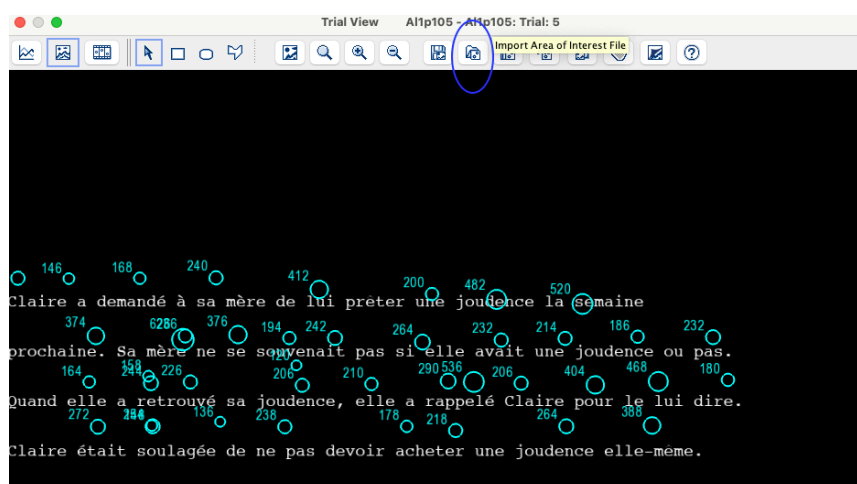


Figure 41. Importer de nouvelles coordonnées de ROIs

Cette façon élégante de corriger les données est à faire sujet par sujet, essai par essai. S'il y a beaucoup d'essais à corriger, il est possible d'améliorer le script R pour que les fichiers .ias soient générés en boucle. Néanmoins, il semble qu'il n'y ait pas d'alternative pour Data Viewer que de procéder manuellement (les coordonnées initiales des ROIs ont été envoyées par le script Matlab pendant l'expérience).

Dernier point important : Lorsque vous ouvrez un fichier .edf, les ROIs qui sont stockées sont celles que Matlab a envoyé à l'eye-link pendant la prise de données. Cela signifie que si vous corriger des essais et que vous quittez ensuite Data Viewer, lorsque vous ré-ouvrirez le fichier .edf par la suite, votre correction ne sera plus disponible. Deux solutions :

- Vous avez gardé les fichiers .ias et vous vous amusez à tous les réimporter pour chaque essai de chaque sujet qui a nécessité une modification (au secours !)

