

Guide de survie pour



Par Dominic Charbonneau

Table des matières

1. [Présentation](#) :
 - 1.1. [Le logiciel](#);
 - 1.2. [Les composantes](#);
 - 1.3. [L'appareillage](#).
2. [E-Studio](#) :
 - 2.1. [L'interface](#);
 - 2.2. [Les objets](#) :
 - 2.2.1. [Inline](#);
 - 2.2.2. [TextDisplay](#);
 - 2.2.3. [ImageDisplay](#);
 - 2.2.4. [SoundOut](#);
 - 2.2.5. [Slide](#);
 - 2.2.6. [FeedbackDisplay](#);
 - 2.2.7. [List](#);
 - 2.2.8. [Wait](#);
 - 2.2.9. [Procedure](#);
 - 2.2.10. [Label](#);
 - 2.2.11. [PackageCall](#).
 - 2.3. [Concevoir une expérience](#):
 - 2.3.1. [Du plan à E-prime](#);
 - 2.3.2. [Définir les objets](#);
 - 2.3.3. [Définir un essai](#).
3. [E-Run](#) :
4. [E-Recovery](#) :
 - 4.1. [Pourquoi l'utiliser](#);
 - 4.2. [Comment l'utiliser](#).
5. [E-Merge](#) :
6. [E-DataAid](#):
7. [Factor table wizard](#) :
 - 7.1. [Utilité](#);
 - 7.2. [Comment ça fonctionne](#);
 - 7.3. [Les dangers](#).
8. [Aide](#) :
9. [Programmation VB](#) :
 - 9.1. [Sélection et boucles](#);
 - 9.2. [Les attributs](#).
10. [Points importants](#);
11. [Conclusion](#).

SECTION 1 : PRÉSENTATION

E-prime monte en popularité. De plus en plus de laboratoire de recherche en psychologie l'utilisent pour confectionner leurs expériences. Ainsi ce texte se veut une rampe de lancement vers l'univers de la conception d'expériences avec E-prime. Il est conçu dans une optique de simplicité et d'efficacité afin de permettre à des novices dans le domaine de comprendre les bases de ce logiciel et ainsi mieux l'explorer par la suite.

Section 1.1 : Le logiciel

E-prime est un logiciel de conception d'expériences au même type que Mel Professional, PsychLab et PsyScope. Son utilisation s'étend dans tous les domaines de la recherche fondamentale qui touchent aux ordinateurs. Ce logiciel comporte de nombreux avantages :

- Il est programmé à l'aide de Visual Basic ce qui lui permet d'être compatible avec de nombreux autres logiciels tel Microsoft Word, Microsoft Excel, Mathematica et SPSS;
- Son interface graphique est plus conviviale que bien d'autres logiciels, permettant à un novice autant qu'à une personne habituée à d'autres logiciels de s'y retrouver facilement;

- Il est « WYSIWYG », c'est-à-dire, « ce que tu vois est ce que tu auras » (textuellement : *What You See Is What You Get*);
- Il permet de visualiser toute l'expérience dans une seule fenêtre;
- Permet l'utilisation de plusieurs modalités de collecte de données (souris, boîte de réponse, clavier, électroencéphalogramme, etc.).

Cependant il comporte aussi des désavantages comme vous pourrez le voir dans votre exploration de ce logiciel. Un de ces désavantages est la difficulté à l'appivoiser. Ce logiciel vient avec trois livres, qui sont fastidieux, souvent difficile à comprendre et écrits en anglais ce qui peut rendre la tâche d'un nouvel assistant de recherche francophone plus difficile. Ainsi ce petit cahier veut régler ce désavantage majeur.

Dans ce cahier, nous ne traiterons pas en profondeur de chacune des subtilités de ce logiciel, ceci étant fait dans les manuels fournis avec le logiciel. De plus, cela n'aiderait pas à l'objectif de simplification de ce cahier. Ainsi pour plus de détails que ce qui est fourni dans ce cahier, d'autres sources de référence sont citées plus loin dans ce texte.

Section 1.2 : Les composantes

Le logiciel E-prime est divisé en plusieurs sous programmes :



E-studio est la partie principale de ce logiciel. C'est ce sous-programme qui permet de concevoir l'expérience proprement dite.



E-Merge permet de fusionner de nombreux fichiers ensemble pour n'avoir qu'un seul fichier de données.



E-Data-Aid est un chiffrier de données similaire à Excel ou SPSS, mais moins complexe. Il permet de visualiser les données et les manier pour qu'elles prennent l'apparence voulue avant d'être exportées dans le logiciel d'analyse.



E-Recovery est une grande innovation de PSTNET, ce sous programme permet de récupérer les données d'un participant même si celui-ci n'avait pas terminé sa session.



E-Run permet d'exécuter l'expérience sur d'autres ordinateurs sans avoir à installer la licence complète de E-prime.

Section 1.3 : L'appareillage nécessaire

Pour utiliser E-prime, inévitablement vous devez vous procurer la licence auprès du fournisseur. Cette licence fournit un logiciel avec une clef physique (à connecter au port USB ou parallèle) qui vous permet d'utiliser E-prime sur votre ordinateur. Sans cette clef, vous ne pourrez installer le programme et si vous l'installez avec une clef empruntée, vous aurez plusieurs problèmes à l'utiliser. Pour installer ce logiciel sur votre ordinateur, vous devez posséder au minimum :

- Windows 95;

- Processeur 120 Mhz Pentium;
- 100 MB d'espace;
- CD-ROM;
- 16-32 MB de RAM;
- Carte graphique compatible avec DirectX;
- Port parallèle ou USB.

Les informations et prix de ce logiciel sont disponible sur le site Web du logiciel :

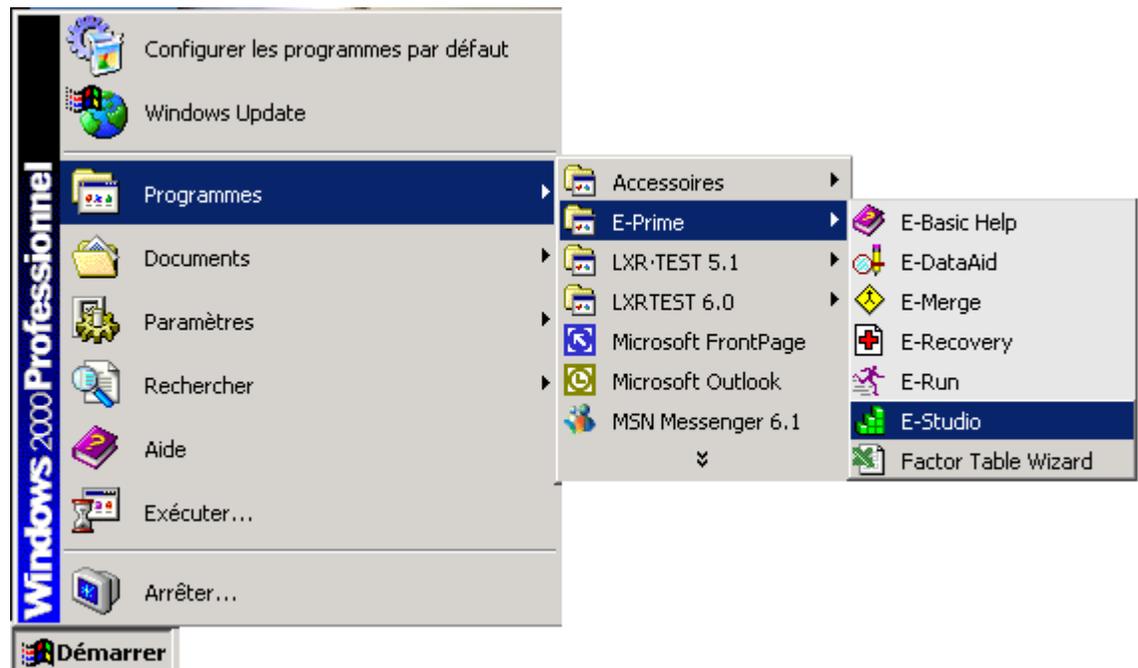
<http://www.pstnet.com/products/e-prime/>

SECTION 2 : E-STUDIO

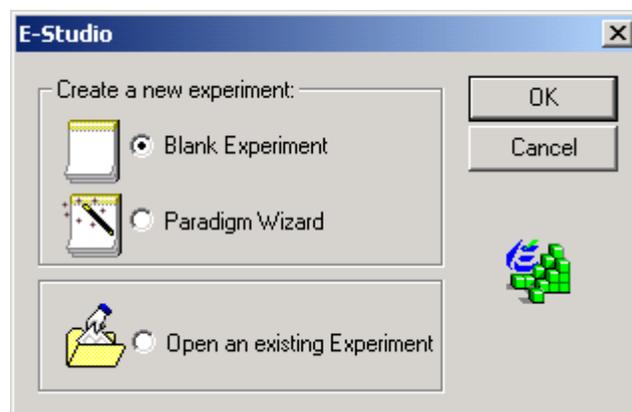
E-studio est la partie principale de ce logiciel. C'est ce sous-programme qui permet de concevoir l'expérience proprement dite

Section 2.1 : L'interface

Une fois installé convenablement sur votre ordinateur, tous les sous-programmes sont accessibles via le menu démarrer, dans la liste des programmes. Sélectionner la composante **E-studio** :



