Auteur	Jean Jeuvrey
Date	21/11/2014
Version	1
Thèmes abordés	Tutoriel pour la prise en main de l'eyelink1000 avec E-prime

SOMMAIRE

Définition des termes

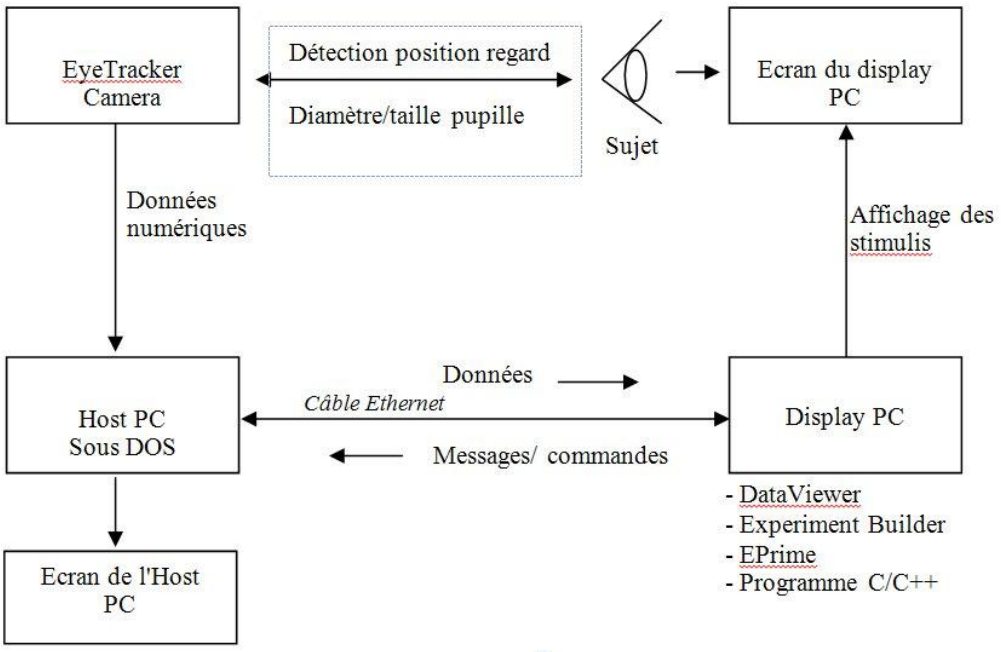
- Host PC : PC hôte, correspond au PC intégré dans le système EyeLink1000
- Display PC : Correspond à l'écran du PC sur lequel est installé le logiciel qui génère les stimulis visuels.

Introduction

Le système eyelink1000 est un dispositif oculométrique qui permet de déterminer la position des points de regard sur un écran de taille variable. Ce dispositif a donc vocation d'être utilisé en intérieur dans des situations ou la tête du sujet est immobile ou quasi-immobile. Dans un premier temps, je vais présenter les principales étapes d'installations du système eyelink1000. Je vous renverrais vers des liens qui vous apporteront des informations supplémentaires.

1- Installation du système Eyelink1000

Architecture générale



Installation du système eyelink1000

Concernant l'installation du système EyeLink 1000, je vous renvoie à la partie références et lien utiles (manuel installation p16). Il convient ensuite de choisir le type de lentille adaptée à notre type d'expérimentation (manuel d'installation p 23). Il faut alors correctement placer la caméra devant l'écran du "display PC" (manuel d'installation p29). Si vous utilisez la mentonnière, elle doit être réglée de telle sorte que le support de menton soit horizontalement aligné avec le centre de l'écran du "Display PC"(manuel d'installation p32). Dans le cas où le sujet est en mode "tête libre" (avec pastille sur le front pour compensation des mouvements de tête), il faut alors que le regard "horizontal" du sujet porte sur le 3eme quart supérieur de l'écran (en considérant le bas de l'écran comme étant l'origine) (manuel d'installation p33)

