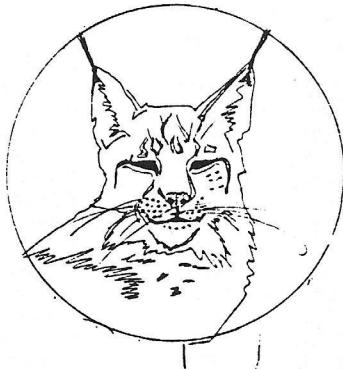


L'ŒIL DE LYNX



ROUTINES EN ROM ADRESSE MÉMOIRE

REDEMARRAGES

RST	8H	Routine d'inscription écran.
RST	10H	Examine si le caractère à l'adresse pointée par DE est " ". Sinon DE jusqu'à ce qu'il n'y ait pas d'espace.
RST	18H	Evalue l'expression en polonaise inversée montrés par DE comme un nombre entier binaire dans HL.
RST	20H	Regarde si l'octet après RST est le même que A sinon inscrit une erreur de syntaxe.
RST	28H	Evalue si l'expression en polonaise inversée montré en DE en virgule flottante dans WRA1 (Working register area 1).
RST	30H	Saut vers la routine moniteur TRAP.
RST	38H	Saut vers RAN RSTRAM (voir variables RAM).

ADRESSES

33H	INBLUE	Comme POP HL JP (HL).
69H	INGREEN	Lit l'octet dans le bank red/blue en HL dans L avec H = 0.
70H	FONT	Lit l'octet dans le bank vert/vert alternatif en HL dans L et H = 0.
C EH	KEYDVR	Donne les adresses de la carte mémoire caractères en A.
9BKH		Gestionnaire de clavier donne le code de la clé pressé en A sans simple clé d'entrée. (ESC + A-Z).
0B65H	RSYNC	Lit la Sync. de la cassette.
0B85H	RBYTE	Lit l'octet de la cassette en A.
0B93H	WSYNC	Ecrit la sync sur la cassette.
0CF2H	MOTON	Active le moteur de la cassette.
0CFBH	MOTOFF	Désactive le moteur du cassette. NB : doit suivre MOTON.
105EH	PRTSTR	Inscrive les chaînes montrées en DE et termine par CR.
1B98H	RDN	Lit les nombres montrés en DE dans WRA1 en virgule flottante.
IC9AH	PHL	Inscrive les nombres binaires en HL sans Zéro.
ID59H	PN	Inscrive les nombres en WRA1.
IED0H	FRNDS	Genère le nouveau nombre aléatoire en RNDNO.
IF0RH	ESC	Définit l'indicateur z si ESC à été pressé.
202FH	DEY	Appel du gestionnaire clavier KEYB.
2132H	ETEXT	Même chose que la commande TEXT en BASIC.
2234H	PHEXHL	Ecrit les nombres binaires en HL.
2568H	CHLDE	Compare les indicateurs HL, DE défini comme HL - DE.
25E2H	RP	Lit les ASCII montré en DE. Interprète en polonaise inversée ce qui est montré par HL.
2B1FH	FNDLN	Trouve la ligne avec le numéro en WRA1. IX montre la ligne et l'indicateur z indique qu'il a trouvé. Autrement IX montre la ligne suivante la plus haute.
3497H	FPINT	Virgule flottante en WRA1 vers les entiers en HL.
34C4H	INTFP	Entier en HL vers les virgules flottantes en WRA1.
350DH	ZWRA1	Efface WRA1 avec des 0.
3539H	PLINE	Inscrive la ligne montré par HL et terminé par 0.
3542H	SWAP	Interchange le contenu de WRA1 et WRA2.
354EH	LZERO	Charge WRA1 avec une virgule flottante à 0.
3561H	CMP	Compare WRA2, WRA1. Indicateur défini, comme ceci : (WRA2) - (WRA1).
35B1H	LNUM1	Charge les nombres montrés par HL dans WRA1.
35BAH	LNUM2	Charge les nombres montrés par HL dans WRA2.
35BFH	LONE	Charge WRA1 avec une virgule flottante 1.
366AH	SBT	Soustrait (WRA2) de (WRA1), stocke la réponse en WRA1.
366DH	AD	Additionne (WRA1) et (WRA2), stocke la réponse en WRA1.
36C8H	MLT	Multiplie (WRA1) et (WRA2), stocke la réponse en WRA1.
37B0H	DIV	WRA1 = WRA1/WRA2.
STACK	61EE	Adresse de stockage du pointeur de pile du Basic.
RNDNO	61F0	Amorce de source d'un nombre aléatoire.
HLSSTORE	61F4	Utilisé pour sauvegarder HL après un CALL.
CURRLP	61F6	Pointeur du premier octet de la ligne Basic qui est exécuté.

EDITORIAL

Chers Lynxistes,

Voici le premier numéro de notre revue trimestrielle, nous lui attribuons le numéro 0, car nous considérons qu'elle doit changer de visage dès le numéro 1.

Voici comment :

Envoyez-nous toutes les remarques et suggestions, ainsi que tous les programmes et trucs que vous désirez partager avec les autres utilisateurs de LYNX.

Notre comité de rédaction constitué de membres de la Société GOLEM, d'utilisateurs et de journalistes professionnels essayeront de répondre à toutes vos questions et se feront un plaisir de placer dans les colonnes de L'ŒIL de LYNX vos programmes.

Cette revue est à vous, c'est vous qui façonnerez son visage. Devenez collaborateur, rédacteur de la revue.

Nous attendons vos initiatives.

C. Chamaillard.

DATAP	61F8	Pointeur utilisé par READ DATA qui se positionne à la fin de la dernière entrée lue.
OTYPE	6206	Type de sortie et statut de LINK.
LASTDSP	6207	Dernier caractère inscrit sur l'écran (utilisé par VDU1,n VDU2,n).
VTYPE	6208	Type de la variable courante.
CONTLP	6209	Pointeur pour CONTINUER la ligne. Ø s'il ne peut continué.
RSTACK	620B	Pointeur de retour de pile.
RSP	620D	Return stack pointeur.
VTBL	620F	Pointeur de variable.
ATBL	6211	Pointeurs des vecteurs variables A-Z, a-z.
STBL	6213	Pointeur de variable chaîne.
FTBL	6215	Pointeur de la table d'évaluation des fonctions.
TTBL	6217	Pointeur de la table des noms qui ne sont pas des commandes.
XTTBL	6219	Pointeur de la table des noms qui sont des commandes.
XITBL	621B	Pointeur de la table de vérification de la syntaxe.
XETBL	621K	Pointeur de la table des commandes d'exécution.
TOV	621F	Sommet des variables (chaînes et vecteurs).
IEXT	6221	Saut vers la syntaxe avec EXT.
EEXT	6224	Saut vers l'exécution d'EXT.
INPLSUB	6227	Appel de MEV à partir d'une ligne comprenant une routine.
EXECSUB	622A	Appel avant l'exécution de chaque ligne.
ZFLAG	622D	Numéro d'indicateur donnant les statuts de Trail et Round.
AUTOFL	622E	Indicateur d'auto-numérotation des lignes, ON/OFF.
CRST	622F	Pointeur de message utilisé pour générer un CR (ØDH) dans la gestion d'écran.
SOB	61FA	Point de départ du Basic.
EOB	61FC	Point d'arrêt du Basic.
POLBUF	61FE	Pointeur de la mémoire tampon utilisé pour la conversation en langage interne.
PRINTD	6200	Adresse du gestionnaire d'impression à l'écran.
LPRNTD	6202	Adresse du gestionnaire d'impression externe.
KEYB	6204	Adresse du gestionnaire de clavier.
KSMS	6231	Simple clé d'entrée.
RPTDLY	6233	Délai de répétition sur le clavier.
LASTK	6235	Dernier caractère du clavier.
STATUS	6236	Utilisé par le gestionnaire du clavier.
RPT	6237	Longueur du temps entre 2 clés répétées.
SHLKT	6239	Utilisé par Shift Lock.
KTBL	623A	26 octets correspondant à des noms raccourcis. Seuls les noms des commandes sont permis. Dans l'ordre A à Z. (simple clé d'entrée)
CURSORX	6254	La composante X du curseur (0 — 126).
CURSORY	6255	La composante Y du curseur (0 — 247).
WINDST	6256	Taille de Window.
CURSTAT	625A	Curseur ON OFF.
INDST	625B	Stockage des couleurs d'encre.
PAPST	625C	Stockage des couleurs papier.
FLASH	625D	Clignotement du curseur vitesse.
INPCUR	625F	Caractères du curseur.
EXFLAG	6261	TRACE, SPEED ON OFF stockés.
GLINE	6262	Saut vers la routine de traçage de ligne.
GOLDX	6265	Coordonnée X du curseur graphique.
GOLDY	6267	Coordonnée Y du curseur graphique.
GNEWX	6268	Stockage de la nouvelle coordonnée avant le traçage de ligne.
GNEWY	626A	Sauvegarder la nouvelle coordonnée de 1 avant le traçage d'une nouvelle ligne.
PROTST	626B	Protégez la sauvegarde.
OUTB	626C	Saut vers la routine qui inscrit les octets de données en A.
CHRTBL	626F	Le masque en C et le déplacement relatif à partir du haut à gauche en HL. Utilise les couleurs déterminées par INK et PAPER.
GPHTBL	6271	Pointeur utilisé pour générer les caractères 32 à 127.
BPERL	6273	Pointeur utilisé pour générer les caractères 128 à 255.
MASK	6275	6 octets par ligne 20 normalement 40 en double largeur.
ERRAM	6288	Masque utilisé par le gestionnaire d'impression = changement avec VDU 20/21.
SLFRAM	628B	Saut vers la fonction lightpen.
BLUBNK	628E	Saut vers la fonction Joystick.
REDBNK	6290	Saut vers la fonction USER0.
GRNBKN	6292	Saut vers la fonction USER1.
NMIRAM	6294	Saut vers la fonction USER2.
RSTRAM	6297	Saut vers la fonction USER3.
ERRTBL	629A	Appelée devant une erreur.
CASLEV	629C	Appelée quand une mise à la ligne apparaît.
COARSE	629D	Pointeur du bank bleu.
FINE	629E	Pointeur du bank rouge.
SYNCLEN	629F	Pointeur du bank vert.
PLYLEV	62A0	Saut ici en NMI.
CASTBL	62A1	Pointeur des messages d'erreurs.
RBIT	62A3	Saut ici en RST 38 H.
WBYTE	62A6	Seuil d'entrée des données sur cassette.
PP	62A9	Ajustement grossier sur la vitesse du cassette.
		Ajustement fin sur la vitesse du cassette.
		Longueur de synchro.
		Référence de voltage pour comparaison ; utilisé pour l'instruction LOAD.
		Pointeur de table utilisé pour générer les ondes.
		Saut vers la routine de lecteur.
		Saut vers l'octet de routine d'écriture.
		Saut vers l'impression sous forme de polonaise inversée.