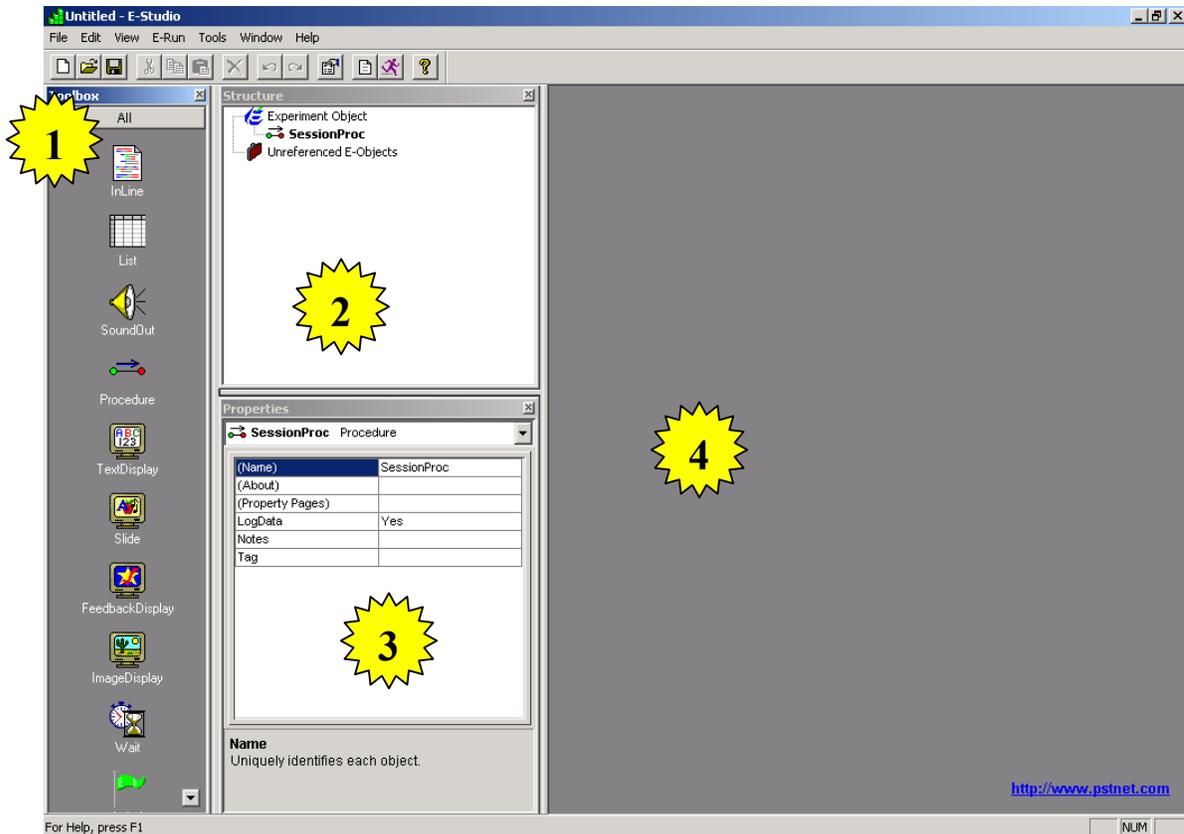
À l'ouverture, il vous est demandé le type de travail que vous voulez faire.



Il est possible de créer une nouvelle expérience, soit à partir d'une expérience vierge (*Blank Experiment*), ou avec le *Paradigm Wizard*. Cette dernière option est moins conseillée car malgré que le gros du travail est déjà accompli, vous vous retrouvez avec une expérience que vous ne connaissez que très peu (on est jamais mieux servi que par

soi-même). Il est aussi possible d'ouvrir une expérience déjà existante (*Open an existing Experiment*).

L'interface comporte quatre sections importantes.



La première (1) fenêtre est la boîte à outils (*ToolBox*) qui comporte l'ensemble des objets que l'utilisateur peut utiliser dans son expérience. Ceux-ci seront explicités plus bas. La deuxième (2) fenêtre est la *Structure* hiérarchique de l'expérience. Ceci permet d'avoir une vue d'ensemble du déroulement de l'expérience. Lors de l'exécution du programme, E-prime descend séquentiellement cette hiérarchie exécutant chacun des objets un à la suite de l'autre. La troisième (3) fenêtre (*Properties*) affiche les propriétés de l'objet actif. Il est cependant déconseillé de l'utiliser de cette façon, cela augmente les

chances de modifier les propriétés d'un mauvais objet sans s'en rendre compte. Nous verrons plus loin une autre façon d'atteindre ces propriétés en minimisant les chances de se tromper. La dernière fenêtre de l'interface (4) est la fenêtre de travail (*WorkSheet*). C'est dans cette fenêtre que l'expression WYSIWYG (*What You See Is What You Get*) prend tout son sens. Cette fenêtre affiche la partie graphique d'un objet (s'il en a une). Par exemple un contrôle de texte montrera dans cette fenêtre le texte qui sera présenté aux participants.

Section 2.2 : Les objets

Traitons maintenant des objets disponibles dans la boîte à outils. Dans cette section, chacun des objets sera décortiqué. Il a été dit plus tôt qu'à chaque objet venaient plusieurs propriétés. Ces propriétés sont toujours accessibles en cliquant sur l'icône des

propriétés : . Cette façon de procéder, nous contraint à modifier uniquement les propriétés de l'objet sur lequel nous travaillons, ce qui minimise les chances de modifier les propriétés des autres objets.

Section 2.2.1 : Inline

Le *InLine*  est une page de code qui peut s'insérer n'importe où dans l'expérience. Cette page de code permet d'ajouter de la programmation Visual Basic à l'expérience afin d'accomplir des tâches qui sont parfois trop complexes à réaliser avec

les menus. Cette page n'a pas de propriétés mis à part la propriété **Log data** qui indique si cet élément va ou non enregistrer des données.

Section 2.2.2 : TextDisplay



Le *Textdisplay* est un éditeur de texte. Il permet d'afficher aux participants du texte au choix du concepteur. Cet objet a une interface graphique qui permet de visualiser ce qui sera présenté aux participants. Ses principales propriétés sont résumées dans le tableau suivant.

Nom :	Description :
Text	Dans l'onglet <i>Général</i> , affiche le texte qui sera présenté.
Le reste de l'onglet <i>Général</i> ainsi que l'onglet <i>Frame</i> servent à spécifier les propriétés de ce texte (couleur, position etc...)	
*Duration	Dans l'onglet <i>Duration/Input</i> , cette propriété détermine la durée d'affichage de l'objet.
*Data Logging	Dans l'onglet <i>Duration/Input</i> , dans le cas où on attend une réponse du participant, il faut spécifier dans cette case le type d'enregistrement (<i>standard, custom, etc.</i>). Chacune des valeurs que peut prendre cette case se réfèrent à l'onglet <i>Logging</i> ou le détails de ce qui est enregistré est donné.
*Input Mask	Dans l'onglet <i>Duration/Input</i> cette section gère toute la collecte de donnée.
*Devices	Dans l'onglet <i>Duration/Input</i> dans la section Input Mask , cette propriété détermine quel périphérique (clavier, souris, etc...) sera utilisé pour collecter la réponse. On peut ajouter un périphérique en cliquant sur le bouton <input type="button" value="Add..."/>
*Allowable	Dans l'onglet <i>Duration/Input</i> dans la section Input Mask , cette propriété sélectionne les touches du périphérique qui pourront être utilisées par le participant, les autres étant désactivées.
*Correct	Dans l'onglet <i>Duration/Input</i> dans la section Input Mask , cette propriété spécifie la bonne réponse afin d'enregistrer le pourcentage de bonnes réponses.
*Time Limit	Dans l'onglet <i>Duration/Input</i> dans la section Input Mask , cette propriété spécifie le temps limite de cet objet si jamais le participant ne répond pas.
*End action	Dans l'onglet <i>Duration/Input</i> dans la section Input Mask , cette propriété détermine l'action qui sera prise une fois la réponse obtenue. Il peut soit se

	terminer, soit aller à un objet <i>Label</i> ou terminer la durée de l'objet (mis à none).
--	--

*Signifie que ces propriétés se retrouvent aussi dans d'autres objets.

Section 2.2.3 : *ImageDisplay*



L'*imageDisplay* permet d'afficher une image. Il possède

pratiquement les mêmes propriétés que le l'objet *textDisplay* mis à part qu'on doit lui spécifier l'image qu'on veut afficher dans la propriété ***Filename***.

Section 2.2.4 : *SoundOut*



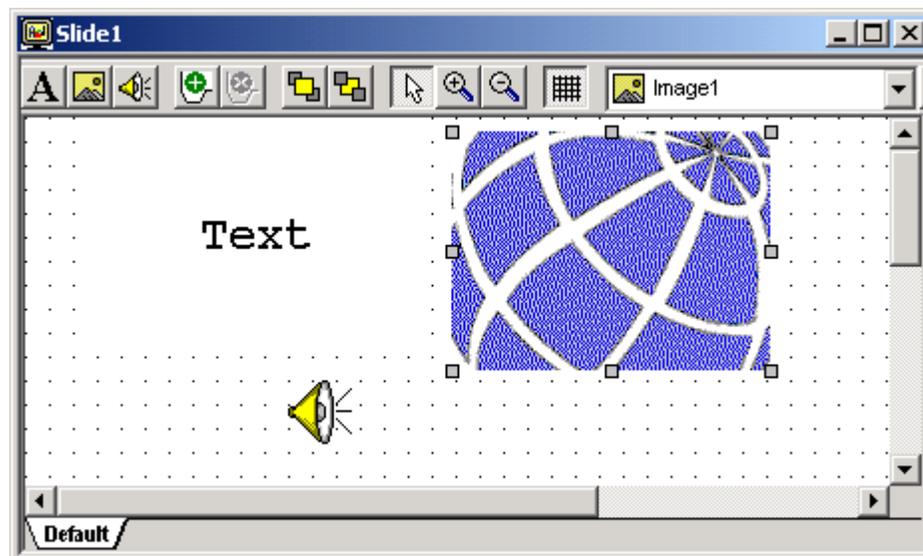
Le *SoundOut* permet d'émettre un son. Cet objet n'a pas de portion graphique. Ainsi en double cliquant dessus, nous accédons directement à sa fenêtre des propriétés. Les principales propriétés sont données dans le tableau suivant.

Nom :	Description :
<i>Filename</i>	Dans l'onglet <i>General</i> , cette propriété permet de sélectionner le fichier WAV qui sera joué.
<i>Max Length</i>	Dans l'onglet <i>General</i> , détermine la longueur maximale que le son peut avoir.
<i>Duration</i>	Dans l'onglet <i>Duration/Input</i> , cette propriété détermine la longueur de présentation du son. Par exemple malgré que le fichier puisse durer 4 secondes, si la valeur mise dans <i>Duration</i> en ms est de 300, alors le son se terminera après 300 ms.