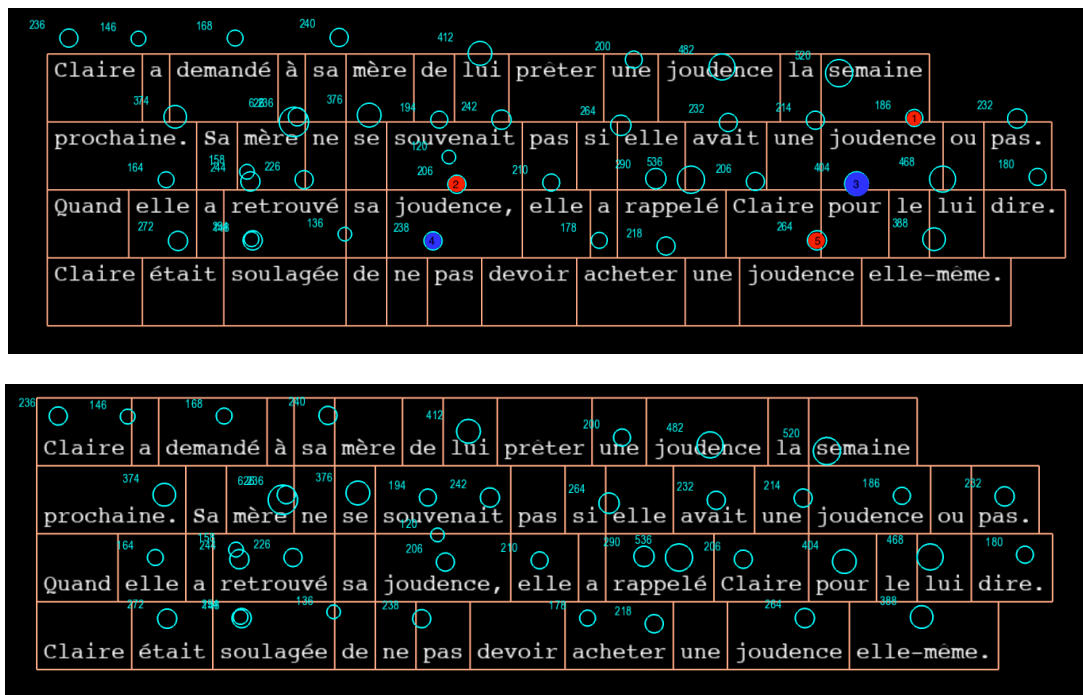


Figure 42. Résultats de la modification des coordonnées des ROIs pour supprimer le décalage vertical

- Vous ne voulez pas tout refaire à la main, et donc dès qu'il y a au moins un essai dont les ROIs ont été modifiées, vous enregistrez votre session : faites **File/Open/Save As**. Cela génère un dossier .res et un fichier .evs. Les deux sont à conserver. Le fichier .evs est l'équivalent du fichier .edf mais en incluant les nouvelles ROIs. Important : je vous conseille aussi de garder quelque part bien rangé les fichiers .ias que vous aurez générés...

5.4.4. Enregistrement de la session Data Viewer

Que vous fassiez des modifications de ROIs ou pas, cela est toujours une très bonne idée d'enregistrer sa session Data Viewer via **Files/Open/Save As**. Cela vous permet d'éviter de perdre beaucoup de temps lors de l'importation des fichiers .edf... La Figure 41 montre que j'ai appelé ma session «essai1». Dans cette session, j'avais chargé les données de deux sujets (*Alt1p107* et *Alt1p105*). Pour ré-ouvrir toute la session, il me suffira de cliquer sur «essai1.evs» et je retrouverai les 2 sujets.

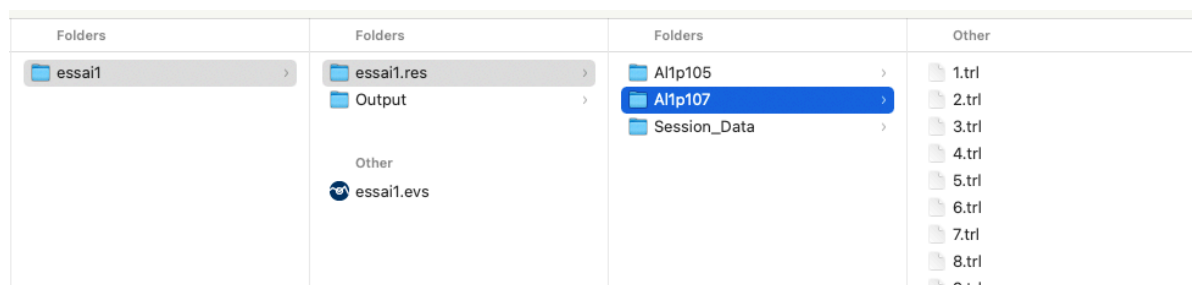


Figure 43. Enregistrement d'une session

5.5. Exportation des données sous Data Viewer

Une fois les paramètres ajustés et les éventuels ROIs corrigés, le but est de créer un rapport (*report*) avec les données voulues, qui pourra alors être traité dans un logiciel de statistiques. Il y a 4 types de rapports possibles à générer (trial, fixation, saccade, interest area) :

