

Le Manuel de Codage

I. Fichier de démonstration de JSesh

Ce fichier (« allmdc.gly ») s'ouvre par défaut avec JSesh, il démontre les capacités de Jsesh concernant les règles du *Manuel de Codage*. Il comprend également des exemples de constructions usuelles. C'est aussi un tutoriel.

N'oubliez pas que Jsesh est un travail en cours.

II. Description du *Manuel de Codage*



Le *Manuel de Codage* comprend un ensemble de règles qui permet de représenter un texte hiéroglyphique à l'aide de caractères ASCII. Il est le résultat d'un travail de collaboration internationale qui est exposé dans la publication suivante : J. Buurman, N. Grimal, M. Hainsworth, J. Hallof, D. van der Plas, *Inventaire des signes hiéroglyphiques en vue de leur saisie informatique*, Paris, 1988 (1^{re} éd. 1984)



Le *Manuel* s'est inspiré du logiciel « Glyph » de J ; Buurman. La première version de « Glyph » datait du début des années 70.

La plupart des éditeurs hiéroglyphiques utilise des versions plus ou moins modifiées du *Manuel*. Des modifications substantielles ont été proposées en 1994 par H. van den Berg pour le logiciel « Winglyph ».

En 2000, lors de *La table ronde Informatique et Egyptologie* de Pise, M.-J. Nederhoff a proposé une intéressante révision *Revised Encoding System*, traitant certaines lacunes du *Manuel*. Nous prévoyons d'en tenir compte dans l'avenir.


A. Codes simples du *Manuel de Codage*

Le principe de base du Manuel de Codage est qu'un signe est décrit par son code dans la police de Gardiner. Par exemples,  est « A1 » et  est « G5 ».

Les variantes de signes (ceux qui apparaissent avec un astérisque dans la liste de Gardiner) sont indiquées par une lettre capitale après le code du signe de base. Par exemples,  est « A14A », alors que  est « A14 ».

Note: quand JSesh ne connaît pas un signe, il le remplace automatiquement par son code.






Par exemple: A600-n:A600*A1 gives A600 A600 .

Un certain nombre de signes peut également être décrit par leur(s) valeur(s) phonétique(s). Par exemple,  peut être codé "b". Une liste complète des valeurs phonétiques disponibles sera mise à disposition dans l' « Aide » de JSesh.



Les codes alphabétiques servant à exprimer les unilitères de l'égyptien sont :

A	ꜥ	i	<i>i</i>	y	<i>y</i>	a	ꜥ
w	<i>w</i>	b	<i>b</i>	p	<i>p</i>	f	<i>f</i>
m	<i>m</i>	n	<i>n</i>	r	<i>r</i>	l	<i>l</i>
h	<i>h</i>	H	<i>ḥ</i>	x	<i>ḫ</i>	X	<i>ḫ</i>
z	<i>s</i>	s	<i>ś</i>	S	<i>š</i>	q	<i>ḳ</i>
k	<i>k</i>	g	<i>g</i>	t	<i>t</i>	T	<i>t</i>
d	<i>d</i>	D	<i>ḏ</i>				

Concernant certains signes unilitères, il existe deux codes spécifiques pour les variantes fréquentes (du Nouvel Empire notamment) :

w		W	ꜥ
m		M	
n		N	

A ces codes, le *Manuel* ajoute également deux groupes :


nTrw	
nn	

B. Grouper des signes





Grouper les signes peut se faire à l'aide de quelques opérateurs simples :

-	Sépare les cadrats	G41-A	
:	Empile les cadrats	p :n	
*	Aligne les cadrats sur un même niveau	p*t:pt	

D groupes plus complexes peuvent être formés en utilisant des parenthèses :





pA-A-N16:N23*Z1-n:(x:t)*U30-xAst:t*Z1



C. Codes spécifiques

..	Cadrat grand espace	-A-.-A-	
.	Cadrat espace court	-A-.-A-	
O	Point de ponctuation noir	-O:.-	
o	Point de ponctuation rouge	-o:.-	





D. Hachures

Quand une zone n'est pas bien lisible, vous pouvez utiliser les codes suivants :

//	Cadrat non lisible	
/	Quart de cadrat (*)	
v/	Partie verticale d'un cadrat	
h/	Partie horizontale d'un cadrat	

On peut combiner des signes « visibles » avec des zones de hachures. Par exemple, p:h/ donne .