VERT ALTERNATIF

Le LYNX a quatre blocs de couleur, trois standards : vert, rouge et bleu (qui, combinés donnent les autres couleurs) et une appelée vert alternatif.

Le vert alternatif est utilisé comme un écran séparé pour stocker une page de texte vert, un graphe, ou pour une animation rapide.

Bien qu'il n'y ait pas de commande Basic pour imprimer en vert alternatif, cela peut être fait en changeant les positions mémoires que le Basic utilise pour envoyer des données vers le bloc vert, afin qu'il envoie les données vers le vert alternatif. Cela peut être fait par la commande suivante :

DPOKE & 6292, & 8000

et retour au vert par :

DPOKE & 6292, & C000.

Note : cela ne va pas donner de vert alternatif, pour cela la commande suivante doit être utilisée :

OUT & 80,16

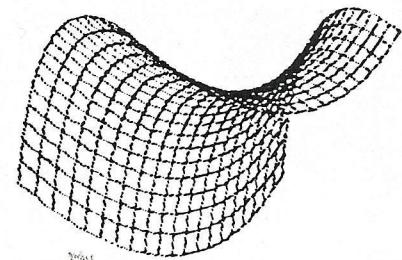
EXPLOSION, ci-dessous, démontre comment cela peut être utilisé.

Cela imprime une explosion verte dans chaque bloc de couleur et puis, avec la boucle REPEAT - UNTIL, les commutent pour donner un effet de flash.

```

100 REM ***** +-----+
110 TEXT
120 DPOKE &6292,8A000
130 CLS
140 PROTECT BLAC
150 PROC LINES(100,RED)
160 PROC LINES(60,YELLOW)
170 DPOKE &6292,&C000
180 PROC LINES(30,WHITE)
190 REPEAT
200 PAUSE 200
210 OUT &0080,4
220 PAUSE 250
230 BEEP 1,5,63
240 PAUSE 100
250 OUT &0080,16
260 UNTIL KEYN
270 END
280 REM ****
290 DEFPROC LINES(X,I)
3010 INP I
310 PROTECT WHITE INK
320 FOR Y=0 TO X
330 MOVE 127,126
340 PLOT 3,RAND(X)-X DIV 2
350 RAND(X)-X DIV 2
360 NEXT Y
370 ENDPROC

```



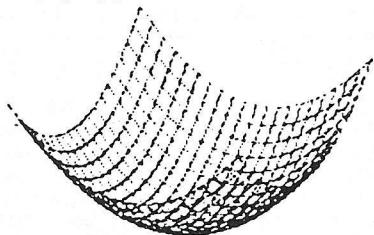
AUTRE PROGRAMME GRAPHIQUE

R. PENROSE

```

100 CLS
110 LET A=1.5
120 LET Y=7.5
130 LET C=COS(RAD(60))
140 LET S=SIN(RAD(60))
150 FOR J=-10 TO 10
160 FPROC TFT(J*Y,-50+J)
170 MOVE P+125,-0+J
180 FOR I=-49 TO 50
190 FPROC TFT(I*Y,P*X)
200 DRAW P+125,-0+200
210 NEXT I
220 NEXT J
230 FOR J=-10 TO 10
240 FPROC TFT(-50+J,Y)
250 MOVE P+125,-0+200
260 FOR I=-49 TO 50
270 FPROC TFT(I*Y,J*Y)
280 DRAW P+125,-0+200
290 NEXT I
300 NEXT J
310 PROTECT GREEN
320 END
330 DEFFPROC TFT(X,Y)
340 LET Z=-(100**2-X**2-Y**2)/
100-500
350 LET F=X*C-Y*S
360 LET A=Y*C+X*S
370 LET Q=A*C+Z*S
380 ENDFPROC

```



Screen dump by R Penrose

LE BASIC LYNX

Il manque au Basic Lynx les instructions ON... GOTU et ON... GOSUB que l'on trouve dans d'autres basic, mais cela est compensé par la possibilité de placer des variables après GOTU et GOSUB. A ce propos parfois, cela n'est pas pratique d'avoir une ligne comme ceci :

200 GOTU 2000 ★ I

Spécialement si vous désirez renommer le tout par la suite (RENUM). Vous pouvez utiliser alors cela :

2000 GOTO L(I) ou le numéro de ligne approprié doit être lu dans le vecteur L () .

TRIANGLES

Ce programme a pour objet de montrer comment le triangle peut être utilisé pour dessiner d'autres formes. Un hexagone, par exemple, peut être six triangles. Ce programme a été écrit pour le LYNX 96 Ko seulement.

```

100 PROTECT A
110 CLS
120 FOR D = 1 TO 6
130 INKO
140 READ X, Y, A, R
150 EXT TRIANGLE 127,124
X,Y,A,R
160 NEXT D
170 GOTO 170
180 DATA 177,44,77,44
190 DATA 207,124,177,44
200 DATA 207,124,177,204
210 DATA 177,204,77,204
220 DATA 44,124,77,204
230 DATA 44,124,77,44

```

SON

J. SPRING

Ce programme peut être utilisé pour produire différents sons. Il peut être employé pour trouver quels emplacements mémoires sont les meilleurs pour les sons que vous désirez.

```

10 FOR B = 1 TO 5000
20 FOR J = 0 TO 100
30 SOUND B.J
40 NEXT J
50 NEXT B

```

Nous vous conseillons de modifier les lignes 10 et 20 pour obtenir d'autres effets sonores.

GRAPHIQUES RAPIDES

Ce petit programme démontre la résolution graphique rapide qui peut être fait sur le LYNX.

```

100 TEXT
110 FOR A = 0 TO 50
120 DRAW RAND (240),
RAND (240)
130 INK RAND (7) + 1
140 INK RAND (7) - INK
150 NEXT A
160 LET I = GETIN
170 END

```

Pour la ligne 110 vous pouvez changer A = 0 FO 50 en A = 0 TON avec une ligne 105 : INPUT N.

ACCELERATION DU TEMPS D'INSCRIPTION

Une critique est souvent faite sur la lenteur d'inscription à l'écran, qui n'est pas un point fort du LYNX. Cela peut être amélioré en Basic de plusieurs façons : soit en utilisant la fonction TEXT (annulée par PROTECT 0) ou la basse résolution graphique ou encore les caractères pré-définis. Mais vous pouvez utiliser la technique suivante . après les instructions PRINT vous placez un CHR\$ (31) ; (notez le point virgule).

Exemple :

```

10 CLS
20 PRINT "Placer du texte sur"
30 PRINT "L'écran comme ceci"
40 PRINT "Est un peu lent"
50 PRINT "Mais voyez plutôt"
70 PAUSE 20000
80 LET AS = CHR$ (31)
90 PRINT "Mais inscrire du
texte"; AS;
100 PRINT "Sur l'écran de cette
façon"; AS;
110 PRINT "Est plus rapide"; AS;
120 PRINT "Vous en conviendrez"
AS;
130 END

```

SON sur le LYNX

Comment appeler BEEP à partir du code Machine :

La procédure suivante va produire un BEEP sur votre LYNX. Premièrement le haut-parleur est connecté par l'envoi de 1 vers le PORT 80 H. Puis le PORT 84 H est alternativement sollicité par des 0 et des nombres allant de 1 à 63 inclus. Cela permet le contrôle de l'amplitude (volume). Le délai entre les envois alternés de 0 et de non-zéro détermine la fréquence. Le nombre de cycles est un paramètre qui détermine le nombre d'exécution de la boucle.

```

OUT (80 H), 1 (allumage)
OUT (84 H), (amplitude)
DELAY
OUT (84 H), 0
DELAY
REPEAT (nombre de cycles)
RETURN

```



PORT 84 H

BIT	7	6	5	4	3	2	1	0
	1	1						

NB : Ne définissez pas le bit 7 dans votre propre routine.

NB 2 : Le volume que vous donnerez est en rapport avec l'adresse 3FH. Pour utiliser le sous-programme qui précède, vous utilisez CALL : dans votre programme. Ce qui va exécuter ce sous-programme à chaque fois que vous appuierez sur RETURN.