Field	Contents
RECORDING_SESSION_LABEL	Label of the data file
* AVERAGE_FIXATION_DURATION	Average duration (in milliseconds) of all selected fixations in the trial
* AVERAGE_SACCADE_AMPLITUDE	Average size (in degrees of visual angle) of all selected saccades in the trial
DURATION	Duration of the trial recording between the “START” message and the “END” message
* FIXATION_COUNT	Total number of fixations in the trial
INDEX	Sequential order of the trial in the recording
* SACCADE_COUNT	Total number of saccades in the trial
TRIAL_LABEL	Label of the trial

Figure 44. Exemples de variable à inclure dans un trial report

- **TRIAL REPORT** : Ce type de rapport sert à résumer la performance pour un essai entier. On entend par essai un «screen». Dans le cas de lecture de texte, c’est utile par exemple pour connaître par exemple la durée moyenne de lecture de chaque texte par écran (mais à noter que les fichiers Matlab contiennent déjà l’information). Pour générer le rapport, allez dans [Analysis/Report/Trial Report](#), sélectionnez les noms des variables créées sous Matlab et les champs présentés dans la Figure 42. Une fois que les champs sont sélectionnés, choisissez [Next](#), et spécifiez un nom et un emplacement pour le fichier de sortie. Choisissez un format .txt pour faciliter l’importation sous R).
- **FIXATION REPORT** : Ce type de rapport permet de créer une présentation en colonne des fixations par les événements. Pour le générer, allez dans [Analysis/Report/Fixation Report](#).
- **SACCADE REPORT** : Ce type de rapport permet de créer une présentation en colonne des différentes saccades effectuées. Pour le générer, allez dans [Analysis/Report/Saccade Report](#).
- **INTEREST AREA REPORT** : Ce type de rapport permet de créer une présentation en colonne des mesures enregistrées pour chaque région d’intérêt. Dans le cas de textes où des cibles sont clairement définies sur certaines ROIs, utilisez ce type de rapport. Pour générer le rapport, allez dans [Analysis/Report/Interest Area Report](#). Sélectionnez les noms des variables créées dans Matlab et l’ensemble des champs présentés à la Figure 44.

Toutes ces recommandations sont bien sûr à adapter en fonction de vos analyses prévues (par ex. si vous êtes très intéressé.e par la dilatation de la pupille, vous devez inclure les variables qui correspondent à cela).

Important : vous devez extraire un rapport (quelle que soit sa nature) par sujet. Soit vous le faites manuellement pour chaque sujet et dans ce cas, faites bien en sorte que le nom de fichier de chaque rapport ait la même structure, pour faciliter l’importation pour des analyses futures sous R par exemple (par ex. IAR-s1-p1, IAR-s1-p2, IAR-s1-p3, etc...). Soit vous décidez d’exporter les données de tous les sujets de votre session (probablement plus approprié) en cochant la case appropriée (cf. Figure 43). Vous pourrez donner un nom racine à votre fichier (par ex. IAR) et le nom du fichier des sujets sera ajouté. Les fichiers seront stockés dans le dossier «output» de votre dossier de session enregistré.

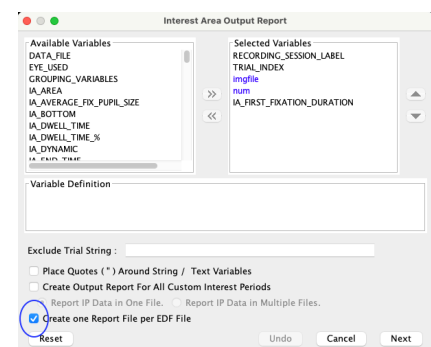


Figure 45. Case à cocher pour exporter les rapports de tous les sujets chargés dans une session