(*) Pour hachurer une partie de cadrat, vous pouvez utiliser le code « # » comme suit :

#1	Hachure quart de cadrat supérieur gauche	p*t:pt#1	
#2	Hachure quart de cadrat supérieur droit	p*t:pt#2	
#3	Hachure quart de cadrat inférieur gauche	p*t:pt#3	
#4	Hachure quart de cadrat inférieur droit	p*t:pt#4	

Ces codes peuvent se combiner. Par exemples :





p*t:pt#13	
p*t:pt#14	
p*t:pt#1234	

Pour hachurer une partie de texte plus large, vous pouvez utiliser les codes « #b » (begin, début de hachure) et « #e » (end, fin de hachure) relatifs aux zones de hachures :

i-W-bA:k-A1-#b-Xr:r-n:h:t-#e-xt:Z1


E. Surimpression de signes

Il est possible de combiner des signes en utilisant l'opérateur « ## ». Le résultat n'est pas toujours très bon mais la possibilité existe. Par exemple :

-m##a-	
-a##b-	
-M3##M1-	
-M1##M3-	

Note : l'opérateur officiel est « # » mais je suggère de le remplacer plutôt par l'opérateur « ## ». En effet, l'opérateur « # » est utilisé de manière abusive et finit par signifier trop de choses. En tous les cas, JSesh transformera automatiquement le code « # » en « ## », durant la prochaine sauvegarde du texte effectuée.

F. Les zones rouges

Une zone de texte peut être écrite en rouge grâce aux codes « \$r » (début de zone rouge) et « \$b » (début de zone noire). Par exemple :

- \$r-HAt:a-m-s-b-sbA-A-i-i-t:Y1-arq:Z1-m-anx-n:x-Y1v-\$b-mt:t-r:W-DbA-DbA-Hw-A2-Z3-n-W-DA-A-Y1v-

Des signes peuvent également être écrits en rouge individuellement. C’est une spécificité de JSesh. Par exemple : -p*p:p\red*p- donne .

G. Inversion et rotation de signes

Un signe peut être inversé (verticalement) en utilisant le code « \ » après lui. Par exemple, -A1\ - donne

Dans JSesh, vous pouvez incliner un signe suivant n’importe quel angle en utilisant l’opérateur « \R » suivi de l’expression de l’angle en degrés. Par exemple :

-anx\R30-G5-G7-	
-A1-A\R90-A\R180-A\R270-A\R360-	

Ces deux fonctions peuvent également être combinées. Par exemple :

-A1-A\\R90-A\\R180-A\\R270-A\\R360-	
-------------------------------------	--


Note : L’opérateur « \ » est non standard. Il a été proposé dans le logiciel Winglyph en 1994, mais apparemment sans être utilisé.


Le *Manuel*, à l’origine, ne propose rien du tout concernant les rotations et la plupart de ses mises en application utilisent uniquement les rotations de 90° en utilisant les deux opérateurs « \r » et « \t ».

\r + 1, 2, 3, 4	Rotation	-A1\r1- A1\r2- A1\r3- A1\r4-	
\t + 1, 2, 3, 4	Rotation + inversion	-A1\t1- A1\t2- A1\t3- A1\t4-	

H. Redimensionnements




Il est possible de redimensionner un signe en utilisant l'opérateur « \ » suivi de l'expression d'un pourcentage. La taille du signe de base sera modifiée en conséquence.

Par exemple, `-A1-A1\80-A1\50-A1\20-` donne 


Le pourcentage peut même est supérieur à 100. Par exemple, `-t-t\80-t\200-t\400-t\1000-` donne 

Note : La taille du signe est limitée par la taille du cadrat.



Cet opérateur n'existait pas dans le Manuel de 1988 mais il est largement utilisé.

Il peut être utilisé pour personnaliser l'effet de redimensionnement (automatique) d'un cadrat. Si on considère le groupe , il s'écrit « `zA:zA*zA` ». On peut remarquer que le signe « `zA` »  du dessus est un plus large que les deux autres. On peut corriger cela en utilisant un redimensionnement : `-zA\70 :zA\70*zA\70-`, ce qui donne .

Suivant la typographie standard, la taille des glyphes est régie par un facteur décroissant égal à $\sqrt{2}$. Cela produit un effet visuel équilibré. Vous pouvez essayer en utilisant les


facteurs suivants : `-A1-A1\70-A\49-A\35-A1\24-` donne .
 ($\sqrt{2} \sim 1,41$; $24 \times \sqrt{2} \sim 34$; $35 \times \sqrt{2} \sim 49,5$ et $70 \times \sqrt{2} \sim 99$.)