Pour appeler BEEP à partir du code Machine :

```

210000 BEEP 1 : LDHL, nombre de cycles
110000 : LDDE, fréquence .
0600 : LDB, volume
D380 : OUT (80), A connection du haut-
parleur.
78 : LD A,B
D5 : PUSH DE
E5 : PUSH HL
C34E09 : JP 094 EH; pas d'appel

```

L'utilisation de SOUND dans des programmes en CODE MACHINE Sound utilise une série de nombre démarrant à l'adresse indiquée et s'arrêtant quand il rencontre en premier une valeur indiquant 0 d'amplitude. La commande fait comme si elle lisait une partition, exécutant strictement ce qu'elle trouve. Par exemple un volume peut être augmenter très progressivement avec la partition suivante :

Puisque 0 termine la partition, vous devez utiliser 1 comme valeur la plus basse pour annuler la fonction de lecture d'amplitude. Si vous incluez dans votre table des nombres supérieurs à 63, ils vont être analysés puis

traités comme des Ø. Aussi la partition s'arrêtera à cet endroit.
Pour utiliser SOUND dans un programme en CODE MACHINE :
21ØØØØ SOUND 1 : LDHL, partition des amplitudes
11ØØØØ LDDE, délai

3E01
D 38Ø
I4
CD 9ØØØ
C9

LDA,1
OUT (80 H), A
Connecte le haut-parleur
INC D
CALL Ø99Ø H
RET



EXT et USER...

EXT vous permet d'ajouter des commandes BASIC. Toutes vos extensions prennent le préfixe EXT. Vous pouvez écrire vos propres routines pour introduire, examiner la syntaxe et l'exécution ; alternativement si vos extensions ont le même format que celles existantes, vous pouvez utiliser les routines en ROM.

Si vous essayez d'utiliser EXT et que cela donne NOT YET IMPLEMENTED, cela ne veut pas dire que les routines EXT ne sont pas dans vos ROM, cela signifie que vous n'avez pas encore défini de commande d'extension.

USER vous permet d'ajouter des fonctions. Pour l'utiliser vous devez comprendre les mécanismes que l'interpréteur BASIC va chercher et travailler une fonction argument. USER prend un seul point argument flottant. Les vecteurs de la routine sont à 627 C, 627F, 6282 et 6285 (voir la liste des variables système). La valeur est passée de WRA 1 à la fin de la fonction.

Y A T'IL DE LA VIE APRÈS NEW ?

Le programme qui vient de subir la fonction NEW a été restauré avec un POKE 26957, 192. Sa première ligne avec le N° 1. De la même façon POKE 26957, 128 à la même effet que NEW.

CONTROLE REMOTE

Si vous ne trouvez pas de contrôle REMOTE sur votre LYNX, ne cherchez pas sur votre LYNX, ne cherchez pas sur votre cassette-phone quand vous sauvegardez et chargez, vous devez avoir un problème de polarité. Les circuits du LYNX ont été conçus pour contrôler une polarité que l'on trouve généralement sur les lecteurs de cassettes avec la fonction REMOTE intégrée, mais pas pour les circuits avec une polarité opposée. N'utilisez pas dans ce cas le jack REMOTE.

CONTROLE DES JOYSTICKS

Une interface pour connection avec des joysticks de type ATTARI est prévu pour octobre au prix de 210 francs TTC.



PROJECTION VIDEO

Il sera possible d'utiliser le LYNX pour diriger des unités projections vidéo en utilisant le signal vidéo composite, plus la masse, du connecteur de crayon optique.

LES COMPLÉMENTS POUR LE 48 K LYNX

Les possesseurs de LYNX 48 K ne vont pas être négligés.

Vous pourrez faire grimper votre 48 K à 96 K, simplement vous bénéficiez de l'avantage : SERVICE +.

La procédure est simple, vous nous envoyez votre 48 K à SEGIMEX, par l'intermédiaire de votre revendeur. Vous devez nous joindre votre nom et adresse et un chèque de francs.

La différence de prix entre le 48 Ko, complété à 96 Ko et la version 96 Ko d'origine, couvre les frais d'envoi et de main d'œuvre ainsi que la vérification complète de l'appareil.

PLUS DE PROBLEME AVEC RESERVE

Il semblerait que l'utilisation de la commande RESERVE au début d'un programme pose un problème.

(RESERVE, sert à réservé un espace pour un ensemble de caractères alternatifs). Car il peut se trouver hors mémoire si le programme s'exécute plusieurs fois successivement. Comme c'est sûrement le cas si le programme commence comme ceci :

10 RESERVE HIMEM -100

Il vous faut rentrer une instruction qui limite l'utilisation :

10 RESERVE 40 857

Si dans un programme précédent un HIMEM supérieur a été défini :

10 IF HIMEM>40852 THEN RESERVE 40852

AMÉLIOREZ VOS PROGRAMMES

Vous pouvez transformer le programme suivant :

10 LET AS = GET \$

20 LET A = VAL (A\$)

30 IFA = Ø THEN GOTO 10

Comme ceci :

10 LET A = GETN -48

20 IF A<OR A>9 THEN GOTO 10

Cela permet aussi l'entrée numérique Ø

AMELIORATIONS R. POATE

PROBLEME — Premièrement :

Un des problèmes que j'ai trouvé sur le LYNX est que l'instruction INPUT ne me donne pas les facilités que je désire : Le LYNX vous permet d'entrer un simple RETURN (en utilisant GET\$) ou une donnée (en utilisant INPUT), mais pas les deux.

Deuxièmement :

Je veux être capable d'édition les données. Si un utilisateur entre une donnée qui rend invalide un test, je veux être capable de lui donner sa chance d'édition la donnée et de la remplacer.

Troisièmement :

Je veux restreindre le nombre de données que l'utilisateur peut rentrer. Une instruction normale permet 240 caractères maximum et vous désirez avoir 127 caractères. Finalement, je veux être capable de prévenir l'entrée d'une seule clé durant un INPUT.

SOLUTION

Ce qui suit est ma solution à ces problèmes. En résumé : c'est une routine en code machine glissée entre la ligne d'input et le code d'exécution d'INPUT. Je place sur l'écran un GS qui permet à l'utilisateur de l'édition. Elle rend inopérant INPUT si l'utilisateur appuie sur RETURN, vous pouvez définir le nombre de caractères que l'utilisateur peut rentrer et cela prévient de l'entrée d'une seule clé.

Commentaires du programme :

Avant tout vous définissez P\$ et si vous voulez éditer le texte, vous définissez G\$ (si non G\$ = " "). Puis vous appelez la procédure de définition G\$ qui à les paramètres suivants :

PROGEN — EDIT (n,i,P,p,W,w,b,d)
avec n comme nombre de caractères permis
i est l'indicateur Vrai Faux pour signifier si un (RETURN) seul est permis

P,p sont les coordonnées du curseur
W,w sont les coordonnées en haut à gauche pour l'introduction de Window

b,d sont la largeur et la profondeur de la fenêtre en caractères
G\$ est le texte concerné.

J'ai trouvé que cela est une bonne idée de définir le nombre de caractères plus 1. Cela tient compte du curseur.

Créez une fenêtre avec au moins 2 caractères de plus que le nombre de caractères permis. Cela permet d'introduire ? de INPUT et l'espace suivant.

Le programme suivant démontre l'utilisation de GEN.EDIT. Il permet de taper du texte puis de l'édition.

```

100 CODE CD CC 10 06 FO 3E FO
90 47 F5 CD EF 10 ES 21 E2 10
22 BC 62 E1 F1 B8 CO ES 21
00 0D 22 00 60 E1 C9
110 PROTECT 0
120 DIM G$(40).P$(20)
VDU 1, BLACK,2,YELLOW,
4,2,GREEN
140 LET G$=" "
150 LET P$="Entrez quelque chose"
160 PROC GEN.INPUT(20,0,3,55,70,
55,12,2)
170 LET P$="Maintenant éditez le"
180 PROC GEN.INPUT(20,1,3,100,
70,100,12,2)
190 PAPER YELLOW
200 PRINT a 3,200 "G$ EST"
210 END
220 REM
230 REM chrs,CR,Prompt,Wind,Br,
Depth
240 DEFPROC GEN.INPUT (n,i,
P,p,W,w,b,d)
250 PRINT a P,p:CHR$(18):P$;
CHR$(18);
260 WINDOW W,b★3+W.w,
w,+10★d
270 VDU 23
280 FOR I=1 TO d
290 VDU 30,31
300 NEXT I
310 VDU 23
320 FOR I=1 TO LEN(G$)
330 POKE &SPFP+1.ASC MID$ (G$,I,1)
340 NEXT I
350 POKE &5FFF+I,&000D
360 DPOKE &62BC,LCTN(100)
370 POKE LCTN(100)+4,n
380 INPUT GS
390 POKE LCTN(100)+4,240
400 IF PEEK(&6000)=i THEN
GOTO 270
410 IF PEEK(&6000)=0 THEN
LET GS=" "
420 WINDOW 3,123,5,245
430 ENDPROC

```

```

100 PROC INSTR
110 REPEAT
120 LET G=0,V=0 e!
130 REPEAT
140 PROC SETUP
150 WHILE d=GREEN AND
e=GREEN
160 PROC MOVE
170 PROC DIRECTION
180 PROC INCR
190 PROC CHECK
200 WEND
210 BEEP 2000,100,63
220 LET N=3000 DIV t
230 IF d=GREEN THEN LET
W=1,G=G+N
240 ELSE LET W=2,V=V+N
250 INK BLACK
260 PRINT a 12,30;"joueur"W;
"gagne"
270 PRINT a 33,5;G; a 90,5;V;
280 PAUSE 3000
290 UNTIL G>99 OR V<99
300 BEEP 200★W,200/W,53
310 PRINT a 3,5;"joueur"W;"est le
meilleur de tous";
320 PAUSE 50000
330 UNTIL FALSE
340 DEFPROC MOVE
350 INK RED
360 DOT A,B
370 INK YELLOW
380 DOT E,F
390 DOT E+1,F
400 INK BLUE
410 DOT P,Q
420 INK CYAN
430 DOT T,U
440 DOT T+1,U