	Field	Contents
	RECORDING_SESSION_LABEL	Label of the data file
	DATA_FILE	File name of the recording
	IA_AREA	Pixel area for the current interest area.
*	IA_DWELL_TIME	Dwell time (i.e., summation of the duration across all fixations) on the current interest area
*	IA_FIRST_FIXATION_DURATION	Duration of the first fixation event that was within the current interest area.
	IA_FIRST_FIXATION_INDEX	Ordinal sequence of the first fixation that was within the current interest area.
	IA_FIRST_FIXATION_X	The X position of the first fixation event that was within the current interest area.
	IA_FIRST_FIXATION_Y	The Y position of the first fixation event that was within the current interest area.
	IA_FIRST_FIX_PROGRESSIVE	Checks whether later interest areas have been visited before the first fixation enters the current interest area. 1 if NO higher IA ID in earlier fixations before the first fixation in the current interest area; 0 otherwise. This measure is useful in reading to check whether the first run of fixations in this interest area is in fact first-pass fixations.
	IA_FIRST_RUN_DWELL_TIME	Dwell time (i.e., summation of the duration across all fixations) of the first run within the current interest area.
	IA_FIRST_RUN_FIXATION_COUNT	Number of all fixations in a trial falling in the first run of the current interest area.
*	IA_FIXATION_COUNT	Total fixations falling in the interest area
	IA_ID	Ordinal ID of the current interest area
	IA_LABEL	Label for the current interest area
	IA_POINTS	List of x,y points for freehand Interest Areas.
	IA_REGRESSION_IN	Whether the current interest area received at least one regression from later interest areas (e.g., later parts of the sentence). 1 if interest area was entered from a higher IA_ID (from the right in English); 0 if not.
*	IA_REGRESSION_IN_COUNT	Number of times interest area was entered from a higher IA_ID (from the right in English).
	IA_REGRESSION_OUT	Whether regression(s) was made from the current interest area to earlier interest areas (e.g., previous parts of the sentence) prior to leaving that interest area in a forward direction. 1 if a saccade exits the current interest area to a lower IA_ID (to the left in English) before a later interest area was fixated; 0 if not.
*	IA_REGRESSION_OUT_COUNT	Number of times interest area was exited to a lower IA_ID (to the left in English) before a higher IA_ID was fixated in the trial.
	IA_REGRESSION_OUT_FULL	Whether regression(s) was made from the current interest area to earlier interest areas (e.g., previous parts of the sentence). 1 if a saccade exits the current interest area to a lower IA_ID (to the left in English); 0 if not. Note that IA_REGRESSION_OUT only considers first-pass regressions whereas IA_REGRESSION_OUT_FULL considers all regressions, regardless whether later interest areas have been visited or not.
*	IA_REGRESSION_OUT_FULL_COUNT	Number of times interest area was exited to a lower IA_ID (to the left in English).
	IA_RUN_COUNT	Number of times the Interest Area was entered and left (runs)
	IA_SKIP	An interest area is considered skipped (i.e., IA_SKIP = 1) if no fixation occurred in first-pass reading.
	TRIAL_INDEX	Sequential order of the trial in the recording.
	TRIAL_LABEL	Label of the trial

Figure 46. Principales variables à inclure dans un *Interest area report*. Bleu = données concernant les paramètres. Vert = données concernant l'ensemble des fixations. Rouge = données concernant la première fixation. Orange = données concernant les régressions et saccades. * = données à considérer directement pour l'analyse de lecture de texte ou de phrase.

5.6. Traitement des données sous R / RStudio

Il y a deux types de fichiers à traiter : les fichiers .txt éventuels correspondant aux données enregistrées par Matlab pour chaque sujet lors de l'expérience et les fichiers report extraits via Data Viewer pour chaque sujet.

5.6.1. Fichiers matlab .txt

En fonction de votre expérience, un ou deux fichiers Matlab auront été générés pour chaque participant. Dans l'exemple A de la Figure 45, l'expérience était une lecture de petits textes où chaque écran correspondait à un essai. Il y a un seul fichier (A) reprenant notamment le temps passé sur chaque écran (colonne D, en secs), le temps mis pour répondre à une question sur l'écran (colonne E) en cas de question (colonne I). Le second exemple correspond à la lecture d'un texte continu sur plusieurs écrans. Le fichier représenté en B correspond au temps passé sur chaque écran avec les metadata. Le fichier représenté en C correspond aux questions et réponses données par le sujet.