2- Premier démarrage

On allume le "host PC", on choisit la partition Eyelink qui lance un programme qui s'exécute sous DOS (manuel installation p73).
En cliquant sur le menu "setup cameras" on doit pouvoir visualiser l'image transmise par la caméra à l'host PC.
Il faut paramétrer un fichier de configuration "PHYSICAL.INI" qui se situe sous C:\ELCL\EXE.
Ainsi pour accéder à ce fichier il faut déjà quitter le programme EyeLink (onglet exit), on se positionne ensuite sous C:\ELCL\EXE de la façon suivante (manuel installation p78) :

CD C:\ELCL\EXE
ATTRIB -R PHYSICAL.INI // *permet de modifier le fichier*
EDIT PHYSICAL.INI

On peut alors paramétrer les dimensions de l'écran du "display PC " ainsi que la position de l'origine et enfin la distance sujet - écran (manuel installation p76) . Il faut ensuite sauver les nouveaux paramètres.
NB: Il est conseillé de sortir sans sauver si vous avez fait une erreur que vous n'arrivez pas à corriger dans le fichier physical.ini.
Une fois sortie du fichier de configuration physical.ini, on écrit dans l'invite DOS (manuel installation p79) :

ATTRIB +R PHYSICAL.INI // *On re-protége en écriture le fichier.*

Il est aussi possible de paramétrer ce fichier de configuration en se positionnant dans la partition Windows, ce qui, à priori, semble plus simple (manuel installation p78) .
On peut ensuite revenir au programme de la partition EyeLink via les commandes suivantes:

CD C:\ELCL\EXE [enter]
ELCL.EXE [enter]

3- Installation des logiciels et configuration du "display PC"

Comme vous avez pu vous en apercevoir sur le schéma de l'architecture générale, le "display PC " est utilisé pour réaliser des paradigmes expérimentaux (via E-Prime par exemple) tout en contrôlant et dialoguant avec le système EyeLink 1000 et plus particulièrement "l'Host PC". Il est nécessaire d'installer sur le « display PC »:

- E-prime ainsi que
(voir sur site Sr research> support >download pour les logiciels suivants)
- Eyelink developper kit for windows/Mac/linux (dans la rubrique eyelink display soft)
- Eyelink data viewer (si vous l'utilisez - dans la rubrique Data analysis du support du site)

Il est essentiel de configurer les paramètres TCP/IP de votre ordinateur (Display PC) afin que votre "display PC" puisse communiquer avec "l'Host PC" via le lien Ethernet (manuel installation p81).

Présentation générale de "l'Host PC"

Le logiciel installé sous DOS sur "l' Host PC" intègre en effet toutes les fonctionnalités nécessaires pour configurer le mode de capture de l'eye tracker mais aussi réaliser de la capture des points de regards (manuel p12) . Je rappelle que l'on peut accéder à ce logiciel via les commandes DOS à partir de l'invite de commande sous DOS:

CD C:\ELCL\EXE [enter] ELCL.EXE [enter]

L'EyeLink 1000 peut fonctionner de 2 façons différentes (manuel p20) ;
En mode "Link", c'est-à-dire que l'eye tracker peut être contrôlé par le "Display PC" via un lien Ethernet. Avec une programmation appropriée, il est théoriquement possible d'avoir un contrôle complet de l'eye tracker via le "Display PC". Le "Display PC" envoie en effet des commandes via le lien ethernet au logiciel installé sous DOS sur "l'Host PC.
En mode "standalone" (évoqué lors de la présentation de l'architecture générale), dans ce mode l'eye tracker est un système indépendant contrôlé par un opérateur via le logiciel installé sous DOS dans le "Host PC".

NB: Appelons interface logicielle de l'EyeLink1000 le logiciel installé sous DOS sur "l'Host PC" permettant le contrôle et la configuration de l'eye tracker.
Les fonctionnalités et le mode d'emploi de cette interface logicielle sont très largement décrits dans le manuel (manuel p21) . Je vais uniquement présenter une des fonctionnalités du "setup" de la caméra (the camera setup screen) L'interface logicielle de l'EyeLink 1000 donne accès au menu Setup de la caméra (Camera SetUp). C'est dans ce menu que l'on va régler la focale de la camera, cela nécessite bien entendu d'avoir un sujet dans l'objectif de la caméra (manuel p54) . Dans un premier temps, il convient de vérifier que le sujet est bien positionné (en hauteur) par rapport à l'objectif de la caméra, des précisions sont apportées dans le manuel (manuel p60) . Ensuite on règle la focale de la caméra en agissant manuellement et lentement sur son objectif. Les caractéristiques d'un bon réglage sont décrites dans le manuel (manuel p62). Il faut ensuite régler la précision de la détection de la pupille (Setting pupils Thresholds) (manuel p74) ainsi que la qualité de la réflexion cornéenne (Setting the Corneal Reflection CR Threshold) (manuel p75).