```

```

450 BEEP 74,10,63
460 ENDPROC
470 DEFPROC DIRECTION
480 LET J=H,M=K,N=X,
Z=Y
490 LET H=INT((NOT(INP
(&0080) BNAND &001
0) DIV &0010)-(NOT(INP
(&0080) BNAND &002
0) DIV &0020))
500 LET K=INT((NOT(INP
(&0280) BNAND 8) D IV 8)
-(NOT(INP(&0280) BNAND
&0010) DIV &0010))
510 LET X=INT((NOT(INP
(&0980) BNAND&0020)
DIV &0020)-(NOT(INP
(&0980) BNAND 4) DIV 4))
520 LET Y=INT((NOT(INP
(&0880) BNAND 8) DIV 8)
-(NOT(INP(&0780) BNAND
&0020) DIV &0020))
530 IF H=0 AND K=0 THEN
LET H=J,K=M
540 IF X=0 AND Y=0 THEN
LET X=N,Y=Z
550 ENDPROC
560 DEFPROC INCR
570 LET E=C,F=D,C=A,
D=B
580 LET A=A+H,B=B+K
590 LET T=R,U=S,R=P.
S=Q
600 LET P=P+X,Q=Q+Y
610 LET t=t+1
620 ENDPROC
630 DEFPROC CHECK
640 PROC POINT(A,B)
650 LET d=c
660 PROC POINT(P,Q)
670 LET e=c
680 ENDPROC
690 DEFPROC SETUP
700 WINDOW 3,123,15,245
710 VDU 2, MAGENTA.4.2.
GREEN,23,30
720 FOR B=1 TO 22
730 PRINT
740 NEXT B
750 LET t = 0
760 LET A = 19,B=128,C=A
D=129,E=A,F=130
770 LET P=234,Q=130,R=P.
S=129,T=P,U=128
780 INK RED
790 DOT C,D
800 INK BLUE
810 DOT R,S
820 LET H=0,K=-1
830 LET X=0,Y=1
840 LET d=GREEN,d
850 VDU1.BLACK,2,MAGENTA
860 PRINT 3,5;joueur 1 :"G:
TAB 20," joueur 2 :"V:
870 ENDPROC
880 DEFPROC INSTR
890 WINDOW 3,123,5,245
900 TEXT
910 PRINT TAB 17:CHR$ ('24)
"SNAKE":CHR$ (25):
920 INK BLUE
930 PROTECT YELLOW
940 Prine 9,35; "L'objet du jeu
d'entourer votre..."'
950 PRINT "adversaire, le forçant
ainsi à hurler"
960 PRINT "votre queue, sa
queue ou le nier"
970 PROTECT BLACK
980 VDU 1.RED,31
990 PRINT TAB 15. "controls"
1000 VDU 1, YELLOW,31
1010 PRINT "joueur 1"
1020 PRINT "flèche en bas...
gauche"
1030 PRINT "flèche en haut...
droite"
1040 PRINT "S.....haut"
1050 PRINT "Z.....bas"
1060 VDU 1.CYAN,31
1070 PRINT "joueur 2"

```

```

1080 PRINT "flèche gauche...
gauche"
1090 PRINT "flèche droite...droite"
1100 PRINT "....haut"
1110 PRINT "....bas"
1120 VDU 1.MAGENTA,31
1130 PRINT TAB9,CHR$(18)
appuyer sur une touche pour
continuer; CHR$(18);
1140 LET A$=GET$
1150 ENDPROC
1160 DEFPROC POINT(x,y)
LET a=x DIV 8+32★y,
b=2★★(7-xMOD8)
1180 CALL &0069,&8000+a
1190 LET c=(HL BNAND b) DIVb
1200 CALL &0069,&C000+a
1210 LET c=c+(HL BNAND b)
DIV (b/2)
1220 CALL &0070,&c000+a
1230 LET c=c+(HL BNAND b)
DIV (b/4)
1240 ENDPROC

```

SNAKE
Idée de Chris CYTERA.

LECTURE DE L'ECRAN EN BASIC

Vous avez peut être envie de vérifier vos programmes avant de les lire sur l'écran. Si vous créez un jeu, par exemple, il sera peut-être utile de savoir où la collision doit avoir lieu. Avec les autres micros, vous pouvez simplement employer la commande Basic PEEK pour exécuter cette fonction. Mais le LYNX ne fonctionne pas ainsi. Néanmoins, un effet similaire peut être réalisé par une procédure différente et le programme exposé ici nous démontre comment.

Voici comment il fonctionne : Le but du programme est de garder une trace de ce qui a été sur écrit, afin de le reprendre par la suite.

PROC READ NOT est utilisé par le programme pour trouver (découvrir) si le point spécifié par X et Y est vert ou non. CALL & 0070 à la ligne 2210 appelle la routine de code machine dans MEM. de l'ordinateur. Il utilise l'expression & C000 - Y*32 + X DIV 8. Comme une adresse mémoire sur laquelle il peut lire un octet dans le registre HL (Road only variable). & C000 (ou C000 H) est le début du bloc réservé au vert.

Y ★ 32 + X DIV 8 utilise les coordonnées X et Y pour trouver l'octet dans le bloc. 2 ★★ (7 - X MOD 8) est utilisé en ligne 2220 et 2230 pour extraire le BIT de HL.

Et le résultat, soit 1 ou 0 est affecté à la variable V.

Sketch Pad, si vous utilisez ce programme, un point va apparaître sur l'écran. Cela sert à dessiner doucement ou rapidement sur l'écran. Les flèches du clavier, quand elles sont pressées individuellement ou simultanément, déplacez le point à une des 8 directions.

Si elles sont pressées en même temps que la barre d'espace, le point va se déplacer plus vite.

Si vous appuyez avec la touche RETURN cela va tracer une ligne dans la direction prise.

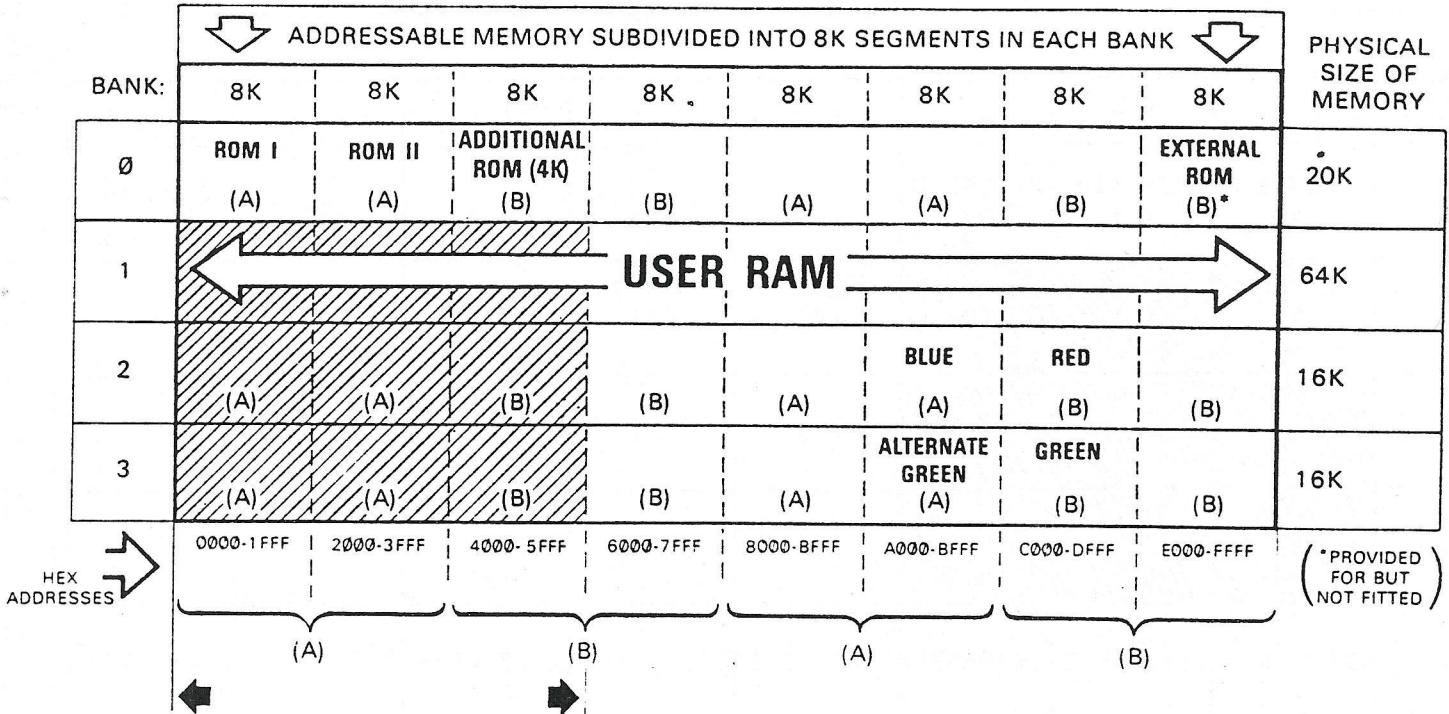
Si vous appuyez sur C, l'écran va s'effacer.

ROUTINE

FAST est une routine en code machine qui sert à augmenter la vitesse d'impression des caractères sur l'écran. Elle peut aussi être adaptée avec POKE ou DPOKE de façon "à :

- Changer la taille des caractères ;
- Changer la couleur d'encre en chacune des 5 couleurs (sans perte de vitesse) ;

LYNX 96K memory map



Partie ombragée (0000 5FFF) est masquée par les ROM.