Ces fichiers sont à traiter comme pour n'importe quelle expérience (importation sous R, groupement des données, moyennages, etc...).

A

	A	B	C	D	E	F	G	H	I	J	K
1	numPs	liste	numTrial	timeScreen	timeQuest	screen	domOeil	age	quest	corrRespQue	respQuest
2	105	1	1	12.025362	0	screen_30.jpg	1	19	none	NA	none
3	105	1	2	14.93589	2.237563	screen_3.jpg	1	19	Clara a rev8i n		space
4	105	1	3	12.485403	0	screen_18.jpg	1	19	none	NA	none
5	105	1	4	12.680787	0	screen_62.jpg	1	19	none	NA	none
6	105	1	5	11.645251	0	screen_44.jpg	1	19	none	NA	none
7	105	1	6	10.76802	0	screen_78.jpg	1	19	none	NA	none
8	105	1	7	12.332061	0	screen_61.jpg	1	19	none	NA	none
9	105	1	8	13.659235	0	screen_72.jpg	1	19	none	NA	none
10	105	1	9	11.219152	0	screen_51.jpg	1	19	none	NA	none
11	105	1	10	13.401949	0	screen_23.jpg	1	19	none	NA	none
12	105	1	11	13.507306	3.440168	screen_75.jpg	1	19	MV@lina a s c		space
13	105	1	12	10.92903	0	screen_25.jpg	1	19	none	NA	none
14	105	1	13	14.26471	0	screen_37.jpg	1	19	none	NA	none
15	105	1	14	15.196786	0	screen_1.jpg	1	19	none	NA	none
16	105	1	15	14.784127	0	screen_86.jpg	1	19	none	NA	none

B

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	numPs	texte	numScreen	timeScreen	screen	vue	domOeil	lat	LM	pbLect	niveau	filiere	age	genre
2	11	1	1	90.439853	screen_1.jpg	0	1	1	1	1	0 3 2 ans solva	1	21	2
3	11	1	2	85.216709	screen_2.jpg	0	1	1	1	1	0 3 2 ans solva	1	21	2
4	11	1	3	72.625087	screen_3.jpg	0	1	1	1	1	0 3 2 ans solva	1	21	2
5	11	1	4	70.966511	screen_4.jpg	0	1	1	1	1	0 3 2 ans solva	1	21	2
6	11	1	5	81.330782	screen_5.jpg	0	1	1	1	1	0 3 2 ans solva	1	21	2
7	11	1	6	78.281134	screen_6.jpg	0	1	1	1	1	0 3 2 ans solva	1	21	2
8	11	1	7	85.396566	screen_7.jpg	0	1	1	1	1	0 3 2 ans solva	1	21	2
9	11	1	8	86.21488	screen_8.jpg	0	1	1	1	1	0 3 2 ans solva	1	21	2
10	11	1	9	82.059159	screen_9.jpg	0	1	1	1	1	0 3 2 ans solva	1	21	2

C

	A	B	C	D	E	F	G	H
1	numPs	session	texte	numQuest	timeResp	question	acc	keyResp
2	11	1	1	1	3.346905	Les deux protagonistes s c	c	
3	11	1	1	2	3.270576	Les deux protagonistes v n	n	
4	11	1	1	3	2.610514	Les deux protagonistes s n	n	
5	11	1	1	4	3.197974	Les protagonistes ont toi c	c	
6	11	1	1	5	6.520462	Les deux protagonistes ti c	c	
7	11	1	1	6	5.735356	Les deux protagonistes o n	c	
8	11	1	1	7	2.446321	Les protagonistes n'aime n	n	
9	11	1	1	8	10.445943	Les deux protagonistes , c	c	
10								

Figure 47. Type de fichiers matlab possiblement générés. (A) : Un seul fichier pour une expérience où chaque écran correspond à un essai. (B) et (C) : Deux fichiers générés pour une expérience de lecture continue d'un texte (B = informations sur chaque écran, C = information sur les questions finales).

