

Micro
Informatique
Standards AMSTRAD

CPC infos

JEU :
**CHARLIE
BOND
SOUND**

DISQUES :
EXPLOITEZ
LES FORMATS
SPECIAUX



M 2604 - 34 - 23,00 F



Mensuel N°34
Juillet 1991

SOMMAIRE

N°34

-
- 3 Fanz' infos
 - 8 Actualité
 - 11 Formatages spéciaux
 - 22 Chronique du téléchargement
 - 25 Easy Screen
 - 34 Banc d'essai Music Machine
 - 36 Trucs et rectificatifs
 - 40 Charlie Bond Sound
 - 48 Ecriture Arabe

Bulletin d'abonnement _____ 54

Remerciements à la société TITUS pour la photographie de couverture (jeu Préhistorik).



CPC infos

La Haie de Pan - 35170 BRUZ
Tél : 99.52.98.11 - Télécopie : 99.52.78.57
Serveurs : 3615 ARCADES et 3615 MHZ
Terminal NMPP E83
Gérant, directeur de publication
Sylvio FAUREZ

REDACTION

Rédacteur spécialisé
Olivier SAOLETTI
Secrétaire
Catherine FAUREZ

Ont collaboré à ce numéro :

Guy DUBUS, Yannick GOUR
Michel LA ROSA, Claude LE MOULLEC

FABRICATION

Directeur de fabrication
Edmond COUDERT
Maquettiste
Jean-Luc AULNETTE

ABONNEMENTS

Nathalie FAUREZ - Tél. : 99.52.98.11

PUBLICITE

IZARD Création
15, rue St-Melaine
35000 RENNES - Tél. 99.38.95.33

GESTION RESEAU : NMPP

SORACOM E. COUDERT
Fax. 99.52.78.57 - Terminal E83

SOCIETE MAYENNAISE D'IMPRESSION 53100 MAYENNE

Les articles et programmes que nous publions dans ce numéro bénéficient, pour une grande part, du droit d'auteur. De ce fait, ils ne peuvent être imités, contrefaits, copiés par quelque procédé que ce soit, même partiellement sans l'autorisation écrite de la Société SORACOM et de l'auteur concerné. Les opinions exprimées n'engagent que la responsabilité de leurs auteurs. Les différents montages présentés ne peuvent être réalisés que dans un but privé ou scientifique mais non commercial. Ces réserves s'appliquent également aux logiciels publiés dans la revue.

CPC infos est édité par les Editions SORACOM, éditrice de PCompatibles Magazine et MEGAHertz Magazine. RCS Rennes B 319 816 302.

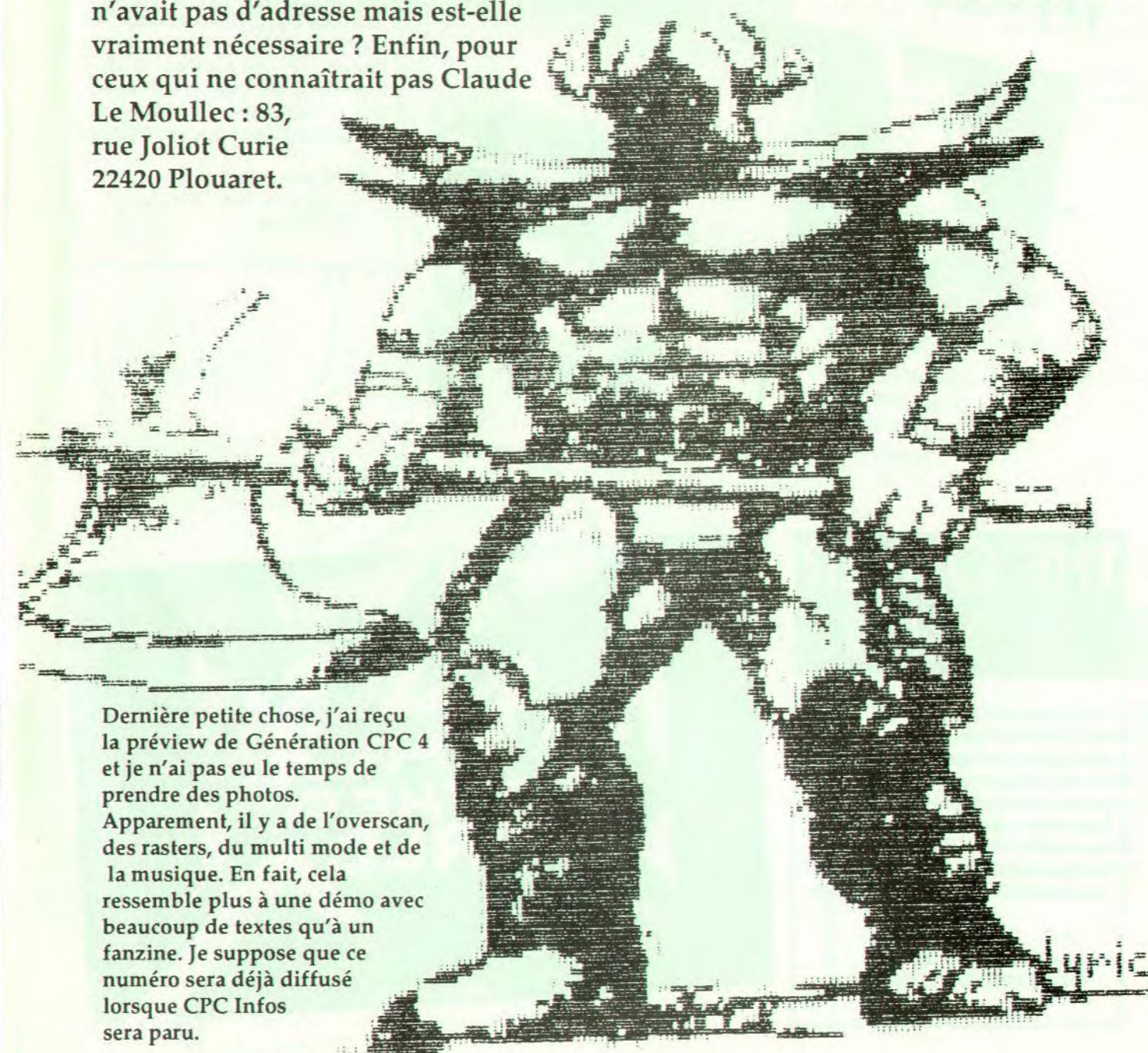
AMSTRAD est une marque déposée.
CPC infos est une revue mensuel totalement indépendante d'AMSTRAD GB et d'AMSTRAD FRANCE.