Section 2.2.5 : *Slide*



L'objet *Slide* est très polyvalent et permet de jumeler plusieurs objets en même temps. Par exemple il est possible de présenter au participant plusieurs fenêtres de texte, images et sons en même temps à l'aide de cet objet. Ses propriétés sont similaires à ceux du *Textdisplay*, mis à part la partie texte. Pour mettre plusieurs sous-objets dans cette feuille il suffit de cliquer sur l'icône correspondant (Texte: , Image: , Son: ) au haut de la fenêtre et de le déposer à l'endroit désiré sur l'interface. À ce moment là, ce nouvel sous-objet agit comme un s'il était un objet à part entière.

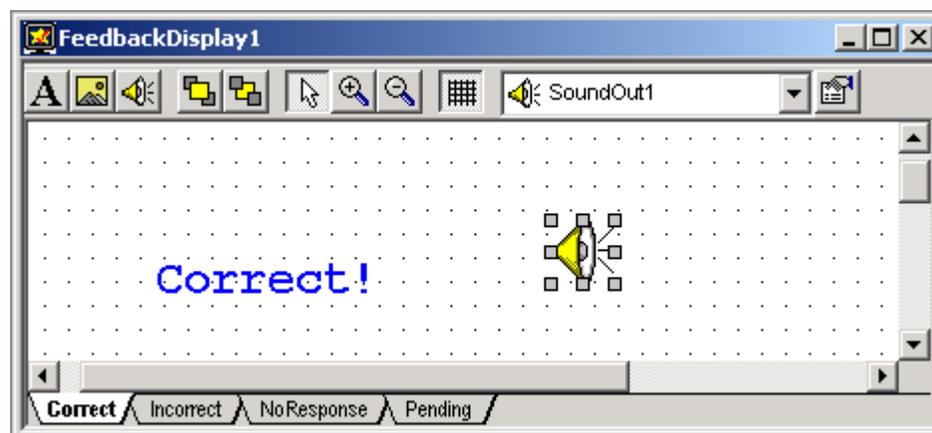


Section 2.2.6 : *FeedBackDisplay*



L'objet *FeedbackDisplay* permet la rétroaction au participant. Il comporte de multiples possibilités de rétroaction comme les temps de réponse, les

pourcentages de réponses correctes ainsi que bon nombre de mesures de dispersions tel l'écart type, le maximum, le minimum et la moyenne. Ces mesures sont décrites à la page 82 du manuel « Reference Guide ». L'objet *FeedBackDisplay* a une interface graphique similaire à celle de l'objet *Slide*, mis à part qu'il contient trois panneaux correspondant aux cas où la réponse géré par l'objet spécifié dans la propriété ***Input Object Name*** est « Correct », « Incorrect » ou « Absent ». Sur chacun de ces panneaux (atteignable par les onglets au bas de l'objet) il est possible de spécifier le comportement de celle-ci individuellement tout comme dans un objet *Slide*. Par exemple on peut insérer un son dans le panneau Correct comme le montre l'exemple ci bas, mais ne pas en inclure dans les autres panneaux.



Section 2.2.7 : List



L'objet *List* est le plus important de tous. Il sert de plaque tournante de l'expérience. Cet objet spécifie le nombre de blocs, d'essais, ainsi que la hiérarchie qui

sous-tend l'expérience. Dans un objet *List* il est possible de définir toutes les conditions ainsi que l'ordre dans lequel nous voulons les présenter. Son fonctionnement est simple. Comme il a été dit, lors de l'exécution du programme, E-prime descend séquentiellement la hiérarchie donnée dans la fenêtre de Structure. Lorsqu'il arrive à un objet *List*, à chaque essai, il choisira une ligne de cette liste et exécutera la procédure sous jacente selon les paramètres de cette liste.

L'objet *List* comporte de nombreuses propriétés. Le tableau suivant spécifie les propriétés de première importance. Les propriétés qui ne sont pas décrites sont habituellement déjà spécifiées et gèrent des fonctions plus complexes de l'objet *List*. Pour de plus amples informations sur ces propriétés, référez-vous aux manuels fournis avec le logiciel.

Nom :	Description :
<i>Order</i>	Dans l'onglet <i>Selection</i> , cette propriété permet de spécifier le mode de sélection des items dans la liste.
<i>Reset Sampling</i>	Dans l'onglet <i>Reset/Exit</i> , cette propriété spécifie après combien de sélection dans la liste les items déjà sélectionnés peuvent être remis dans la liste.
<i>Exit List</i>	Dans l'onglet <i>Reset/Exit</i> , cette propriété spécifie après combien de sélection dans la liste on doit sortir de la liste pour passer à l'objet suivant.

Lorsqu'on double click sur l'objet *List*, nous accédons à la fenêtre générale de celle-ci. L'objet *List* n'a pas d'interface graphique visible au participant.

ID	Weight	Nested	Procedure	Premier	Deuxieme
1	1		EssProc	arbre	arbre
2	1		EssProc	heure	heure
3	1		EssProc	divan	divan
4	1		EssProc	livre	livre
5	1		EssProc	vieux	vieux
6	1		EssProc	pomme	pomme
7	1		EssProc	terre	terre

L'objet *List* se comporte de façon similaire à un fichier Excel. Dans la barre de menu du haut, les quatre premiers icônes permettent d'ajouter, une ligne, plusieurs lignes, une colonne et plusieurs colonnes respectivement. Le suivant spécifie si nous voulons que la rubrique « Summary » sous-jacentes à ces icônes soit visible ou non. Les deux derniers icônes servent à supprimer une ligne ou une colonne. Comme il a été dit plus haut, l'objet *List* est la plaque tournante de toute l'expérience. L'attribut **Weight** spécifie le nombre de fois que la ligne sélectionnée par le programme sera exécutée. **Nested** Spécifie si une autre liste sous-jacente doit être incluse ou non. Cette propriété est utile lorsque nous voulons que les essais ou les blocs utilise des stimuli de différentes listes. Enfin, **Procedure** détermine le nom de la procédure ce qui permettra de passer à un niveau inférieur de la hiérarchie. Ces trois attributs sont les attributs de base toujours présent lors de l'ouverture d'un nouvel objet *list*. Il est possible d'ajouter autant

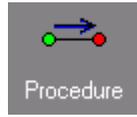
d'attributs que vous voulez et d'en spécifier le titre en double-cliquant dessus. C'est dans cette liste que les conditions seront spécifiées. Dans l'exemple ci-haut, la liste nommée « EssProc » spécifie les attributs pour chaque essai. Ainsi, l'expérience comporte 40 essais. À chaque essai, e-prime choisit une ligne selon la propriété **Order** et exécute la procédure spécifiée dans la colonne **Procedure**. Les attributs spécifiés à la droite de cette colonne sont les diverses conditions qui gèrent chacun des essais. Ainsi, dans l'exemple ci-haut, dans la procédure EssProc, à un certain point il sera demandé les attributs Premier et Deuxième correspondant à la ligne où E-prime sera rendu. Ainsi par cet objet, nous pouvons spécifier des caractéristiques qui seront les mêmes tout au long de l'expérience. Cette notion sera maintes fois reprise lorsque viendra le temps de faire des exemples.

Section 2.2.8 : *Wait*



L'objet *Wait* permet d'établir un délai entre deux objets. Cet objet ne comporte pas de partie graphique, il ne fait que suspendre l'exécution du programme pendant le temps désiré. Sa principale propriété est **Duration** dans l'onglet *Duration/input*. Cette propriété déterminera le temps d'attente avant de passer au prochain objet. Étant donné que cet objet n'a pas de partie graphique, pendant le temps d'attente l'ordinateur continue d'afficher ce qui était présent avant le début de cet objet.

Section 2.2.9 : *Procedure*



L'objet *procedure* est habituellement mis automatiquement lorsqu'on spécifie un nom dans la colonne **Procedure** dans un objet *List*. Cet objet permet de passer à un niveau inférieur dans la hiérarchie (par exemple d'un bloc à un essai). Ainsi cet objet n'a que la propriété **Log data**. Cette propriété est utile dans les cas où nous désirons que tout ce qui est inclus dans cette procédure ne soit pas enregistré.

Section 2.2.10 : *Label*



L'objet *Label* est un marqueur de position qui est utile lors de conditions qui sont fonctions de la réponse du participant. Par exemple, on peut vouloir que le participant saute un objet s'il répond trop rapidement. Cet objet n'a pas de partie graphique, il suffit de le positionner dans la hiérarchie pour en définir les propriétés. Cependant, il est possible d'utiliser l'objet *Label* qu'une seule fois par procédure et il est impossible de « sauter » d'une procédure à l'autre. L'objet *Label* doit être initialisé et placé dans la même procédure.

Section 2.2.11 : *PackageCall*



L'objet *PackageCall* est utile pour annexer à E-prime des périphérique ou logiciel qui ne viennent pas automatiquement avec le logiciel. Par exemple pour synchroniser un électroencéphalogramme avec e-prime, cet objet sera utile.

Section 2.3 : Concevoir une expérience

Dans cette section nous allons concevoir une expérience simple afin de mettre en pratique les nombreux éléments que nous venons de voir. Il est fort possible que vous ayez à revenir à la précédente section à plusieurs reprises pour mieux consolider des notions sur lesquelles vous avez moins porté attention.

Lorsque viens le temps de programmer une expérience, il est essentiel de s'être fait d'abord un plan. Dans ce plan devrait être inclus

- le nombre de blocs;
- le nombre d'essais;
- comment se constitue un essai;
- le temps de présentation de chaque affichage;
- comment se constituent les conditions;
- les modalités de présentation des stimuli (aléatoire, séquentiel, contrebalancé etc.).

Nous allons ici programmer une expérience même-différent. Nous avons choisit cette expérience car elle est utile dans beaucoup de cas de comparaison de stimuli et elle constitue une bonne introduction pour E-prime.