5.6.2. Fichiers 'report' de Data Viewer

a) Cas de Trial reports

La Figure 46 présente en A l'output d'un Trial report pour un sujet en fonction des champs demandés lors de l'exportation (voir Section 5.5). Ici, chaque ligne correspond à un écran. Les données temporelles sont données en millisecondes. La première chose à constater est que la durée pour chaque écran est similaire au centième de seconde avec ce qui est donné dans l'output de Matlab : bonne nouvelle !

Avec ces données, on peut calculer notamment le temps moyen passé sur chaque écran (et donc sur tout le texte, s'il s'agit de la lecture d'un texte continu), le nombre moyen de fixations, de saccades, et autre. Pour cela, il faut passer par les analyses classiques sous R : importation des données de tous les sujets, mise ensemble, suppression du premier item s'il s'agit d'un écran test (ce qui est le cas dans la Figure 46), moyennages divers. Il est aussi possible de supprimer les écrans problématiques (cf. ci-dessous).

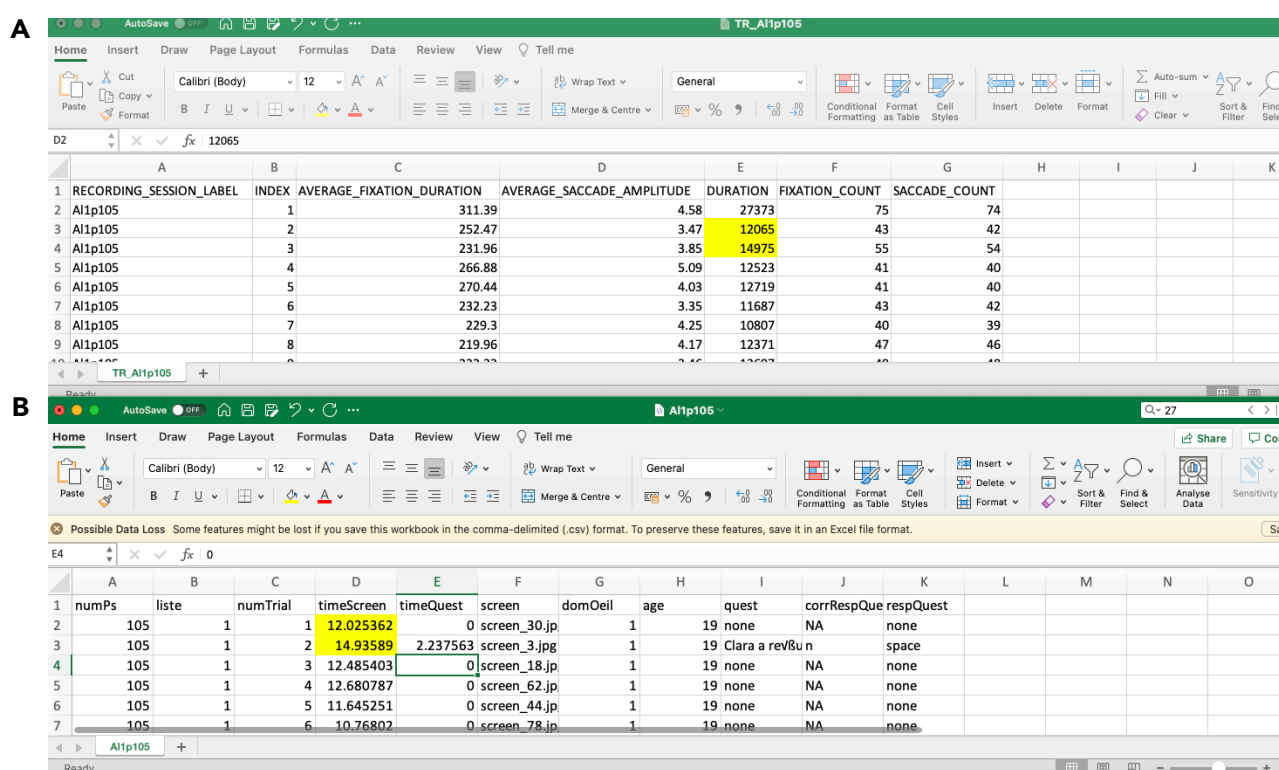


Figure 48. Données générales sur chaque écran provenant de DataViewer (A) comparée à certaines issues de Matlab (B)

b) Cas des Interest area reports

Comme expliqué précédemment, ce type de rapport est nécessaire si vous vous intéressez à la fixation dans des zones d'intérêt. La Figure 47 donne un aperçu du type de données disponibles. Ici, chaque ligne correspond à une zone d'intérêt. Si votre expérience correspond à la lecture de 100 textes comprenant 120 mots chacun pour lequel une ROI a été définie, alors vous aurez 12,000 lignes par sujet. Les lignes où beaucoup de données sont vides correspondent à des ROIs pour lesquelles il n'y a pas eu de fixations.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
1	imgfile	num	RECORDING DATA FILE		IA AREA	IA_DWELL	IA_FIRST_FIXATION_DURATION	IA_FIRST_FIXATION_DURATION	IA_FIRST_FIXATION_DURATION	IA_FIRST_FIXATION_DURATION	IA_FIRST_FIXATION_DURATION	IA_FIRST_FIXATION_DURATION	IA_FIRST_FIXATION_DURATION	IA_FIRST_FIXATION_DURATION	IA_FIRST_FIXATION_DURATION	IA_LABEL	IA_POINTS	IA_REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESSIA	REGRESS