Les 24 K de RAM utilisateur ici cachées peuvent être employées via les commandes EXT.

Note : A et B (entre parenthèses) indiquent quelles adresses sont reflétées ainsi :
 (A) apparaît 2 fois, aux adresses 0000
 " FFFF et 8000 BFFF ;
 (D) apparaît 2 fois, aux adresses 4000
 7 FFFF et C000 FFFF.

— Place les caractères sur les lignes spécifiques ;

— Imprime et inscrit le vert alternatif. Malheureusement, il y a quelques contreparties : quelques commandes de gestion d'écran sont supprimées. Les commandes PROTECT, INK et PAPER n'ont pas d'effet avec FAST. L'écriture dans les espaces mémoires des couleurs peut être modifiées uniquement avec DPOKE et POKE. De plus le papier est toujours noir.

Les caractères dans FAST ont 8 pixels (1 octet) de large et il n'y a que 32 caractères par ligne maximum.

FAST ne lit pas la mémoire vidéo, la commande VDU 21 (sur écriture ne va pas fonctionner).

Les caractères avec FAST peuvent avoir différentes tailles mais uniquement avec un facteur de 4.

Ainsi dans le cas présent tous les caractères ont 8 points de haut. Les jambages des minuscules sont coupés mais le reste est intact. Pour copier la routine en RAM, placez vous en mode MONITOR et utilisez la commande M (Modify) pour introduire le programme ; puis D (DUMP vers la cassette) pour le sauvegarder.

Enfin retournez en BASIC via J (JUMP).

Puis tapez la commande :

DPOKE & 62B9, (adresse de départ, & 9600 par exemple).

Si vous entrez le code correctement l'encre va devenir jaune. Si vous ne l'obtenez pas, rechargez avec MLOAD.

Une fois que le programme fonctionne, les commandes suivantes vont nous permettre de l'adapter.

POKE (adresse de départ) + & 9, (hauteur 4) cela va déterminer la hauteur des caractères.

POKE (adresse de départ) + & 44, valeur envoyée au PORT & FFFF cela va déterminer sur quel bank vous allez écrire.

```

9600 F5 E5 D5 C5 D9 E5 D5 C5 ueMEYEUE
9608 06 02 CD CE 00 D9 2A 54 ..MN.Y*T
9610 62 7D 6C 26 00 44 29 29 b)1&.D)
9618 29 29 29 0E FF OC D6 03 ))).V.
9620 30 FB 09 01 00 C0 09 D9 0f...e.Y
9628 C5 46 23 4E 23 56 23 5E EFENFVE^
9630 23 E5 D5 D9 01 20 00 E5 feUY..e
9638 09 E5 09 54 5D 09 EB D9 .e.Tj.kY
9640 E1 D1 D9 JE 17 01 FF FF aQY>
9648 ED 79 C1 JE 40 D3 80 D6 ayA>S.V
9650 20 D3 80 70 EB 71 D9 71 S.pkqYq
9658 EB 70 D6 20 D3 80 AF 01 kPV S./.
9660 FF FF ED 79 E1 D9 11 20ayA.
9668 00 19 D9 C1 10 BA C1 D1 ..YA.:AQ
9670 E1 D9 C1 D1 E1 F1 C9 00 aYAQaQI.
9678 00 00 01 00 00 00 00 00 .....

```

& 13 va écrire sur le bank RED/BLUE & 15 va écrire sur le bank GREEN/ALTERNATIVE GREEN

& 17 va écrire sur les deux BANK précédents.

DPOKE (adresse de départ) + & 24, adresse du block de couleur & A000 va écrire sur le BLUE AND OR ALTERNATIVE GREEN, & C000 va écrire sur le RED AND/OR.

POKE (adresse de départ) + & 4c & 4c va inscrire GREEN. & 4c, & 50 va inscrire ALTERNATIVE GREEN.

Quand vous utilisez FAST, vous trouverez que cela ne va effacer qu'une partie de la ligne que l'on recouvre. Cela peut être corrigé en utilisant WINDOW 0, 93, 1,251 vous permettant 31x25 caractères par écran.

Malheureusement FAST n'est pas compatible à 100 % avec le BASIC.

Et si vous utilisez des virgules avec l'instruction PRINT cela va planter la machine. Si vous les bannisez il n'y aura plus de problème.

À présent l'espace entre des lignes de texte est de 10 points. FAST peut vous permettre d'en économiser 2. En ajoutant cette routine :

```

★ M 9700
9700 < 00 > D6 02 C9
9703 < FF >
Utilisez le Moniteur puis revenir au Basic.
Et tapez :
DPOKE & 628 c . & 9700
POKE & 628 b . & C3
Pour revenir à 10 points d'espaces tapez :
POKE & 628 b . & C9
Afin d'obtenir 31 colonnes X 31 lignes
tapez :
WINDOW 0.93,0.248

```

UN ECRAN RAPIDE

Si vous donnez à votre LYNX un travail qui n'est pas nécessaire, la vitesse de maniement sera réduite. Aussi, c'est une bonne idée de prendre un minimum de précautions quand vous écrivez sur l'écran. Voici quelques tuyaux qui vous aideront à intensifier le travail du LYNX.

PROTECT (protéger) toutes les couleurs sauf celle que vous utilisez. Faites suivre toute instruction PRINT de ";".

Autrement l'ordinateur devra exécuter toute la procédure, c'est-à-dire inscrire toute la ligne. CHR\$ (31) est un bon substitut. Utilisez la commande TAB pour positionner le texte. Si cela n'est pas possible, utilisez CHR\$ (12) pour accélérer le mouvement

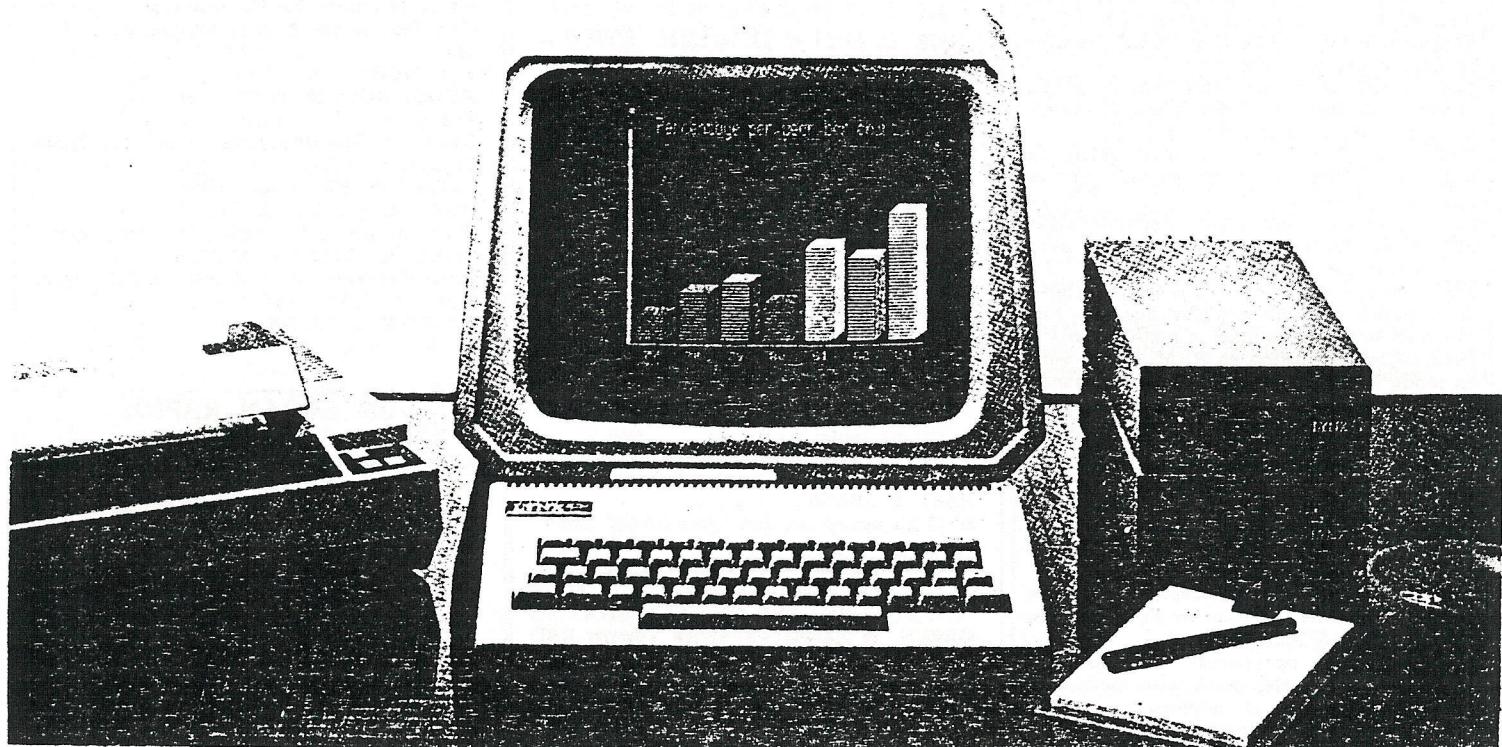
Carte mémoire du Lynx 48 k

BANK	8 K	8 K	8 K	8 K	8 K	8 K	8 K	SIZE OF PHYSICAL MEMORY
Ø	ROM I (A)	ROM II (A)	EXTENDED ROM (B) *	(B)	(A)	(A)	(B)	EXTERNAL ROM (B) *
1	(A)	(A)	(B)	USER (B)	RAM (A)	(A)	(B)	16 K
2	(A)	(A)	(B)	(B)	(A)	BLUE (A)	RED (B)	16 K
3	(A)	(A)	(B)	(B)	(A)	ALTERNATE GREEN (A)	GREEN (B)	32 K
a cesse d'être utilisé par la basic		0000-1FFF	2000-3FFF	4000-5FFF	6000-7FFF	8000-9FFF	A000-BFFF	C000-DFFF E000-FFFF
		(A)	(B)			(A)		(B)

Aire ombragée masquée par les ROM. ainsi les reflets (A) et (B) sont apparents que lorsque les ROM sont déconnectées.