Construisons d'abord le plan. Disons que nous voulons présenter au participant 400 essais divisés en 10 blocs de 40 essais. Un essai est composé d'un point de fixation de 800 ms, d'une première chaîne de caractères présentée pendant 50 ms, suivie d'un masque de 34 ms et d'un deuxième mot présenté 100 ms. Suivra ensuite un affichage test

où le participant devra inscrire « 1 » s'il s'agit du même mot et « 2 » s'il s'agit d'un mot différent. Une rétroaction sera ensuite donnée. Ainsi le plan doit ressembler à ceci :

<u>Objets E-prime :</u>	<u>Durée des objets :</u>
Texte d'introduction	Barre d'espacement
Liste de 10 blocs	
Texte de début de bloc	Barre d'espacement
Liste de 40 essais	
Point de fixation	800 ms.
Premier mot	50 ms.
Masque	34 ms.
Deuxième Mot	Réponse
Rétroaction	Barre d'espacement
Texte de fin de bloc	Barre d'espacement
Texte de remerciements	Barre d'espacement

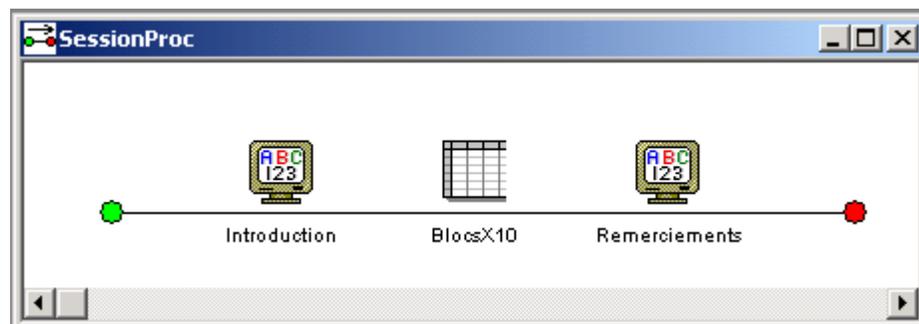
Section 2.3.1: Du plan à E-prime

Les textes d'introduction et de remerciements seront inclus dans l'expérience à l'aide d'objet *TextDisplay* tout comme les textes de début de bloc et de fin de bloc. Le point de fixation, le premier mot, le masque, le deuxième mot et l'affichage test seront tous inclus à l'aide d'objet *Slide*, tandis que la rétroaction sera donnée à l'aide de l'objet *FeedbackDisplay*. Comme il est dit plus haut, chaque passage à un niveau inférieur de la hiérarchie soit lorsqu'on veut compléter 10 blocs ou 40 essais, sera inclus dans l'expérience à l'aide d'objet *List*.

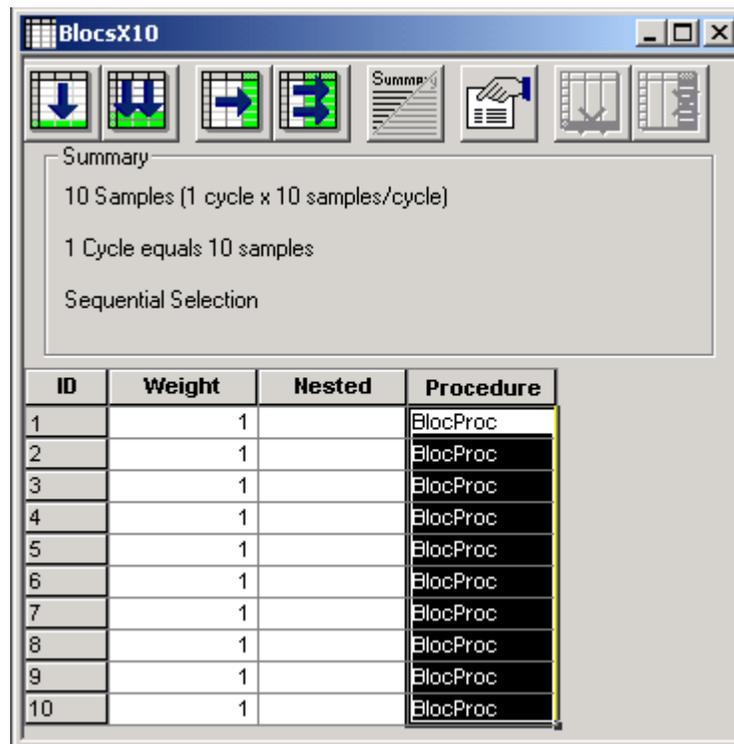
Lorsqu'on ouvre l'expérience tout les objets sont dans la boîte à outils et rien ou presque ce retrouve dans la fenêtre de structure. La première étape consiste à ouvrir la ligne du temps *SessionProc* et d'y glisser (« *Drag and Drop* ») les éléments voulus. Il est préférable de glisser les éléments de la boîte à outils dans l'ordre de gauche à droite du

plan. Ainsi nous débuterons par glisser que trois éléments soit les textes d'introduction, de remerciement et une liste qui nous servira pour les 10 blocs.

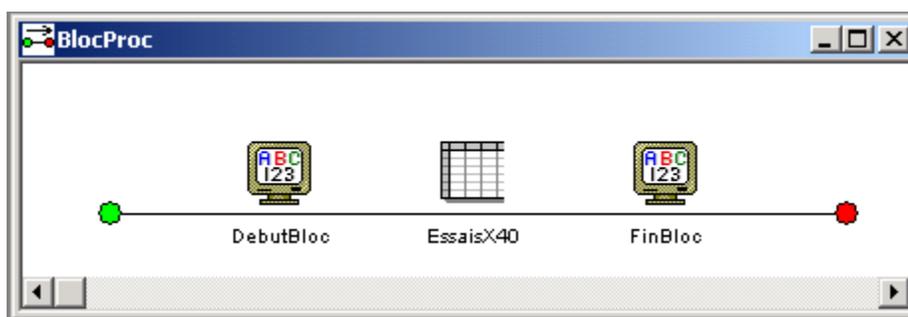
On peut ensuite renommer ceux-ci avec des noms plus indicatifs de leur utilité tel que montré dans la figure ci-dessous.



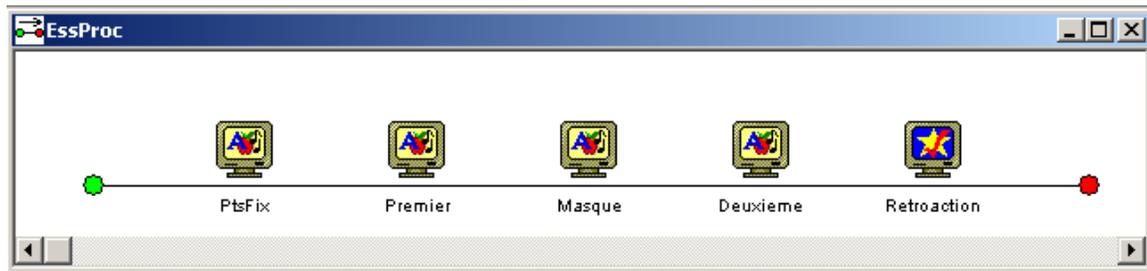
Ensuite, puisque nous voulons créer un niveau inférieur dans la hiérarchie, il faut implémenter une nouvelle procédure. Comme il est indiqué plus haut, ceci se fait à l'aide de l'objet *List*. Commençons donc par définir celle-ci. D'abord nous ajoutons 9 lignes à celle déjà existante pour former une liste de 10 lignes (10 blocs). C'est dans la colonne ***Procedure*** que nous créons une nouvelle procédure que nous nommerons « BlocProc ». Il suffit ensuite de généraliser à tous les niveaux.



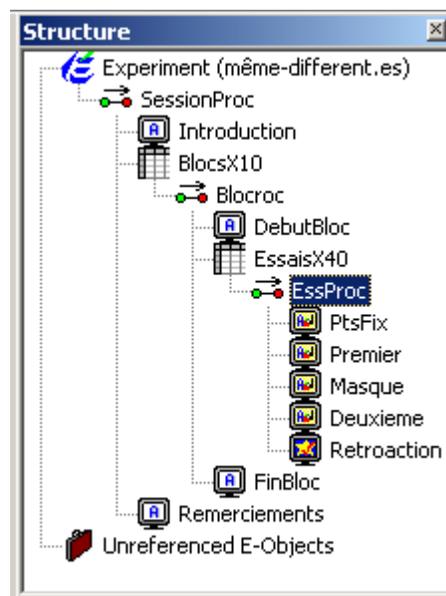
Nous sommes maintenant rendus à ajouter le prochain niveau de la hiérarchie dans notre plan. Ainsi, nous glissons dans la procédure « BlocProc » deux objets *TextDisplay* (texte de début et de fin de bloc) et un objet *List* pour les 40 essais.



Encore une fois avant de poursuivre il faut spécifier les propriétés de la liste. Nous ajouterons donc 39 essais, et créerons une procédure pour passer au niveau hiérarchique des essais nommé « EssProc ». À ce point de la conception de l'expérience, il ne reste qu'à glisser dans cette dernière procédure les objet constituant un essai.