Figure 49. Données exportables via un rapport « interest area »

Pour traiter ce type de fichier, on va passer par les étapes classiques de traitement sous R (importation de tous les sujets, mise ensemble,...). Il y a aussi un nombre d'étapes spécifiques à faire :

- Ne gardez que les ROIs qui vous intéressent (donc supprimer les lignes avec 'f' dans la colonne IA_LABEL, par exemple avec le package `dplyr`).
- Très probablement, vous serez intéressé.e à numéroter l'apparition de vos cibles si une même cible se répète par essai (par ex. pour votre base de données d : `d %>% group_by(TRIAL) %>% mutate(ordre = 1:n()) %>% as.data.frame()`)
- Vous aurez sûrement exclu des participants et des essais. Pour les participants, retirez avant l'importation les fichiers de ceux que vous ne voulez pas. Pour les essais, vous pourriez écrire un script à part reprenant tous les essais à exclure (Figure 48). Ce script peut être complété au fur et à mesure de l'inspection visuelle même et inclure les raisons d'exclusions. Dans votre script d'analyse, vous introduiriez alors la commande `source("exclusions.R")` si vous avez nommé «exclusions.R» ce script contenant les essais à exclure.

```
= e[-which(e$numPs == 1 & e$TRIAL_LABEL=="Trial: 4"),] # too noisy
= e[-which(e$numPs == 1 & e$TRIAL_LABEL=="Trial: 14"),] # too important lag
= e[-which(e$numPs == 1 & e$TRIAL_LABEL=="Trial: 27"),] # too noisy
= e[-which(e$numPs == 1 & e$TRIAL_LABEL=="Trial: 28"),] # too noisy

= e[-which(e$numPs == 3 & e$TRIAL_LABEL=="Trial: 2"),] # horizontal lag
= e[-which(e$numPs == 3 & e$TRIAL_LABEL=="Trial: 4"),] # horizontal lag
= e[-which(e$numPs == 3 & e$TRIAL_LABEL=="Trial: 10"),] # horizontal lag
= e[-which(e$numPs == 3 & e$TRIAL_LABEL=="Trial: 16"),] # undefined lag
= e[-which(e$numPs == 3 & e$TRIAL_LABEL=="Trial: 29"),]

= e[-which(e$numPs == 4 & e$TRIAL_LABEL=="Trial: 16"),] # partial data
```