On peut tester des combinaisons d'effets « rotation + inversion + redimensionnement » (est-ce bien utile ?) :

<code>xA\50-xA\50\R45-xA\ R45-xA\ 50\R45-!</code>	
<code>-wD-wD\R45-wD-wD\R90-pt\R90-</code>	

I. Signes à ignorer



Il s'agit d'une spécificité de JSesh. Dans le cas où les cadrats ne respectent pas les limites des mots, vous devez parfois citer un mot, ou un texte, qui commence au milieu d'un cadrat. Dans JSesh, l'opérateur « \i » indique que le signe est sans rapport avec la citation. Il sera représenté en couleur gris clair, simplement pour remplir l'espace.

Par exemple, -M:p*t:pt- donne .


J. Groupes comprenant un signe large

Dans les textes égyptiens tardifs et en particulier, dans les textes hiératiques, certains groupes peuvent être très larges. Souvent, ils contiennent un hiéroglyphe qui est très étendu et d'autres signes plus petits en dessous. Le redimensionnement automatique réduit tout le groupe à une largeur de cadrat et rend la représentation du texte à la fois infidèle et difficile à lire.

JSesh met alors à disposition l'opérateur « /l ». Il indique que le signe courant peut avoir la largeur qu'il veut. Par exemples :

Sans l'opérateur « /l »	-m-xt:x\80*t\80*D54\80-	
Avec l'opérateur « /l »	-m-xt\l:x\80*t\80*D54\80-	

Le meilleur résultat est obtenu en combinant cet opérateur « \l » et le redimensionnement.

Par exemple, -ir\200:r*t*W-ir\200\l:r*t*W- donne .

K. Codes grammaticaux

La Manuel établit que le simple espace « » placé un signe indique que ce signe termine un mot et que le double espace « » derrière un signe indique que ce signe termine une phrase.



JSesh enregistre cette information en même temps que les signes, mais actuellement, JSesh n'en fait aucun usage.

Comme alternative, d'après une pratique du logiciel Winglyph, vous pouvez utiliser un trait souligné (underscore) « _ » à la place de l'espace. Par exemple, -w=f_-m_-p*t:pt_ donne







L. Positionnement absolu

Bien que son utilisation ait été repensée à partir de zéro, JSesh emprunte la syntaxe qui suit au logiciel MacScribe. Un groupe constitué à l'aide de l'opérateur « ** » utilisera un positionnement absolu. Chaque signe du groupe doit être suivi par une position de la forme « x, y, s » où :

- « x » est la distance qui sépare le signe du côté gauche du groupe exprimée en $\frac{1}{1000}$ de la hauteur du signe  (A1) ;
- « y » est la distance qui sépare le signe du côté supérieur du groupe exprimée en $\frac{1}{1000}$ de la hauteur du signe  (A1) ;
- « s » est le dimensionnement du signe exprimé en pourcentage.

Si rien n'est spécifié, les valeurs par défaut sont « 0,0,100 ».

Exemples :

-stp{{0,0,100}}**n{{0,800,100}}**ra{{700,0,70}}-	
-A1{{200,20,100}}**T31\R50{{120,0,40}}-	
-A1-stp**n{{0,800,100}}**ra{{600,0,70}}-A1-	 

Note importante : Nous utilisions auparavant l'opérateur « && », ce qui était une faute car pour le logiciel MacScribe, son action est quelque peu différente. Dorénavant, JSesh utilise l'opérateur « ** ».

M. Ligatures

Un système de ligature a été ajouté depuis peu pour former des groupes spécifiques. C'est une extension « standard » du *Manuel de Codage*.







JSesh connaît un certain nombre de groupes spécifiques. L'opérateur « & » indique que les signes doivent être regroupés de manière spécifique.


Par exemple : -G1&X1-stp&n&ra-D&d-Ax&x-t&w&t- donne      .

N. Ligatures complexes


Depuis la version « 2 alpha 20 », JSesh comprend un mécanisme, inspiré du logiciel MacScribe, servant à lier un signe et un groupe complexe. Pour l'expliquer rapidement, chaque signe comporte deux zones rectangulaires dans lesquelles un groupe peut s'accrocher, une zone devant et une zone derrière le signe.

Par exemples, pour la zone située devant le signe, on utilise l'opérateur « ^^^ » :

Insertion de  devant 	<code>-(Z1:Z1*Z1)^^^nTr-</code>	
Insertion de  devant 	<code>-X2^^^nTr-</code>	

Cela fonctionne bien avec la plupart des oiseaux. Le code `-t^^^w-t^^^m-t^^^A-` donne .