Note : A et B (entre parenthèses) indiquent quelles adresses sont reflétées :
 (A) apparaît 2 fois, aux adresses 0000-3FFF et 8000-BFFF;
 (B) apparaît 2 fois, aux adresses 4000-7FFF et C000-FFFF.

* prévu non pré



LOGICIELS GRATUITS

Toutes les machines en 96 Ko, soit qu'elles aient été acheté neuve ou été complété, auront un contrôleur d'impression en ROM. Mais vous aurez la possibilité de gérer une imprimante avec une machine 48 Ko. Pour une impression en série, vous avez besoin d'un cable et du logiciel. Le cable sera commercialisé au prix de 72 francs et le logiciel sera une cassette gratuite. Pour les impressions en parallèle vous avez besoin d'une interface disponible en octobre au prix de 700 francs.

128K LYNX

Conçu pour du travail sérieux dans les domaines scientifique et d'éducations, le 128K LYNX qui sera lancé à la fin de l'automne est le plus haut de la gamme pour 1983. La capacité essentielle de ce 128 Ko est de fonctionner sous CP M et ainsi ouvrir la porte à une énorme bibliothèque de logiciels.. La sortie RS 232 C type permet la communication avec d'autres périphériques et d'autres ordinateurs.

EXTENSION DE MEMOIRE

L'extension de mémoire inclue une gestion d'écran encore plus puissante : doublement de la résolution et augmentation de 40 à 80 caractères sur chaque ligne, permettant l'utilisation de traitement de texte.

COIN CASSETTE

Il y a deux aspects dans la vie d'un nouvel utilisateur de micro qui peuvent réduire la perte des cheveux.

La première façon de perdre ses cheveux est de passer 3 heures sans succès, à être de charger un programme à partir d'un cassétophone.

La deuxième, arrive quand vous perdez 100 lignes de programmes car le cassétophone vient juste de s'arrêter quand vous avez appuyer sur l'instruction SAVE. Des problèmes de cette nature viennent généralement du cassétophone. Les niveaux de volume et de tonalité sont imparfaitement définis. Des corps étrangers sur la tête de lecture ont peut-être effacé le signal de début de programme et il ne peut être reconnu par l'ordinateur.

Ou la faute provient de la tête de lecture écriture elle-même, particulièrement si vous utilisez un vieux cassétophone. La fiabilité des têtes tend à diminuer après quelques temps : cela produit des blancs sur la cassette. Ce qui n'est pas apparent lorsque vous écoutez de la musique.

Les signaux du programme transmis par l'ordinateur doivent être transmis clairement et à un taux constant. Mais s'il y a un blanc sur la cassette l'ordinateur va empêcher le stockage.

La solution au problème du cassétophone est très difficile ; quelques machines sauvegarderont, ne chargeront pas ; d'autres feront le contraire ; et certaines ne vont sauvegarder et charger que leurs programmes sans accepter les logiciels pré-enregistrés. Les lecteurs de cassettes ont été conçus pour l'audition bien avant que les ordinateurs utilisent eux aussi les signaux d'enregistrement et de fait certains possèdent un signal de phase inversé pour la tête de lecture et d'autres un signal de phase non inversé.

Cela fait peu de différence pour un haut parleur, mais crée toute la différence dans le monde de l'ordinateur. C'est pour cette raison que les cassettes enregistrées sur un magnétophone ne vont pas nécessairement fonctionner sur un autre.

Ecrivez-nous pour nous signaler les caractéristiques et les références de votre appareil s'il charge et sauvegarde de façon fiable. Nous pouvons vous conseiller déjà les cassétophones suivants :

Elftone EL 8019 (Compucorder)
EL 8025
Dixons TR 30
TR 12
Bush 3150
Sharp RD 610 K
RD 620 E
Ferguson 3 T 27
Panasonic RQ 8200
C 2120
Slimline
Sony TCMI21
TCM3
Binatone PIPER (Mini-Cassette)
W H SMITH CCR 800
Tensai CR 106
Radio Shack CCR 81
Golem DI 1
Go'lem CE 1
Golem SE 1

Voici également quelques petits trucs pour essayer d'adapter votre cassette, si celui-ci ne charge pas. (Il existe deux façons de ne pas charger malgré la contradiction : soit il apparaît rien à l'écran, soit vous avez un premier guillemet, dans ce second cas vous êtes proche du niveau de transmission correct, il faut un petit ajustement). Vous avez juste à côté de la fente d'introduction des cassettes, un petit trou par lequel, lorsque l'on introduit un tournevis de joaillerie, on atteint la vis de réglage de la tête de lecture.

Vous introduisez la cassette non déchiffrée, vous appuyez sur PLAY après avoir enlevé le gros Jack noir (EAR) vous entendez alors un bruit très fort qui est audible. Il faut alors tourner la vis de réglage de la tête de lecture de façon à obtenir le son le plus aigu possible.

Vous vous replacez ensuite dans les conditions normales de chargement et vous devez obtenir un résultat, si votre cassétophone est correct.

Autres trucs en vrac :

— Si votre programme n'est pas sauvegardé, ou si il ne se charge pas et que vous désirez arrêter l'opération. Appuyez sur ESC, en ayant laissé le cassétophone en fonctionnement (La touche PLAY enfoncée).

— En cas de grande difficulté sur un lecteur récalcitrant appuyez sur REWIND un bref instant et presque en même temps sur ESC (La touche PLAY étant toujours enfoncée).

— Si votre télécommande (Remote) ne fonctionne pas, il faut inverser les fiches DIN 6 et 7, c'est-à-dire les plus en haut à gauche et à droite.

LE SCROLLING

TECHNIQUE ET PROGRAMME

L'impression à l'écran du LYNX n'a pas de défilement normalement, car le composant utilisé pour contrôler la gestion vidéo — le CATHODE RAY TUBE CONTROLLER (CRTC) — travaille en unités de 4 points verticalement. Le caractère LYNX est défini avec 10 Points de haut.

La gestion d'écran est dirigée par un composant spécial — le 6845. Ce CRTC a au total 18 registres — qui gère les emplacements mémoire dans le composant afin de vérifier ses fonctions.

Chaque registre peut stocker un nombre binaire, incluant 8 bits libellés de 0 à 7 de droite à gauche. Chaque bit peut avoir une valeur 1 ou 0 (ensemble haut ou ensemble bas) ainsi par exemple :

BIT : 7 6 5 4 3 2 1 0

VALUE : X00000010

Le bit 2 fait partie de l'ensemble haut. Tous les autres bits sont de l'ensemble bas. La

valeur de chacun de ces bits est importante. Le fait de changer l'un deux va modifier l'opération.

COMMENT OBTENIR LE DÉFILEMENT (SCROLLING)

Envoyer les instructions au CRTIC et obtenir le défilement de l'écran nécessite d'envoyer des instructions à 2 ports Z80 en utilisant la commande BASIC : OUT.

Les ports Z80 déterminant le défilement sont 86 H (H = HEXADECIMAL) et 87 H : 86 est utilisé pour sélectionner le registre CRTIC, 87 pour envoyer les données vers les registres.

Nous sommes concernés par 2 registres 12 et 13. Les bits 0 à 4 du registre 13 affectent le déplacement horizontal de l'écran. Si, ils sont tous définis dans l'ensemble bas, la position est normale.

Pour changer la position, vous devez changer un bit ou plus vers 1. La somme du déplacement est déterminée par la valeur de ces 5 bits, lu comme un nombre binaire à 5 décimales.

Ainsi, par exemple, OUT & 86, 13 va sélectionner le registre 13, puis OUT & 87,2 va envoyer la valeur 2 (10 en binaire) vers le registre 13. Le déplacement vertical est déterminé par la valeur des bits 5-7 du registre 13 et des bits 0-2 du registre 12. Faites attention de ne pas positionner les bits 3 - 5 du registre 12, dans l'ensemble haut, ou la machine va faire crash ! Si vous posez une valeur 1 dans le registre 13, l'écran va défiler horizontalement de 8 points ; si vous donnez une valeur 32, il va défiler de 4 points verticalement.

DÉFILEMENT AVEC STAR ROVER

Star Rover est un petit programme qui démontre les possibilités de défilement. Il n'a pas de fin : quand vous n'avez plus d'énergie vous donnez un message d'erreur. Vous pouvez essayer d'ajouter une autre procédure, FINDEJEU (appelé en ligne 348) pour donner une fin plus professionnelle au jeu. Une table montre les fonctions de chaque variable.

Le défilement est manié comme ceci : dans la procédure JUMP, le Port 86 H a déjà été défini pour le registre 13 par la routine en langage machine appelée en ligne 130. La ligne 220 alors utilise le Port 87 H pour envoyer les valeurs de 2 vers le registre 13, qui alors fera défiler l'écran vers le haut. Après avoir changer la section du bas de l'écran et avoir fait apparaître les vaisseaux de l'espace, la valeur de 2 est augmentée de 32 ainsi quand la procédure SAUT est appelée, l'écran se déplace vers le haut à nouveau.