Figure 50. Exemple de lignes de code pour exclure des essais. On suppose ici que la base de données s'appelle « e » et que l'on a créé une colonne avec le numéro des participants (« numPs »)

- Les informations sur les essais sont très minimalistes dans le fichier qui provient de DataViewer. Vous pourriez donc vouloir les intégrer avec un fichier plus complet. Il est difficile ici de donner un exemple car c'est totalement dépendant de votre expérience, de si vous avez plusieurs listes ou pas, etc.
- Voici les variables dont vous allez avoir typiquement besoin et qu'il peut être utile de renommer :

```
e$totfix = e$IA_DWELL_TIME
e$firstfix = e$IA_FIRST_FIXATION_DURATION
e$nbfix = e$IA_FIXATION_COUNT
e$totfix = as.numeric(as.character(e$totfix))
e$firstfix = as.numeric(as.character(e$firstfix)) # les . pour les mots non fixes sont transformés en NA
e$nbfix = as.numeric(as.character(e$nbfix))
```

- Vous pourriez vouloir utiliser un cut-off pour la durée de fixation (partant de l'idée qu'une fixation trop courte suggère que l'item n'a pas été fixé «pour de vrai»). Dans une étude précédente, j'ai utilisé 100 ms, mais ce critère mérite sûrement d'être discuté. Toujours est-il qu'il faudra faire quelques arrangements dans les données si vous utilisez un cut-off. Voici un exemple :

```
e$firstfix[which(e$firstfix < 100)]

e$firstfixOK = e$firstfix
e$totfixOK = e$totfix
e$nbfixOK = e$nbfix

e$firstfix[e$nbfix==0]=0
e$firstfixOK[e$nbfix==0]=0

e$totfixOK[e$firstfix < 100] = e$totfix[e$firstfix < 100]-
e$firstfix[e$firstfix < 100]
e$nbfixOK[e$firstfix < 100] = e$nbfix[e$firstfix < 100]-1
e$firstfixOK[e$firstfix < 100] = NA
```

Pour le reste des analyses, il s'agit de procéder comme pour n'importe quel jeu de données, avec des moyennages et autres.



ANNEXE

ROIchange.R

```
# Pour pouvoir etre importe dans DataViewer, le fichier final doit:
# - etre au format .ias
# - col 1: RECTANGLE
# - col 2: numerotation
# - col 3: x1
# - col 4: y1
# - col 5: x2
# - col 6: y2
# - col 7: nom de la zone ROI

rm(list=ls())

## PARAMETRES
rootCarac = "~/Documents/WORK/EXP/22-03_THE_A_manipET/Materiel/screenNonRetina/inputS-
criptEXP/liste_1_ok/"
rootOutput = "~/Desktop/"
suj = 105
trial = 5
texte = 62
lag = -30 # 'minus' mean to move up ROIs, 'plus' means to move down ROIs

## UPLOAD
setwd(rootCarac)
coord = read.delim(paste("caracROI_screen_", texte, ".txt", sep=""), header=F)
noms = read.delim(paste("caracTargets_screen_", texte, ".txt", sep=""), header=F, fi-
leEncoding="Latin1")

## ARRANGEMENT
out = data.frame("RECTANGLE", c(1:dim(noms)[1]), coord[,c(1,3,2,4)], noms)
out$V3 = out$V3+lag
out$V4 = out$V4+lag

## EXPORT
setwd("~/Desktop")
write.table(out, paste("newROI_Ss_", suj, "_trial_",trial, ".ias", sep=""), row.na-
mes=F, col.names=F, quote=F)
```