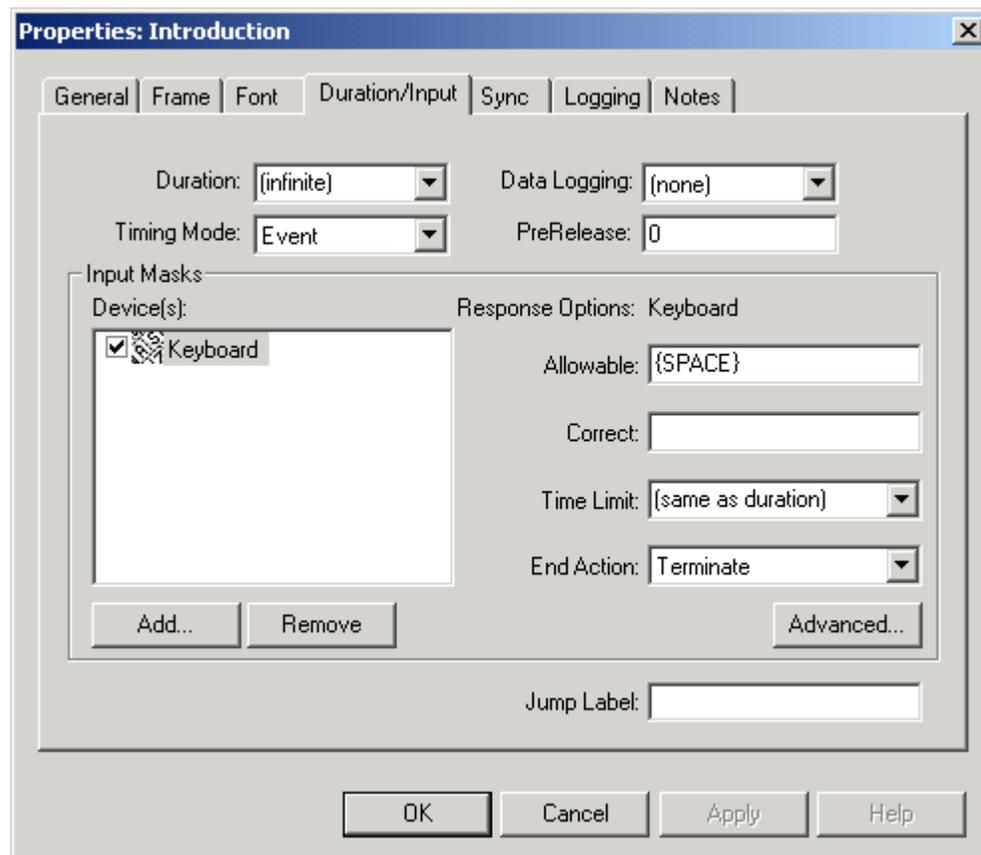


Nous avons maintenant créé la structure de notre expérience, il ne suffit maintenant que de spécifier les propriétés et les stimuli et nous avons terminé.

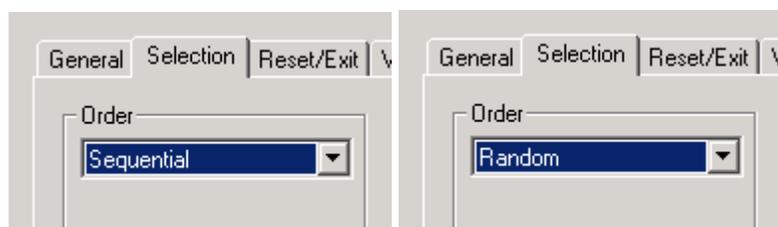


Section 2.3.2: Définir les objets

Pour tous les objets *TextDisplay*, le texte qui sera affiché est à votre discrétion. Pour ces objets il faut spécifier que nous attendons la barre d'espace. Pour cela il faut mettre (infinite) dans la propriété **Duration** et {SPACE} dans **Allowable** après avoir sélectionné le **Keyboard** dans l'*InputMask*.



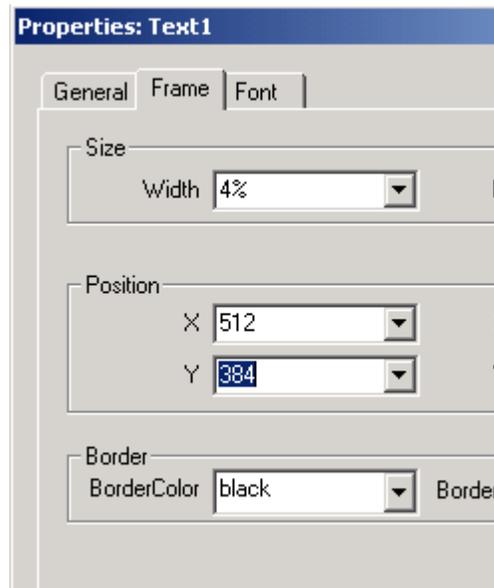
La seule propriété des objets *List* qui est à préciser à ce point-ci est l'ordre de sélection. Pour l'objet *List* « BlocX10 », cette propriété sera mise à Sequential et pour l'objet *List* EssaisX40, cette propriété est mise à Random.



Section 2.3.3 : Définir un essai

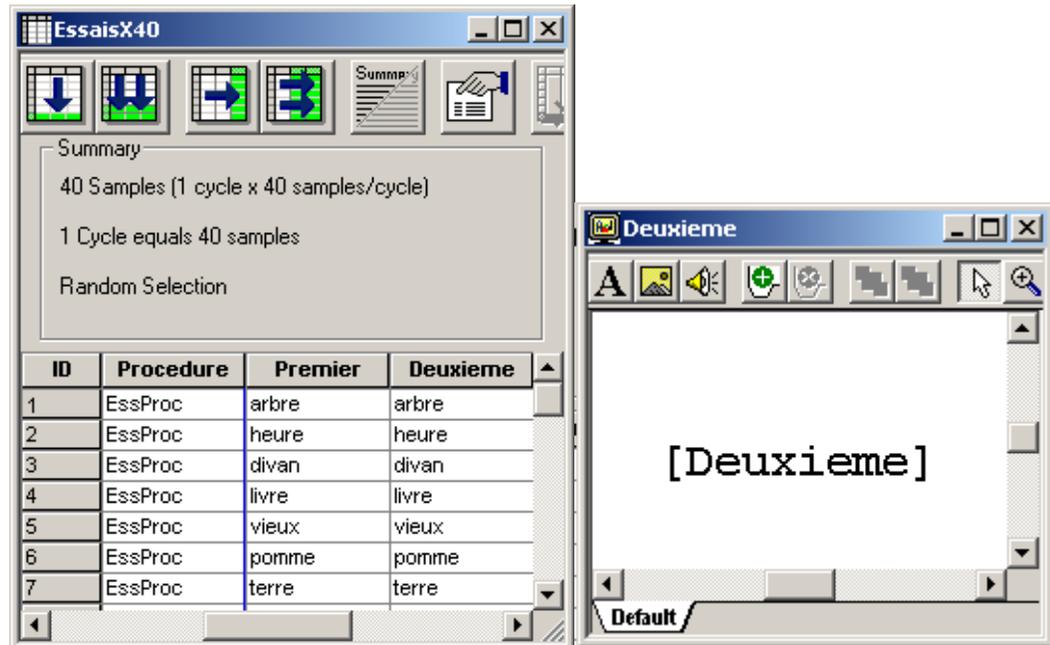
Un essai commence par un point de fixation. Ainsi il suffit de cliquer dans l'objet *Slide* « PtsFix » sur l'icône de texte  et de placer le point de fixation au centre de la

feuille. Il est aussi possible comme il a été dit plus haut de spécifier la position spatial d'un objet via ses propriétés dans l'onglet *Frame* sous la rubrique position.



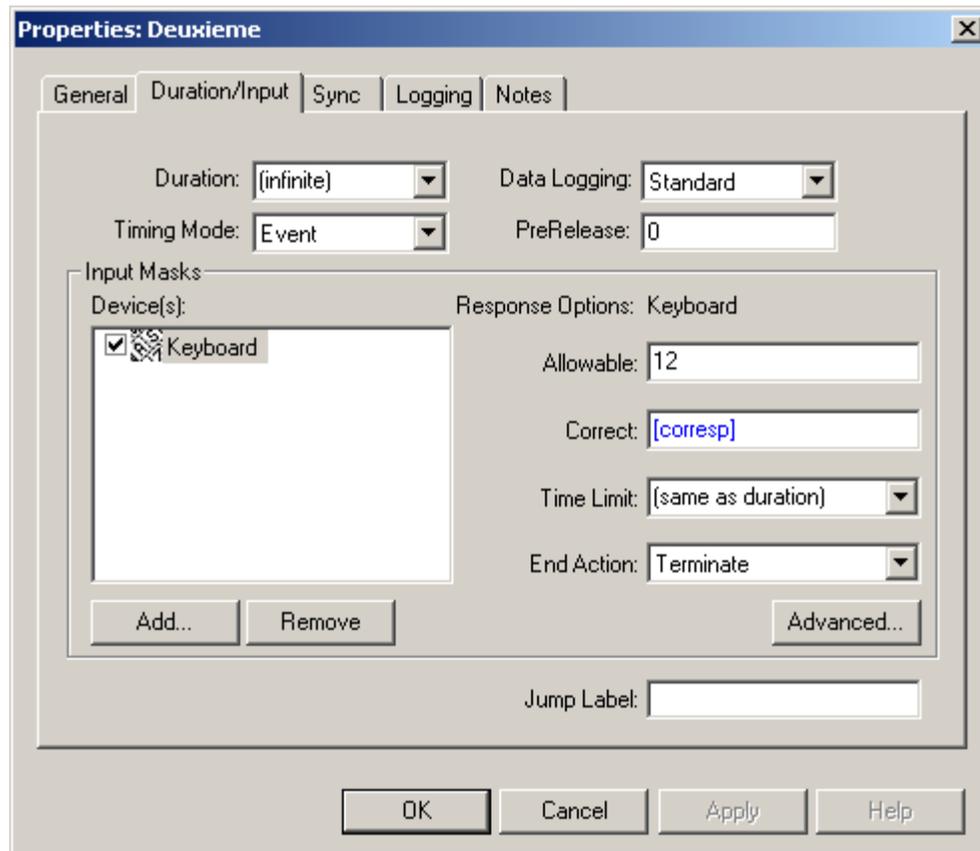
En ce qui concerne l'objet *Slide* « Premier », nous voulons présenter la première chaîne de caractère. Cependant celle-ci devra varier d'essais en essais, nous ne voulons pas toujours présenter la même chaîne de caractère. C'est à ce point que l'objet *List* joue un rôle critique. Lors d'un essai, E-prime sélectionne une ligne dans l'objet *List* précédant et utilise les attributs qui y sont spécifiés. Ainsi dans le cas présent. Pour l'objet *Slide* « premier », nous voulons que E-prime affiche un mot différent à chaque essai, ce mot étant précisé dans la liste précédente sous l'attribut « Premier ». Il ne suffit alors que d'inclure dans une zone de texte l'appel à cet attribut en mettant le nom de l'attribut entre deux crochets ([Premier]). La même chose doit être faite pour l'objet *Slide* « Deuxieme » où dans la zone de texte sera inscrit [Deuxieme]. Cette manière de référencer une zone de texte à un attribut est valable pour tous les champs dans E-prime où il est possible d'écrire du texte et où il y a une liste qui surplomb cet objet (e.g. *Filename*, *Allowable*,