Quand l'écran a été déplacé 16 fois — utilisant pour les 4 bits : 5 à 7 l'ensemble 1 — défiler l'écran d'une autre unité nécessite que tous ces bits doivent être à 0 et le bit 0 du registre 12 doit être à 1.

Pour faire ceci en BASIC, vous avez 4 commandes :

OUT & 86,12

OUT & 87, z

OUT & 86,13

OUT & 87, 0

et l'écran va se déplacer pour parvenir à cela, les instructions ont été introduites en langage machine et cette routine est contenue en ligne 1030.

La valeur de Z es entrée dans la routine en ligne 120 par POKEZ + 6,z.

Pour d'autres jeux, vous pouvez changer la valeur envoyée au registre &" par un POKE#L + 13,z.

TABLE DES VARIABLES

A\$.. Le caractère \$ défini le vaisseau de l'espace.

C.. Paramètre de PROC PERCUTION ;

Donne le code ASCII de l'objet heurté.

C(.. Caractère des Astéroïdes, etc... dans

la ligne spécifié par indice.
 E.. Niveau d'énergie du vaisseau de l'espace.
 L.. Affectation de la ligne 1030.
 M.. Position verticale dans la fenêtre en bas de l'écran (unités de 4 points).
 P ().. Position horizontale de l'astéroïde, etc... sur la ligne spécifié par un indice.
 V et W.. Compteurs pour BEEP dans les PROC collision entre autres.
 X.. Compteur pour PROC INITIAL et horizontal position du vaisseau spatial.
 Y.. Position verticale du vaisseau.
 Z.. Valeur à entrer dans le registre 13 pour le défilement.
 Valeur à entrer dans le registre 12 pour le défilement.

STAR ROVER - Le Programme

Vous êtes le commandant d'un vaisseau de l'espace intergalactique. Avec un niveau d'énergie qui baisse dangereusement et vos moteurs détraqués, vous êtes perdus dans un champs d'astéroïdes. En utilisant les flèches vous gouvernez à travers l'écran, évitant les astéroïdes. Heureusement quelques uns des fragments ont une haute valeur énergétique et en les interceptant vous pouvez gagner de l'énergie supplémentaire. La collision avec des astéroïdes vous prend de l'énergie. Et bien quand le niveau d'énergie atteint 0...

```

10 TEXT
20 PROTECT YELLOW
25 INK BLUE
28 REM ★★★★ Etoiles
30 FOR V = 1 TO 200 STEP 5
40 : DOT RAND (V), RAND (256)
50 NEXT V
80 WINDOW Ø, 127, Ø, 255
90 PROGINITIALE
110 REPEAT
120 POKE L + Ø,z
130 CALL L
140 REPEAT
150 PROC SAUT
160 UNTIL Z = 256
170 LET Z = Ø, z = (z + 1)
MOD 8, Y = Y MOD 256
180 UNTIL FALSE
200 DEFPORCAUT
220 OUT & ØØØØ, Z
225 PROC ASTEROIDE
230 IF INP (& ØØØØ) <> 255
THEN PROC ACTION
232 PROTECT CYAN
235 PRINT a X,Y; A$;
240 LET Y = Y + 1, M = (M+1)
MOD 64, Z = Z + 32
250 ENDPROC
300 DEFPROC ACTION
310 IF INP (& ØØØØ) = 251
THEN LET X = X - 1
320 IF INP (& ØØØØ) = 223
THEN LET X = X + 1
330 IF X = 2 THEN LET X = 3
340 ELSE IF X = 118 THEN LET
X = 117
342 REM ★★★★ PERTE
D'ENERGIE
345 BEEPE, 1000 DIV E, 63
347 LET E = E - 1
348 IF E < = Ø THEN PROC
FINDEJEU
350 ENDPROC
400 DEFPROC ASTEROIDE
402 PROTECT BLUE
405 PRINT a P(M), M ★ 4, "
420 LET P (M) = RAND (126),
C(M) = 138 + SGN (RAND
(17) - 8) ★ 2
425 IF C(M) <> 138 THEN
INK GREEN
428 PROTECT WHITE - INK
430 PRINT a P(M), M ★ 4;
CHR$ (C(M));
440 REM ★★★★ EXAMEN
DE LA COLLISION
450 IF (P((M - 54) MOD 64)
>= X AND P ((M - 54)
```

```

MOD 64) < = X + 4)
THEN PROC COLLISION
ENDPROC
502 DEFPROC COLLISION (C)
IFC ((M - 54-c) MOD 64)
= 138 THEM LET C = 2Ø
505 FOR V = 4 TO 1 STEP - Ø,5
510 FOR W = 1 + C TO 31
520 BEEP W★V,4,31★V + 1 - W
530 NEXT W
535 NEXT V
536 IF C = 2Ø THEN LET
E = E + 2Ø
538 ELSE LET E = E - 5Ø
539 IF E < Ø THEN PROC
FINDEJEU
540 ENDPROC
1000 DEFPROC INITIALE
1005 DPOKE GRAPHIC, LCTN
(1Ø9Ø)
1006 DIM A$(9), P(63), C(63)
1008 LET Y = 2Ø, M = 61,
Z = Ø,z = Ø,bzLSTN(1Ø3Ø),
E = 15Ø
1009 INK RED
1010 PROTECT CYAN
1011 REM ★★★★ Cette boucle
FOR - NEXT lit les données
en ligne 1Ø2Ø et les utilise
pour déterminer quels caractères
et quels code de contrôle
le vaisseau de l'espace
(AS) va faire apparaître.
1012 FOR X = ITO 9
1014 READ V
1016 LET A$ = A$ + CHR$(V)
1018 NEXT X
1020 DATA 1,2,128,13Ø,22,22,1Ø,
132,134
1021 FOR X = Ø TO 63
1022 LET P (X) = Ø
1023 NEXT X
1025 ENDPROC
1027 REM ★★★★★★★★★★★
1028 REM — Programme en langage
machine pour l'ensemble du
registre 12 avec la valeur du
8° byte et du registre 13 avec
le 16° byte. C'est Ø dire le
défilement de l'écran.
1030 CODE 01 86 ØC ED 41 3E
ØØ D3 87 Ø4 ED 41 3E ØØ
D3 87 C9
1079 REM ★★★★★★★★★★
1080 REM : Code qui définit les
caractères 128 à 148 : lignes
1090 — 116Ø = Vaisseau de
l'espace
1170 — 118Ø = Astéroïdes etc
1090 CODE 00 00 00 00 00 00
00 00 07 07 03 00
1100 CODE 00 00 00 00 00 20
20 20 3C 38 30 00
1110 CODE 01 01 09 09 ØF ØB
01 01 01 01 00 00
1120 CODE 30 30 30 28 28 30
20 00 00 00 00
1170 CODE 2C 18 34 18 00 00 00
00 00 00 00 00 00
1175 CODE 24 18 18 24 00 00 00
00 00 00 00 00
1180 CODE 14 2Ø ØC 20 00 00 00
00 00 00 00 00
2000 REM — Une routine pour
normaliser la gestion d'écran
2005 OUT & ØØØØ,12
2010 OUT & ØØØØ,Ø
2020 OUT & ØØØØ,13
2030 OUT & ØØØØ,ØØ
2035 WINDOW 3,123,2,245
```

LE RESET

après « s'être planté »
 Retrouver le contrôle du LYNX après s'être planté, nécessite de déconnecter l'appareil. Faites la déconnection au NIVEAU DE L'INTERRUPTEUR DE L'ALIMENTATION et NON AU NIVEAU DE LA MACHINE. Mais il y a une commande qui réinitiali-

lise le LYNX. Si le crash n'est pas trop mauvais, et qui vaut la peine d'être essayé, même si cela n'apparaît pas à l'écran quand vous le tapez.

Tapez :

CALL Ø

Vous pouvez récupérer les instructions en tapant POKE 26957, 192, pour la plupart des cas.

IMPRIMANTES ET LE LYNX

Imprimante série

Le LYNX a besoin d'un logiciel additionnel pour gérer l'imprimante avec un cordon, spécialement pour la SEKOSHA GP 250. Le logiciel et le guide seront disponibles en novembre au prix de 72 francs.

Imprimante parallèle

Elles seront gérées à partir d'une interface connectée au bus d'extensions. Elles seront disponibles en novembre au prix de 700 F.

SAUVEGARDER LES VARIABLES

Voici une petite routine qui peut être ajoutée à tous les programmes en ligne de code et qui sauvegardera des variables, niveaux de PROTECT et couleurs INK et PAPER.

```

1000 CODE EB 21 F6 61 E5 2A IF 62
E5 21 ØØ ØØ C3 FF 3E (RETURN)
1010 REM "nom du programme"
1020 CALL LCTN (1ØØØ), LCTN (1Ø1Ø)
1030 REM - le programme va s'exécuter
à partir d'ici
```

SAUVEGARDER LE PROGRAMME EN UTILISANT : GOTO 1Ø2Ø

(Le nom du programme est contenu en ligne 1Ø1Ø) Pour changer le programme utilisez : M LOAD "nom du programme"). Le programme va automatiquement s'exécuter à partir de la ligne suivante qui contient le CALL LCTN.