Des indications précises sont données dans le manuel concernant la calibration, en particulier les moyens d'obtenir une calibration de qualité afin d'obtenir des enregistrements précis et fiables par la

suite (manuel p80) . Des informations sur la correction de la dérive (Drift correction) sont aussi apportées dans le manuel (manuel p97) .
Nous allons réaliser les expérimentations suivantes en utilisant l'eyetracker en mode "link". ***L'eye tracker sera donc contrôlé par un "Display PC" au travers du logiciel E-Prime***. La réalisation de l'exemple s'effectuera en mode desktop (remote) - monocular avec une lentille adaptée de 16mm, seul les points de regards de l'oil droit seront observés et le sujet aura la possibilité de bouger la tête (tête libre) tout en respectant certaines limites de mouvements.

4- Procédure afin de réaliser une expérimentation avec E-prime et l'eyelink1000 :

E-prime et eyelink1000

Si vous souhaitez faire en sorte que votre programme E-Prime fonctionne avec l'eyetracker il faut opérer quelques adaptations sur votre programme. Ces adaptations consistent à rajouter des scripts e-prime de façon à ce que le programme E-prime puisse communiquer avec l'eyetracker.
Ces ajouts sont précisément décrit en anglais dans le pdf « E-prime integration ».
Les principaux scripts E-prime que vous allez modifier sont les scripts « startRecording » et « stop recording ».

Prenons l'exemple du programme E-prime exampleEprimeET.es2 qui est un programme simple réalisé par SR-research et qui à été légèrement modifié par nos soins.

5- Exemple «exampleEprimeET.es2 »

Le script de connection « elconnect »
Ce script va permettre principalement d'établir une connection entre Eprime et l'eyetracker

Le script de configuration de la caméra et de calibration « elCameraSetup »
Ce script va permettre de paramétrer la caméra (réglage manuel de la focale), de renseigner le fichier de sauvegarde des données (test.edf par défaut que vous pouvez modifiez si vous le souhaitez) et surtout de réaliser la calibration (avec 9 points ici : HV9)

Le script de drift correct « DriftCorrect » permet de corriger (en général juste avant le commencement de l'expérimentation) la dérive. Cela s'effectue en général avec une croix de fixation que l'on demande au sujet de fixer, l'expérimentation ne peut commencer que si le point de regard du sujet coïncide avec ce point de fixation.

Le script startRecording

NB : les lignes de code sont écrites en bleu et les lignes de code en commentaire (donc non effective) sont en vert

tracker.sendMessage "TRIALID " & TrialList.GetCurrentAttrib("trialid")
Cette commande permet d'envoyer à l'eyetracker l'identifiant de l'essai. Cet identifiant est définit dans la liste réalisée par l'utilisateur (voir dans trialList la colonne TrialId). Cet identifiant pourra ensuite être visible dans le fichier de donnée de l'eyeTracker.

tracker.sendCommand "clear_screen 0"
Cette commande permet de nettoyer l'écran?

tracker.sendCommand "draw_box " & Display.XRes/2 -50 & " " & Display.YRes/2 - 50 & " " & Display.XRes/2 + 50 & " " & Display.YRes/2 + 50 & " 7"
Cette commande permet d'afficher un carré d'affichage avec des dimensions précises sur l'écran de l'expérimentateur ?

'tracker.sendMessage "!V IMGLOAD FILL " &List.GetCurrentAttrib("imageName")
Cette commande permet d'avoir les stimuli visible dans le dataviewer (logiciel d'analyse), dans ce programme, il n'y-a pas d'image donc cette commande est mise en commentaire.