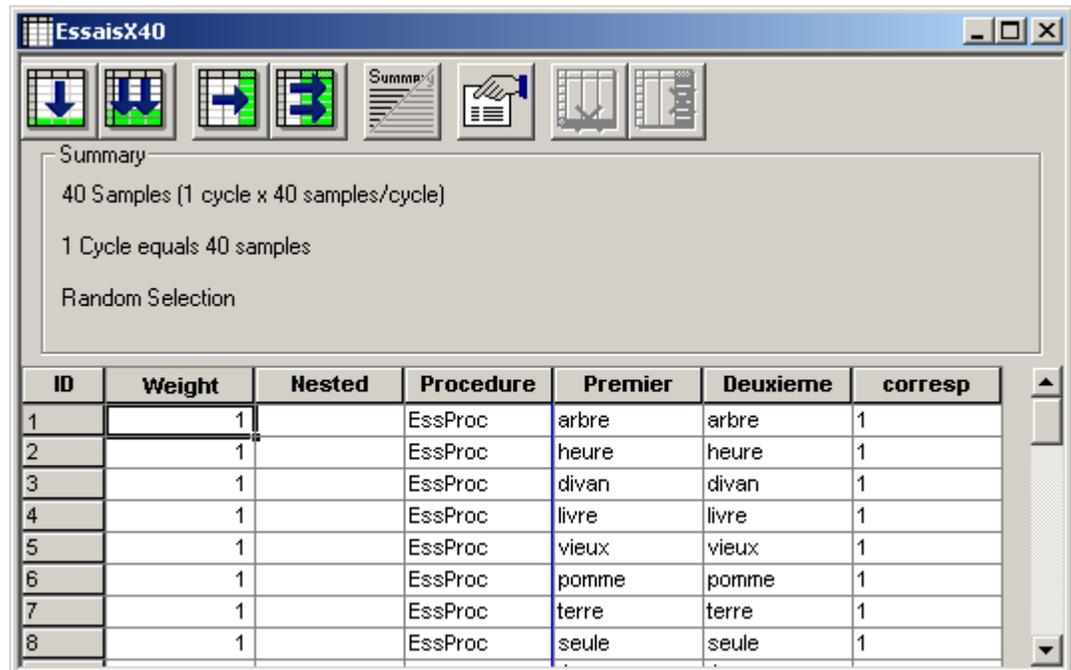
Correct, tout les attributs dans une liste, *duration*, *text*, etc.). Ainsi l'objet EssaisX40 doit ressembler à ce point à ceci :



Ensuite il ne reste qu'à spécifier la durée de chacun de ces objets. Pour les objets PtsFix, Premier et Masque, les durées qui seront inscrites dans *Duration* seront respectivement selon le plan 800, 50 et 34. Pour ce qui est de l'objet Deuxième, nous voulons collecter la réponse. Ainsi il faut spécifier ses propriétés de façon à ce qu'il enregistre le pourcentage de bonnes réponses ainsi que le temps de réponse. Pour faire cela il faut d'abord permettre au participant d'utiliser le clavier en sélectionnant « Keyboard » dans *Input Mask* et spécifier dans *Allowable* les touches qui seront permises, dans ce cas-ci : 1 et 2. En dernier lieu il faut spécifier quelle sera la bonne réponse. Étant donné que la bonne réponse varie selon le stimulus présenté, tout comme pour les objets Premier et Deuxième il faut référencer cette propriété à l'objet *list* qui le

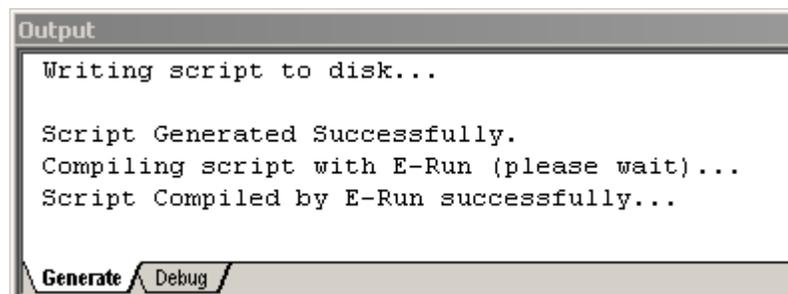
surplombe en mettant le nom de l'attribut entre [] dans le champ Correct et désigné un attribut dans la liste EssaisX40.





Le dernier objet à configurer est l'objet de rétroaction. Pour cela il suffit de spécifier dans la propriété ***Input Object Name*** l'objet auquel on veut donner une rétroaction (ici l'objet est Deuxième). Sans oublier de spécifier la durée de la rétroaction dans l'onglet *Duration/Input*.

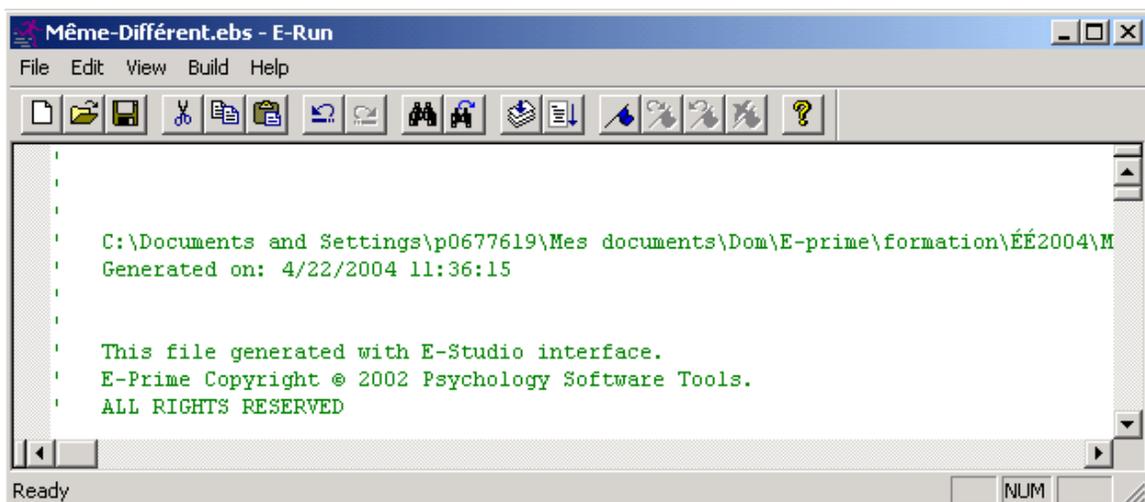
À ce point ci, l'expérience est complétée et devrait fonctionner. Pour vérifier si des erreurs sont présentes, cliquer sur le bouton de compilation  et la fenêtre Output vous indiquera les erreurs commise. Si la fenêtre output affiche ceci :



alors vous n'avez pas commis d'erreur et vous pouvez exécuter votre expérience et savourer le fruit de vos dur labeurs.

SECTION 3 : E-RUN

Une fois l'expérience conçue, il est temps d'expérimenter avec des participants. Parfois nous voulons expérimenter à plusieurs ordinateurs à la fois, mais la licence de E-prime coûtant cher, il serait bon d'avoir un sous-programme qui puisse exécuter le programme sans avoir besoin du logiciel entier. E-Run fait cela. Il se vent à part à un prix moindre et permet d'exécuter l'expérience. Pour démarrer E-run, il suffit de le sélectionner dans le menu **démarrer** et à ce moment là une page de code compilée apparaît.



The screenshot shows a window titled "Même-Différent.ebs - E-Run" with a menu bar (File, Edit, View, Build, Help) and a toolbar. The main area displays the following text:

```
C:\Documents and Settings\p0677619\Mes documents\Dom\E-prime\formation\ÉE2004\M
Generated on: 4/22/2004 11:36:15

This file generated with E-Studio interface.
E-Prime Copyright © 2002 Psychology Software Tools.
ALL RIGHTS RESERVED
```

The status bar at the bottom shows "Ready" and a "NUM" button.

Malgré qu'il est déconseillé de le faire, la barre de menu du haut offre des options habituelle d'un traitement de texte afin de pouvoir modifier le script. Mis à part ces bouton, les deux plus important bouton sont le bouton compilateur  et le bouton de démarrage de l'expérience . En appuyant sur le premier, le compilateur nous indique si des erreurs dans le programme ont eu lieu. Le second démarre l'expérience.

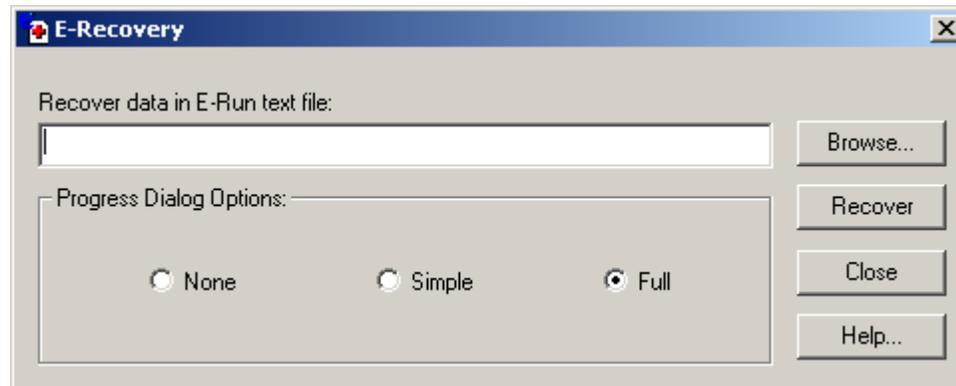
SECTION 4 : E-RECOVERY

Section 4.1 : Pourquoi l'utiliser

E-prime apporte une grande innovation par rapport à son ancêtre Mel Professionnal soit E-Recovery. Par le passé, les données étaient enregistrées à la fin de la session d'expérimentation. Ainsi il était possible qu'une panne de courant survienne durant une expérience et les données accumulées jusqu'à présent étaient perdues. E-prime enregistre les données à chaque essai dans un fichier texte et converti celui-ci en .edat à la fin de la session. Ainsi dans le cas d'un arrêt de l'expérimentation, E-Recovery peut prendre ce fichier texte et le convertir en .edat le rendant lisible par E-DataAid. Les données ne sont donc plus perdu.