D'autres exemples, pour la zone située après le signe, où on utilise l'opérateur « &&& » :

<code>-Ax&&&x-.-A&&&t-.-w&&&t-.-ns&&&(mSa*Z3)-.-D&&&(d:d:t)-</code>


Ces deux opérateurs peuvent être combinés. Par exemple, `-t^^^w&&&t-` donne .

Quand JSesh ne connaît pas explicitement les zones d'accrochage d'un signe, il essaie de le deviner.


Note : Dans les prochaines versions de JSesh, les opérateurs « ^^^ » et « &&& » seront remplacés respectivement par « ^^ » et « && ».

O. Modifications par un utilisateur d'un signe

Pour permettre des extensions du Manuel, JSesh peut interpréter toutes modifications placées derrière le signe, si elles respectent les règles suivantes :

- une modification est introduite par le code « \ » ;
- elle contient une séquence de lettres qui désignent son nom et une séquence de chiffres qui désignent sa valeur.

Par exemple :

<code>-A1\\IdontMeanAthing\number1000-</code>	
---	---

P. Cartouches, Serekh, etc..

Les cartouches sont représentés entre les codes « -<- » et « ->- ».

Exemples de cartouches :

-sw:t-bit:t-<-ra-mn:n-xpr->-anx-DA-s-	
-<-sw-t:n-G7-N41:t*B1-G7-t:A-A51-s-y:B1-G7->-	

Les Serekh sont représentés entre les codes « -<S- » et « ->- ». Par exemple :

-<S-E1:D40-mAat-mr->-	
-----------------------	--

Les signes-Hwt sont représentés entre les codes « -<H- » et « ->- ». Par exemple :

-<H-Z3-A19->-	
---------------	--

Les « enceintes » sont représentées entre les codes « -<F- » et « ->- ».

-<F-bA-A-y:r-i-i-Aa18-Z1-mr:r-W:t-A2-r-D:t:N17->-	
---	--

Certaines parties du cartouche peuvent être absentes, ou peuvent être tracées à un endroit différent. Dans ce cas, la lettre-code (qui peut être absente) « s » ou « h » doit être écrite en minuscule. Un chiffre indique ce qui doit être tracé.



Pour les cartouches, les différentes constructions possibles sont les suivantes :

0	Aucun « nœud »
1	Pas de « nœud »
2	Un « nœud »

Exemples de cartouches :

-<1-ra-mn:n-xpr-1>-	
-<1-ra-mn:n-xpr-0>-	
-<2-ra-mn:n-xpr-1>-	




Exemples de serekh :

-<S-E1:D40-mAat-mr>-	
-<s2-E1:D40-mAat-mr-s1>-	

Pour les signes-Hwt, les différentes constructions possibles sont les suivantes :

0	Rien
1	Aucun « carré »
2	Un « carré » en bas
3	Un « carré » en haut

Exemples de signes-Hwt :

-<h0-Z3-A19-h3>-	
-<h3-nTr-h1>-	
-<h2-nTr-h2>-	

Q. Textes alphabétiques

Un texte alphabétique est introduit pas le code « + » suivi d'une lettre :

+	Texte de commentaire (invisible)
+l	Texte latin
+i	Texte <i>italique</i>
+b	Texte gras
+t	Texte en translitération
+c (*)	Texte copte
+h (*)	Texte hébreu
+g (*)	Texte grec
+r (*)	Texte cyrillique
+s	(Retour au) texte hiéroglyphique

(*) Ces fonctionnalités ne sont pas encore supportées par JSesh.

S. Retour à la ligne



Un retour à la ligne est indiqué par le code « -! ». Si un retour à la ligne est de la forme « -! » + pourcentage + « % », cela indique qu'un espace d'un certain pourcentage par rapport à la normale sera produit. Par exemple, pour obtenir un espace quatre fois plus grand que la normale, il suffit de terminer la ligne par -!=400%.

T. Saut de page

Un saut de page est indiqué par le code « !! ».



U. Lignes horizontales

Une ligne peut être tracée en utilisant le code « {l} » + « *number1,number2* » où


- *number1* donne la position de départ en $\frac{1}{200}$ de la largeur du signe  ;
- *number2* donne la position de fin en $\frac{1}{200}$ de la largeur du signe .

La lettre utilisée peut être « l » ou « L » pour tracer respectivement une ligne fine ou épaisse.


Par exemples :

{l200,800}-!	
{L200,800}-!	




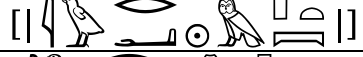

V. Tabulation

Une tabulation est représentée par le code « ? » + *number* où *number* indique la grandeur de l'espace de tabulation en $\frac{1}{200}$ de la largeur du signe .