PROGRAMMES :

PROGRAMMEURS - VOICI VOTRE PAGE ! SON par Don Thomasson

Ce programme démontre comment le LYNX peut jouer un morceau de musique. Quand la question « KEY ? » apparaît à l'écran, pressez un nombre pour continuer. Les lignes 120 à 160 définissent la racine douzième de 2 (2★(1/12)) dans B (), cela définit des intervalles de demi-ton en terme de fréquence. Les lignes 170 - 240 définissent une table d'où sont extraits les demi-tons utilisés avec les clés introduites. A la ligne 250, une clé est déterminée en introduisant un nombre, qui doit être un entier (la fonction arrondie (round) intervient, si ce n'est pas un entier). Le morceau est défini par les lignes 350 à 430 puis joue dans la clé choisie. La ligne 430 est une terminaison standard. Ce programme peut servir de base à la conception d'autres morceaux.

```

1ØØ REM définition de l'échelle
musicale.
11Ø CLS
12Ø LET A = 2 ★★ (1/12)
13Ø DIM B (48), C (32)
14Ø FOR X = Ø TO 47
15Ø LET B (X) = A ★★ X
16Ø NEXT X
17Ø FOR X = 1 TO 32
18Ø READ C (X)
19Ø NEXT X
20Ø DATA 1,3,5,6,8,10,12,13
21Ø DATA 15,17,18,20,22,24,25
22Ø DATA 27,29,30,32,34,36,37
23Ø DATA 39,41,43,44,46,48,49
24Ø DATA 51,53,55
```

```

250 INPUT "KEY": K
260 RESTORE 350
270 REPEAT
280 READ T
290 READ J
300 LET E = C(D) + K
310 LET F = 1 B (E)
320 BEEP 1000★F.T.★50 F, 63
330 UNTIL T = 0
340 GOTO 250
350 DATA 1,8,1,8,1,9,1,8,1,8,2,5
360 DATA 1,6,1,5,1,6,1,7,2,8,2,8
370 DATA 1,8,1,8,1,8,1,9,1,8,1,8,2,5
380 DATA 1,6,1,5,1,6,1,7,2,8,2,8
390 DATA 1,12,1,11,1,10,1,9,1,10,1,
      9,2,8
400 DATA 1,6,1,5,1,6,1,7,2,8,2,8
410 DATA 1,5,1,5,1,6,1,7,1,8,1,8,2,9
420 DATA 1,12,1,11,1,10,1,9,2,8,2,
      11,4,8
430 DATA Ø1

```

AIR RAID

amélioration par T. GRIFFITHS

Ajoutez ces quelques lignes au programme AIR RAID sur la cassette introduction du LYNX et l'ordinateur vous donnera un score de 20 points pour chaque bloc de building que vous détruisez.

Si vous atteignez le sol mais heurtez un building vous obtenez 1000 points de bonus.

Si vous atterrissez correctement vous obtenez un bonus compris entre 2000 et 9998 points.

```

305 LET h = 3000.i = Ø.S = Ø,
      X = Ø.H = Ø
332 IF H = 180 AND X = 1Ø8
      THEN GOTO 340
334 S = Ø
336 i = i
542 IF H = 180 AND X = 1Ø8
      THEN S = S + (RAND 4000)
      +1000) ★2
544 IF H = 180 AND X
      1Ø8 THEN S = S + 1000
735 S = S + 20
1115 PRINT a 10.10: "Plus haut
      score"; h
1061 IF i = Ø OR (H = 180 AND
      X = 1Ø8) THEN GOTO 1069
1063 PRINT a 40,110: "Votre score":S
1065 IF S>h THEN PRINT TAB 8;
      CHR$(18) "Vous avez réussi
      le plus haut score"+CHR$(18)
1067 IF S>h THEN h = S
1068 PAUSE 20000
1069 CLS

```

SPLASHDOWN

par DON THOMASSON

Un court programme graphique qui démontre les capacités couleurs spéciales graphiques du LYNX.

```

100 DIM H (255), L (255)
110 LET B = - SIN (Ø,4),
      C= COS (Ø,4)
120 FOR N = Ø TO 255
130 LET H (N) = Ø
135 L (N) = Ø
140 NEXT N
150 FOR Y = 20Ø TO - 20Ø
      STEP - 1Ø
160 FOR X = - 120 TO 120
170 LET R = SXR(X★X+y★Y)
180 IF R = Ø THEN GOTO 20Ø
190 LET Z = SIN (R/10)
      ★ 1000/R
200 INK INT (R/50) + 2
210 LET U = 128 + X,V = 82
      + INT (B★Y + C★Z)
220 IF V > = L(U)
      THEN GOTO 250
230 DOT U, 240 - V
240 LET "L" (U) = V
250 IF V < = H(U)
      THEN GOTO 280
260 DOT U, 240 - V

```

```

270 LET H (U) = V
280 NEXT X
290 NEXT Y
300 STOP

```

LEVE DE LUNE par Don THOMASSON
Une simple démonstration de dessin d'un cercle, qui illustre aussi les fonctions trigonométriques inverses.

```

100 CLS
110 INK5
120 LET X = 128,Y = 12Ø,
      R = 50
130 FOR N = 1 to R
140 LET A = N-R
150 LET B = ARCCOS(A/R)
160 LEC C = SIN(B) ★R★1.2
170 MOVE X + A,Y - C
180 DRAW X + A,Y, + C
190 MOVE X - A,Y, - C
200 DRAW X-A,Y+C
210 NEXT N
220 VDU 24
230 PRINT a 35,10;" La lune est
      levée "
240 VDU 25
250 INK WHITE
260 GOTO 260

```

DOWN THE TUBE

Un petit programme graphique qui montre une façon de dessiner des cercles de façon inhabituelle. Les commandes de cercles sont ôtées, vous avez seulement besoin des coordonnées X et Y.

```

1 PROC CIRCLE
2 VDU 4,21,25
3 PRCC BRAIN
4 FORX = Ø TO 19Ø STEP 1Ø
5 INK X + i
6 FOR A = Ø TO 36Ø
7 PLOT 4,30+(M(A) + X),
      3Ø + (N(A) + X)
8 PLOT 4,30+(M(A) + (19Ø
      - X)), 3Ø + (N(A) + X)
9 NEXT A
10 NEXT X
11 PROC DUNNIT
12 G = GETN
13 END
14 DEFPROC CIRCLE
15 R = 25,r = 25
16 DIM M (36Ø)
17 DIM N (36Ø)
18 CLS
19 VDU 24
20 PRINT a 40,20; "Je pense"
21 PROC BRAIN
22 FOR A = Ø TO 36Ø
23 B = A★PI/18Ø
24 C = R★COS(B)
25 D = r★SIN(B)
26 M(A) = C
27 N(A) = D
28 NEXT A
29 ENDPROC
30 DEFPROC BRAIN
31 FOR F = Ø TO 10Ø
32 BEEP RAND (3ØØ) + 1,3,63
33 NEXT F
34 ENDPROC
35 DEFPROC DUNNIT
36 FOR J = Ø TO 1
37 FOR S = 1ØØ TO Ø STEP
      - 1Ø
38 FOR U = Ø TO 10Ø
      STEP 1Ø
39 BEEP U + S,10,63
40 NEXT U
41 NEXT S
42 NEXT J
43 ENDPROC

```

A TRAVERS BEETHOVEN

par Chris SAFFIN

Cette table donne les fréquences et les longueurs d'ondes pour les notes. Les notes

produites ne sont pas strictement parlant, musicalement exactes.

NOTE	FREQ.	W LEN
A LA	110	909
A # LA =	116.6	857.3
B SI	123.5	809.7
C DO	130.8	764.5
C # DO =	138.6	721.5
D RE	146.8	681.2
D # RE =	155.6	642.7
E MI	164.8	606.8
F FA	174.6	572.7
F # FA =	185	540.5
G SOL	196	510.2
G # SOL =	207.7	481.5
A LA	220	454.5
A # LA =	233.3	428.6
B SI	246.9	405
.	261.1	383
C DO	277.2	360.7
D RE	293.7	340.5
D # RE =	311.1	321.5
E MI	329.6	303.4
F FA	349.2	286.4
F # FA =	370	270.3
G SOL	392	255
G # SOL =	415.3	240.8
A LA	440	227.3
A # LA =	466.2	214.5
B SI	493.9	202.5
C DO	523.2	191
C # DO =	554.4	180.3
D RE	567.3	170.3
D # RE =	622.3	160.7
E MI	659.3	151.7
F FA	698.5	143.2
F # FA =	710	135
G SOL	784	127.5
G # SOL =	830.6	120.4
A LA	880	113.6
A # LA =	932.3	107.3
B SI	987.8	101.3
C DO	1046.5	95.5
C # DO =	1108.7	90.2
D RE	1174.7	85.3
D # RE =	1244.5	80.3
E MI	1318.5	75.8
F FA	1396.9	71.6
F # FA =	1480	67.5
G SOL	1568	63.8
G # SOL =	1661.2	60.2
A LA	1760	56.8
A # LA =	1864.6	53.6
B SI	1975.5	50.6
C DO	2093	47.8
C # DO =	2217.5	45
D RE	2319.3	42.6
D # RE =	2489	40.2
E MI	2637	37.9
F FA	2793.9	35.8
F # FA =	2959.6	33.8
G SOL	3135.9	31.9
G # SOL =	3322.4	30

CONTOUR D'HYPERBOLE

C. CYTERA

Ce programme calcule un numéro de couleur en fonction de chaque paire de coordonnées existantes sur l'écran et produit un point dans cette couleur et à des coordonnées. Le résultat est un dessin représentant les contours de la carte d'une hyperbole, donnée par l'expression : XXX + X★Y - Y★Y.

Vous pourrez également essayer l'expression X★X + (X + Y) ★, qui produit une ellipse.

```

100 PROTECT BLACK
110 CLS
120 FOR X = -128 TO 127
130 FOR Y = -124 TO 123
140 INK (X★X) + (X-Y)
      DIV 10Ø
150 DOT X + 128, Y + 124
160 NEXT Y
170 NEXT X
180 LET A$ + GET$
190 RUN

```