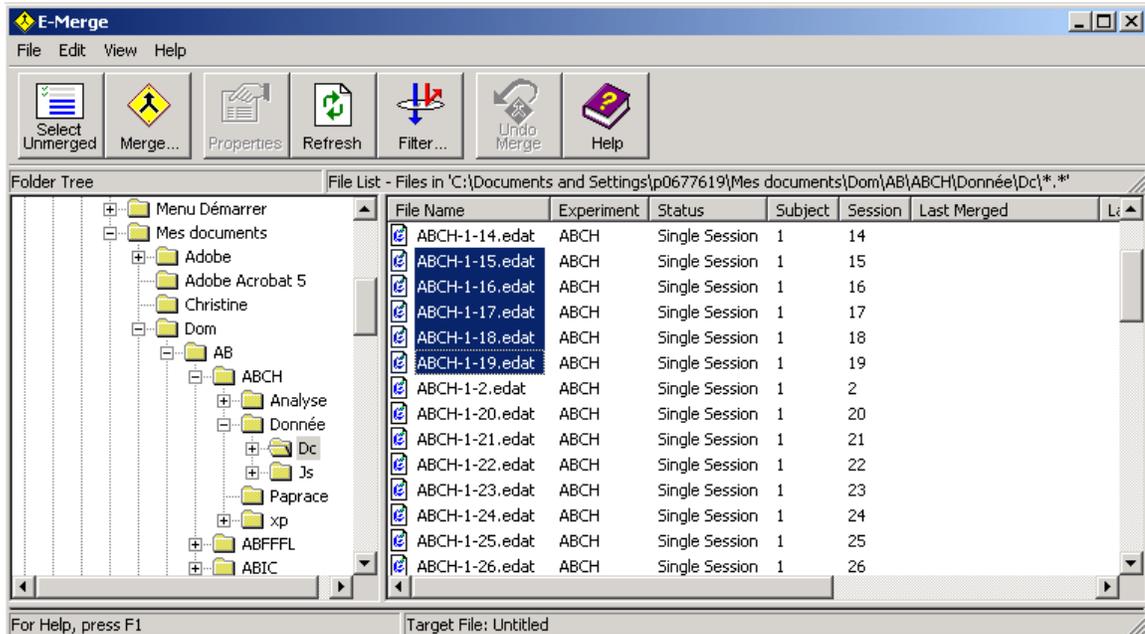
Section 4.2 : Comment l'utiliser

E-Recovery est simple à utiliser. Il suffit de le sélectionner dans le menu démarrer, chercher par le bouton  le fichier en question, cliquer sur « Recover » et nous obtenons dans le même fichier un fichier .edat.



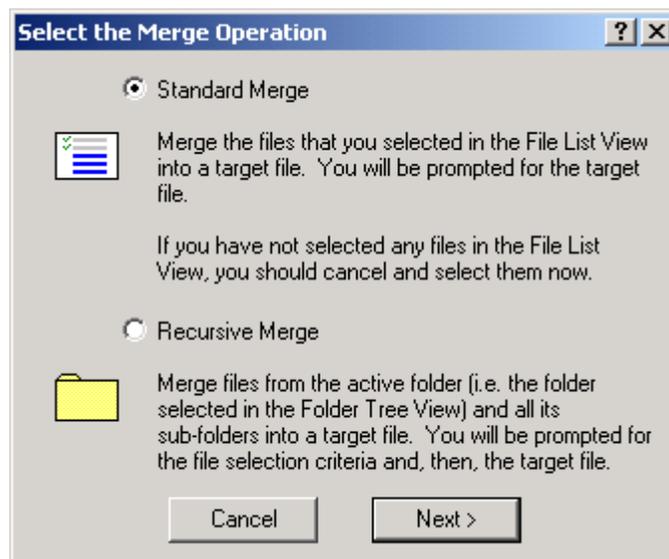
SECTION 5 : E-MERGE

Parfois l'expérience est faite avec plusieurs participants. Ainsi on obtient plusieurs fichiers de données, un pour chaque participant. Si nous n'avons que trois ou quatre participants, il est facile de copier-coller toutes les données dans un même fichier. Mais la tâche se complique lorsque nous avons des dizaines de participants. Ainsi, E-Merge est un sous-programme de E-prime qui fusionne les fichiers en un seul. Son interface est très simple.



Il suffit de sélectionner dans la hiérarchie le chemin jusqu'aux fichiers de

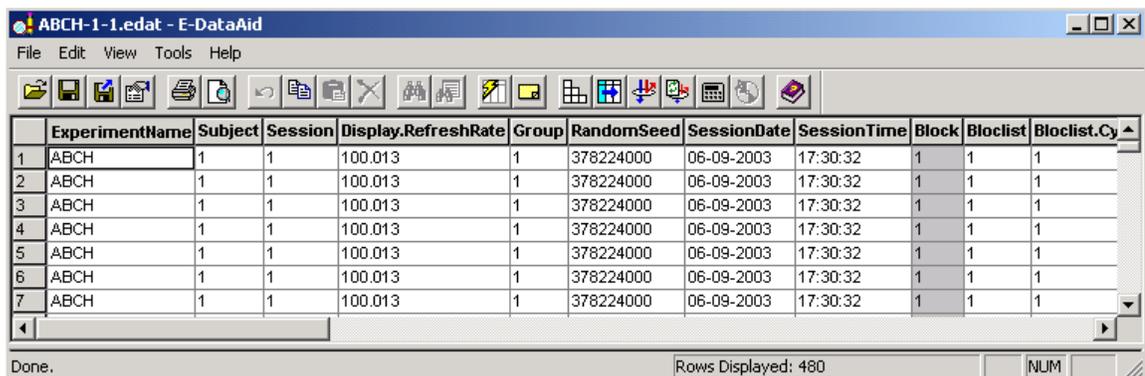
données, sélectionner les fichiers à fusionner et ensuite, cliquer sur le bouton . À l'incitation choisissez « Standard Merge », ceci créera un fichier avec l'extension .emrg contenant toute les fichiers sélectionnés.



SECTION 6 : E-DATAAID

Utiliser E-prime pour l'expérimentation donne des fichiers de type .edat. E-data-aid est un logiciel de traitement de données tel Excel, Spss, mais avec quelques fonctions en moins. Ainsi il est très utile pour faire un tour d'horizon de celles-ci et ensuite les exporter dans le logiciel de notre choix.

L'interface est très similaire à d'autre logiciel de traitement de données.



	ExperimentName	Subject	Session	Display.RefreshRate	Group	RandomSeed	SessionDate	SessionTime	Block	Bloclist	Bloclist.Cy
1	ABCH	1	1	100.013	1	378224000	06-09-2003	17:30:32	1	1	1
2	ABCH	1	1	100.013	1	378224000	06-09-2003	17:30:32	1	1	1
3	ABCH	1	1	100.013	1	378224000	06-09-2003	17:30:32	1	1	1
4	ABCH	1	1	100.013	1	378224000	06-09-2003	17:30:32	1	1	1
5	ABCH	1	1	100.013	1	378224000	06-09-2003	17:30:32	1	1	1
6	ABCH	1	1	100.013	1	378224000	06-09-2003	17:30:32	1	1	1
7	ABCH	1	1	100.013	1	378224000	06-09-2003	17:30:32	1	1	1

Done. Rows Displayed: 480 NUM

Chaque variable du programme est enregistré dans le fichier de donnée. En plus, d'autres variables, ayant parfois aucune utilité, sont enregistré. C'est une grosse lacune de E-prime, il enregistre une très grande quantité de variable et nous ne pouvons pas décider de ne pas les enregistrer. Ceci a pour conséquence de faire pour certaines expériences de très gros fichier de données. E-dataAid offre cependant plusieurs possibilité pour gérer les données. Il est possible de sélectionner des colonnes pertinentes et d'enlever les autres par le bouton . Il est aussi possible de filtrer vos données avec le bouton . Ainsi si vous voulez par exemple enlever tout les premiers blocs d'essais, la fonction filtre vous permet de sélectionner seulement les blocs à garder. E-DataAid offre aussi la possibilité

de faire quelques analyses primaires avec le bouton . Cependant, ceux-ci offre moins de possibilité qu'un logiciel d'analyses statistique. Ainsi vous pouvez exporter vos données vers presque tout autres logiciels statistique (SPSS, EXCEL, etc.) avec le bouton . Ceci créera un fichier compatible avec votre logiciel.

SECTION 7 : FACTOR TABLE WIZARD

Section 7.1: Utilité

Le Factor Table Wizard se retrouve avec les autres sous programme de E-prime, mais est plus un sous-programme Excel. Il arrive parfois que dans une expérience nous voulons créé toutes les combinaisons possibles de nos nombreux stimuli. Cependant, de créer cette liste dans un objet *List* peut être très long. Factor Table Wizard permet d'entrer simplement la liste de vos stimuli et il fait le travail à votre place vous retournant la liste que vous pouvez copier dans votre objet *List*.