'tracker.BitmapToBackdrop List.GetCurrentAttrib("imageName"), 4
Cette commande permet d'envoyer une image sur l'écran de l'expérimentateur, elle est mise en commentaire ici

tracker.sendCommand "set_idle_mode"
Commande xxx?

Sleep 50
On impose un délai de 50 ms de façon à ce que le système de tracking soit prêt à lancer l'enregistrement

tracker.startRecording True, True, False, False
Cette commande permet de lancer l'enregistrement, les options sont à déterminer en fonction des besoins.

elutil.pumpDelay 100
Cette commande permet à l'EyeTracker d'enregistrer 100ms de données après la commande StartRecording de façon à enregistrer les données oculométriques 100ms avant l'apparition du premier stimulus.

Dim timeStart As Long
Cette commande permet de définir une variable

timeStart = elutil.currentTime()
Cette commande permet d'associer à la variable timeStart une valeur de temps (horloge ordinateur)

tracker.sendMessage "Simple_Start"
Cette commande permet d'envoyer un message dans le fichier d'enregistrement, ce message s'affiche juste avant l'affichage des données oculométriques enregistrées sur le fichier.

Le script StopRecording

Lorsque l'essai est terminé, on « rentre » dans le script StopRecording, ce script va permettre entre autre de calculer les moments d'apparition des stimuli. Cela va permettre de synchroniser les stimuli générés par E-Prime avec les données oculométriques enregistrées par l'EyeTracker.
Les variables ainsi calculées sont envoyées au fichier de données (fichier *.edf) via la commande « tracker.sendMessage ».
Par exemple si l'on souhaite envoyer au fichier de données .edf le moment d'apparition d'un stimulus, il faut dans un premier temps calculer le temps écoulé entre l'apparition de ce stimulus et le moment ou l'on « rentre » dans le script StopRecording (variable offset ci-dessous). On soustrait ensuite ce temps obtenu au temps absolu actuel. On obtient alors le temps d'apparition du stimulus (voir exemple dans le code ci-dessous).

```
Dim timeEnd As Long
timeEnd = elutil.currentTime()
Dim offset As Long
offset = (timeEnd - timeStart) - (simple.OnsetTime - simple.StartTime)
```

timeStart et timeEnd sont des temps absolus (respectivement correspondant au temps absolu du début de l'essai et au temps absolu de la fin de l'essai) alors que simple.OnsetTime et simple.StartTime sont des temps relatifs au début de l'essai. Il y-a une petite différence de temps entre le moment ou le stimulus *simple* est « lancé » (simple.StartTime) et le moment où le stimulus *simple* apparait effectivement à l'écran (simple.OnsetTime). Le calcul de l'offset (i.e. le temps écoulé entre l'apparition du stimulus et le temps actuel) tient donc compte de cette « petite » différence pour son calcul.

```
tracker.sendMessage  offset & "  Simple_Onset"
```

On envoi un message au fichier.edf précisant le moment d'apparition du stimulus *simple,* pour cela on associe au message « Simple_Onset » la valeur d'offset calculée (offset).
Le message obtenu dans le fichier.edf pourrait ressembler à cela :
MSG 3320279 20000 Simple_Onset
On a donc un message signifiant que l'apparition du stimulus *simple* à l'écran du sujet s'est effectuée au temps absolu 3320279-20000.

```
elutil.pumpDelay 100
tracker.stopRecording
```

On enregistre 100ms de données oculométrique supplémentaire (elutil.pumpDelay 100)
après la fin de l'essai pour être sur de ne pas « manquer » un enregistrement oculométrique à la fin de l'essai et on stoppe l'enregistrement (tracker.stopRecording).