Par exemples :

?800 +l an exemple +s !	an exemple
?400-A1-?100-A2-!	

W. Codes philologiques

Ajouté	[& et &]	-[&i-w-r:a-ra-m-p*t:pt-&-	<  >
Superflu	[{ et }]	-[{-i-w-r:a-ra-m-p*t:pt-}-	{  }
Effacé	[[et]]	-[[-i-w-r:a-ra-m-p*t:pt-]]-	[]
Auparavant lisible	[" et "]	-["-i-w-r:a-ra-m-p*t:pt-"]-	[]
Ajouté par le scribe	[' et ']	-['-i-w-r:a-ra-m-p*t:pt-']-	'  '

Le *Manuel* original suggère que les indications philologiques soient notées entre parenthèses mais cela ne correspond pas aux usages philologiques en vigueur. A partir de là, le logiciel Winglyph considère que ce sont de simples signes, ce qui entraîne que vous devez les insérer « manuellement ». JSesh supporte les deux interprétations et il lit tout aussi bien les fichiers à extension « .gly » et à extension « .hie ».

A l'intérieur d'un cadrat, le code - p*[[*t*]]:pt_-p*[[*t:pt-]]- donne .

X. Usages osbolètes du *Manuel de Codage*

JSesh est capable de lire des textes qui ne suivent pas exactement le Manuel de Codage. Par exemple, JSesh est très indulgent dans le cas de l'utilisation d'espaces. Cependant, lorsqu'un texte est sauvegardé, JSesh retiendra le code le plus normalisé possible pour assurer une lisibilité maximale.

Annexe : Les codes des unilitères de l'égyptien en majuscules

Bien que les Anciens Égyptiens ne distinguaient pas majuscules et minuscules, nous aimons pouvoir faire la distinction au niveau de la translittération afin d'en faciliter la compréhension.

Voici donc les codes servant à exprimer les unilitères de l'égyptien en majuscules (police « EgyptoItalic ») :

A	Ⲁ	I	Ⲓ	Y	Ⲙ	a	ⲁ
u	W	B	B	P	P	F	F
M	M	N	N	R	R	L	L
H	H	J	Ḥ	c	Ḥ	C	Ḥ
Z	S	-	-	U	Š	Q	Ḳ
K	K	G	G	v	T	V	T
e	D	E	Ḏ				

Mémento « EgyptoItalic » pour les minuscules :

a	i	y	ⲁ	w	b	p	f	m	n	r	h	H	x	X	z	S	q	k	g	t	T	d	D
Ⲁ	Ⲓ	Ⲙ	ⲁ	ⲱ	ⲃ	Ⲕ	ⲕ	ⲙ	ⲛ	ⲣ	ḥ	Ḥ	ḫ	Ḫ	ḏ	š	ḳ	ḳ	ḡ	ṯ	ṯ	ḏ	Ḏ

Mémento « EgyptoItalic » pour les majuscules :

A	I	Y	a	u	B	P	F	M	N	R	J	J	c	C	Z	U	Q	K	G	v	V	e	E
Ⲁ	Ⲓ	Ⲙ	ⲁ	W	B	P	F	M	N	R	H	Ḥ	Ḥ	Ḥ	S	Š	Ḳ	K	G	T	T	D	Ḏ

Table des matières

Le Manuel de Codage	1
I. Fichier de démonstration de JSesh	1
II. Description du <i>Manuel de Codage</i>	1
A. Codes simples du <i>Manuel de Codage</i>	2
B. Grouper des signes	3
C. Codes spécifiques.....	3
D. Hachures.....	3
E. Surimpression de signes.....	4
F. Les zones rouges	5
G. Inversion et rotation de signes	5
H. Redimensionnements.....	6
I. Signes à ignorer	7
J. Groupes comprenant un signe large	7
K. Codes grammaticaux	7
L. Positionnement absolu.....	8
M. Ligatures	8
N. Ligatures complexes	9
O. Modifications par un utilisateur d'un signe	9
P. Cartouches, Serekh, etc.	10
Q. Textes alphabétiques	11
R. Numérotation des lignes	12
S. Retour à la ligne.....	13
T. Saut de page.....	13
U. Lignes horizontales.....	13
V. Tabulation.....	13
W. Codes philologiques.....	14
X. Usages osbolètes du <i>Manuel de Codage</i>	14
Annexe : Les codes des unilitères de l'égyptien en majuscules	15