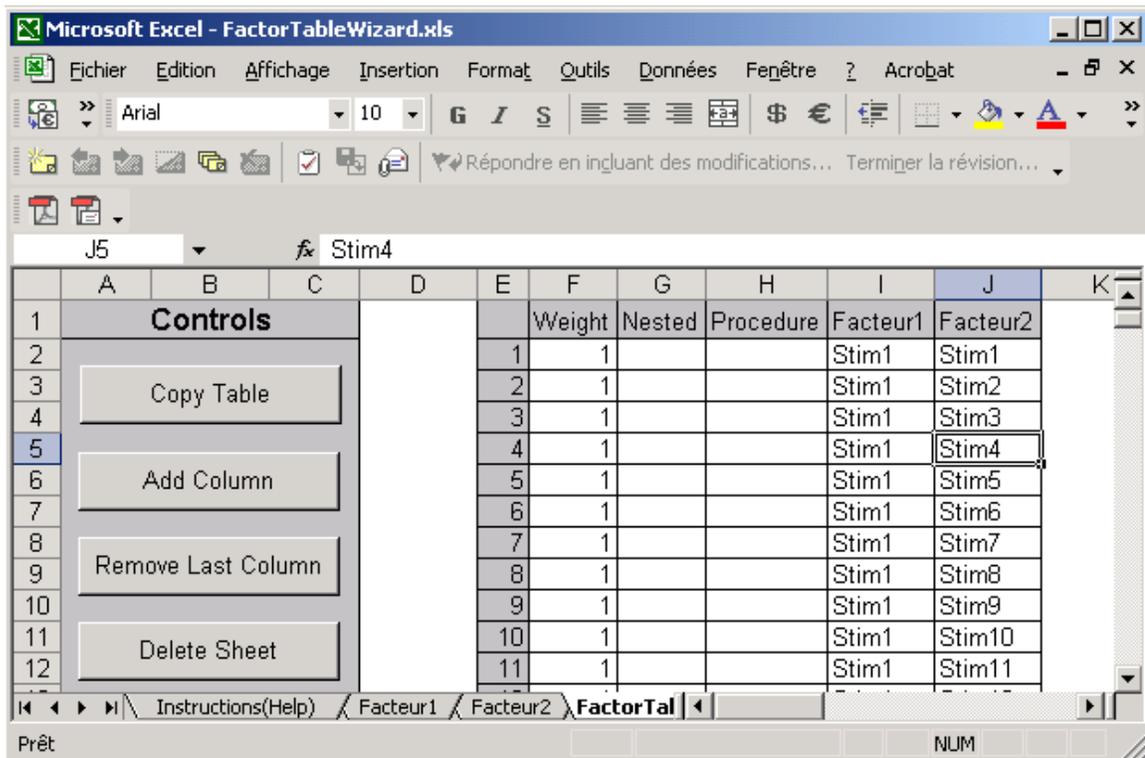
Section 7.2 : Comment ça fonctionne

À l'ouverture de ce sous-programme, après avoir cliquer sur le bouton



il suffit d'entrer la liste de nos stimuli, de répéter cette procédure

une deuxième fois et croisé les deux facteurs avec le bouton  pour obtenir la liste complète incluant les colonnes « Weight », « Nested » et « Procedure ».



Section 7.3 : Les dangers

Cet outils sauve parfois beaucoup de temps, mais il y a aussi des désavantages. Le principale est l'utilisation des macros. En effet, pour utiliser ce sous programme il faut désactiver (ou au moins diminuer) la sécurité des macros de notre ordinateur. Ainsi comme nous le savons, les virus sont souvent enclin à s'attaquer aux macros ce qui peut infecter votre ordinateur. Il est donc conseillé de l'utiliser que sur un ordinateur dont le risque d'avoir des virus est faible et de les réactiver aussitôt fini avec ce sous-programme.

Le deuxième désavantages, est la longueur des listes. En effet, de croiser 50 stimuli avec eux-mêmes donnera une liste de 2500 lignes. Ceci peut surcharger le programme fait avec E-studio et créé des complications. Sans oublier que ceci amène de longues sessions pour les participants.

SECTION 8 : AIDE

Rare sont les occasions où une expérience se constitue sans anicroches. Ainsi souvent nous avons besoin d'aide pour répondre à des questions. Ici sont énumérés quatre bonnes sources de référence mis à part ce document pour tenter de répondre à vos interrogations.

- E-Basic Help; Pour les problèmes de programmation dans les pages de code cette aide est très utile et donne de nombreux exemples de petits programmes pouvant souvent régler le problème;
- Le site Internet de PSTNET : <http://www.pstnet.com/products/e-prime/>;
Ce site Internet comporte dans sa section « Sample » de nombreux programmes complet pouvant exécuter des instructions très utiles tel enregistrer la position de la souris;

- Tutorial; Venant avec la licence E-prime, le tutorial est une bonne introduction à E-prime en nous permettant de faire un bon tour d'horizon du logiciel;
- Le site Internet <http://step.psy.cmu.edu/scripts/categories.html>; Ce site Internet donne de nombreuses expériences classique déjà programmées tel le Stroop, l'effet McCollough, etc. Ainsi il est bon de savoir que celui-ci existe et peut permettre un meilleur contrôle dans le cas où nous voudrions faire une variante de ces nombreuses expériences. De plus ces programmes utilisent souvent peu de programmation. Ainsi, il peut être bon de les regarder afin de voir d'autres exemple de programmes en E-prime.

SECTION 9 : PROGRAMMATION VB

Comme dans la plupart des logiciels de conception d'expériences, il est possible de programmer avec du code. E-prime reprend le code Visual Basic avec quelques petites modifications pour s'ajuster à l'interface e-prime. Il en résulte un code appelé E-basic. Pour insérer des lignes de codes, l'objet *Inline* est essentiel. Après l'avoir inséré dans la hiérarchie, en l'ouvrant, nous faisons face à une feuille blanche où on peut inclure le code E-basic.

 Section 9.1 : Sélection et boucles

Le tableau si-dessous fait état des diverse méthodes de sélection en langage VB. Ainsi il suffit de les implémenter dans votre programme en remplaçant les mots en italique par vos conditions.

<u>La sélection simple avec If</u>	<u>La sélection multiple avec Select Case</u>
If <i>condition 1</i> then <i>Instruction(s) A</i> Elseif <i>condition 2</i> <i>Instruction(s) B</i> End if	Select Case <i>Expression</i> Case <i>Cte1, Cte2, ..., CteN</i> <i>Instruction(s) A</i> Case <i>Borne1 TO Borne2</i> <i>Instruction(s) B</i> Case Else <i>Instruction(s) C</i> End Select

En code E-Basic il est aussi possible d'inclure des boucles. Le tableau suivant fait état des diverses boucles possible.

<u>Boucle de répétition WHILE</u>	<u>Boucle de répétition FOR</u>	<u>Boucle de répétition DO...LOOP</u>
While <i>Condition</i> <i>Instruction(s)</i> Wend	For <i>var = exp1 to exp2 step exp3</i> <i>Instruction(s)</i> Next <i>variable</i>	Do <i>Instruction(s)</i> Loop while <i>condition</i>

 Section 9.2: Les attributs

Comme il a été dit plus tôt, les attributs jouent une part importante dans la conception de l'expérience. Ce sont eux qui nous permettent dans les objets *List* de spécifier les conditions. Il a été dit que nous pouvons par exemple dans un objet *TextDisplay* faire référence à cet attribut en l'appelant avec les []. Il est aussi possible

d'appeler les attributs dans le code E-basic et de les modifier par la suite pour créé des condition parfois plus difficile. Pour appeler un attribut, nous utilisons le code :

```
c.getattrib("nom de l'attribut")
```

Tandis que pour créé un attribut directement dans la page de code e-basic :

```
c.setattrib "nouveau nom de l'attribut", valeur de l'attribut
```

Ainsi avec les sélections, les boucles et les attributs, nombreuses sont les possibilités de programmes.

SECTIONS 10 : POINTS IMPORANTS

En terminant, voici certains points importants à savoir concernant l'utilisation de e-prime.

- E-studio n'accepte que les fichier .BMP comme image et les fichiers .WAV comme son;
- E-prime n'est pas conçu pour la création des stimuli. Malgré qu'il est possible avec les objet Inline, par programmation de créer des stimuli simple, il vaut toujours mieux pour éviter les délais de traitement, créer ses stimuli avant l'expérimentation;

- Lorsque nous exécutons une expérience, il se peut que nous voulions sortir de celle-ci en cours d'expérience. Les touches **Ctrl+Alt+shift** permettent cela;
- Lors de la création d'une expérience, il vaut mieux la construire petit à petit, et ajouter des particularités sur un programme qui marche au lieu de faire un gros programme rempli d'erreurs. Par exemple, commencer par présenter votre stimulus et assurez-vous que cela fonctionne avant d'implémenter une rétroaction.
- Utiliser les listes et leurs fonctions. Si vous avez plusieurs listes dans une même expérience dont seulement des variables incluses dans ces listes diffèrent, alors c'est que vous avez trop de listes. Les seul fois où vous devriez avoir plusieurs listes c'est pour passer à des niveaux inférieurs de la hiérarchie, si vous avez des modes de présentation différents (ex : Random vs Séquentiel) ou lorsque les stimuli sont différents entre les blocs.
- Un programme trop long est un mauvais programme. Les meilleurs programmes sont ceux qui sont courts. Si votre programme est trop long vous multipliez les chances de faire des erreurs et risquer la chance d'oublier de modifier une propriété à un endroit quelconque. Tentez toujours de réduire au maximum la longueur de votre programme en utilisant plus efficacement les listes.

SECTION 11 : CONCLUSION

E-Prime est un logiciel au mille possibilités. Son interface facilite et accélère la programmation. Évidemment comme tout logiciel de programmation, il faut jouer avec pour savoir correctement l'utiliser. Dans ce cahier, nous venons de donner une introduction à ce logiciel qui vous permettra de commencer à l'appivoiser et pouvoir plus aisément l'explorer par vous-même.

Références

Note de l'auteur

Nous aimerions remercier Christine Lefebvre et H el ene Cassista pour leurs commentaires sur des versions ant erieurs de ce texte.

Pour r eimpression, s'adresser   Dominic Charbonneau A/S Denis Cousineau, D epartement de psychologie, Universit e de Montr al, C. P. 6128, succ. Centre-ville, Montr al (Qu ebec) H3C 3J7, CANADA, ou par courrier  lectronique   dominic.charbonneau@Umontreal.CA.