Les commandes qui suivent permettent au logiciel DataViewer d'intégrer les messages et les commandes du fichier.edf.

```
Dim imgname As String
imgname = "img" & TrialList.GetCurrentAttrib("trialid") &".bmp"
Display.Canvas.SaveImage imgname
tracker.sendMessage "!V IMGLOAD FILL " & imgname
```

Cet ensemble de commande permet d'afficher une image en surimposition dans le logiciel DataViewer

```
tracker.sendMessage "!V TRIAL_VAR trial " & TrialList.GetCurrentAttrib("trialid")
tracker.sendMessage "!V TRIAL_VAR trialword " & TrialList.GetCurrentAttrib("trialword")
```

Ces 2 messages permettent au logiciel DataViewer d'identifier les caractéristiques de l'essai.

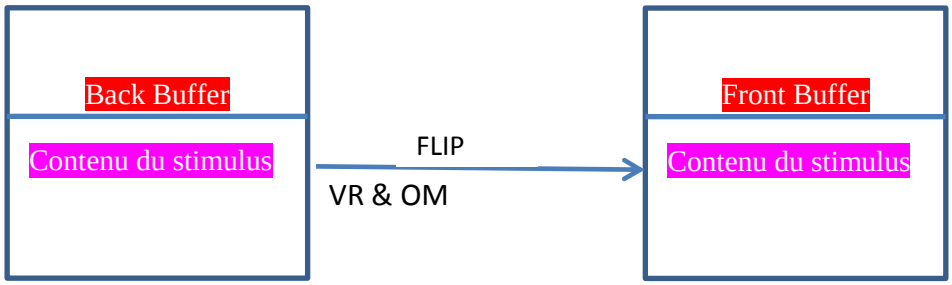
```
tracker.sendMessage "TRIAL_RESULT 1"
```

Cette commande permet de faire savoir au logiciel DataViewer que l'essai est terminé, DataViewer ne tiendra compte d'aucun message ou événements qui suivra cette ligne de commande.

Ils existent bien d'autres messages et commandes qui permettent à E_Prime de communiquer avec l'EyeTracker, voir le "Protocol for EyeLink Data to Viewer Integration" du « EyeLink Data Viewer User Manual »

6- Frequently Asked Question (FAQ)

1- ***Je cherche à synchroniser précisément (à la ms) l'apparition de mon stimulus sur l'écran vu par le sujet avec l'enregistrement des données oculométriques. Est-ce possible ?***
Réponse :



VR : Vertical Retrace, signal indiquant que l'affichage des données peut être effectué
OM : Onset Message, qui indique le moment de l'opération FLIP, ce message peut être envoyé par Experiment Builder, Matlab ect…***mais pas E-Prime***.
Flip : Transfert des données du Back Buffer vers le Front Buffer

Les données du stimulus sont stockées dans le Back Buffer (en vue d'être affiché plus tard), lorsque le VR est détecté par le programme (EB, matlab), les données du Back Buffer sont alors transférées dans le Front Buffer (FLIP). C'est à ce moment que l'on doit envoyer le message OM. Ce message est alors un indicateur précis du moment où le stimulus est visible sur le moniteur **à condition d'utiliser un moniteur CRT.** Les moniteurs LCD ont un temps de réponse variable en fonction de l'écran LCD. Certains moniteurs LCD 120 Hz comme le Viewsonic VX2265WM ou Samsung Syncmaster 2233RZ ont un temps de réponse très faible (de l'ordre de 2 ms) mais pour d'autres moniteurs, ce délai peut atteindre les 20 ms. Ainsi le choix du moniteur dépend de la précision que l'on désire et de la tâche effectuée.

2- ***Quels sont les paramètres de détection des saccades dans l'EL1000 ?***
Le système d'analyse en temps réel de l'EL1000 a deux types de paramètre : un mode « normal » et un mode dit « sensible ». Ces paramètres sont des seuils sur la vitesse et l'accélération du point de regard, la détection des saccades se base sur ces seuils.

Mode normal: Velocity Threshold: 30 deg/s and acceleration threshold: 8000 deg/s²
Mode sensible: Velocity Threshold: 22deg/s and acceleration threshold: 3800 deg/s²

Afin de savoir quelle sensibilité est utilisée pour analyser en ligne les données, allez dans le fichier eye*.log situé dans le répertoire /elcl/exe/ et chercher le texte « select_parser_configuration », si il est à 0 alors le mode normal est utilisé sinon c'est le mode high.