FANZ'

I N F O S

Finalement, j'ai reçu assez de fanzines pour alimenter la rubrique de ce mois ci. Alors continuez à envoyez vos fanzines à la rédaction sur disques ou sur papier, c'est très bien et en plus cela vous fait de la pub. Attention, vérifiez bien la qualité de vos disques avant de les envoyer, car certains sont illisibles : c'est le cas de Return for Escape.

Conclusion, il ne passe pas dans ce numéro. Le mois dernier, Ali Gator Dundee n'avait pas d'adresse mais est-elle vraiment nécessaire ? Enfin, pour ceux qui ne connaîtrait pas Claude Le Moullec : 83, rue Joliot Curie 22420 Plouaret.



Dernière petite chose, j'ai reçu la preview de Génération CPC 4 et je n'ai pas eu le temps de prendre des photos.

Apparement, il y a de l'overscan, des rasters, du multi mode et de la musique. En fait, cela ressemble plus à une démo avec beaucoup de textes qu'à un fanzine. Je suppose que ce numéro sera déjà diffusé lorsque CPC Infos sera paru.



ARKADIA II

Le numéro deux d'Arkadia est particulièrement intéressant : tout d'abord c'est le plus important en quantité puisqu'il occupe 4 faces de disquettes et ensuite, la réalisation est très bonne. Bien sûr, il n'y a pas d'écrans en overscan, de rasters, de scrollings différenciels sur 5 plans (on se calme !) mais des images numérisées bien placées, et une débauche de textes (2 disquettes quand même). C'est clair, c'est agréable à regarder et à écouter. Donc précipitez vous sur Arkadia :

Grégory Conn
1, rue A Le Bras
29200 Brest

THE OTHER WORLD

Le premier numéro de T.O.W. est présenté sur une seule face de disque, ce qui peut sembler un peu léger par rapport aux autres productions, mais ne vaut-il pas mieux privilégier une certaine simplicité pour mieux respecter des délais de parution. Enfin, le numéro 2 comprendra 2 faces.

I.F.C.
Les Pataudes Bisseuil
87220 FEYTIAT



SOS FANZINE

Effectivement, le numéro 4 est plus complet que le numéro 3 et la copie intégrale n'est plus nécessaire pour diffuser le fanzine. En revanche, je ne sais pas si mon exemplaire est buggé ou quoi, mais il y a des problèmes d'accès à certains fichiers (Read Fail). Enfin les rubriques habituelles sont là : test de démos, interviews, pokes, etc... Pour se procurer le fanzine : 1 timbre à 3.80 + 1 timbre à 2.30 + 1 disquette (pleine de préférence) + une enveloppe self-adressée.

Au fait SOS recherche un graphiste et un musicien alors, si vous êtes intéressé, n'hésitez pas :

SOS Fanzine (Moussa)
12, rue de Corse
93600 Aulnay-sous-bois

OU

SOS Fanzine (Steve)
5, rue J. Prévert
93430 Villetteuse



MICROSOFT

Ne cherchez plus ce titre dans la presse des fanzines, puisqu'il change dès le mois de Juin. Désormais vous trouverez Roostrad. Cela ne changera pas le nombre de pages qui est supérieur à 20. Le numéro 7 est prévu en Septembre et il y aura même un numéro hors-série spécial Help

Carole Duguy
Bel-égo
44850 St Mars du désert

ACTUALITÉ



CDS SOFTWARE

Vous qui aimez le foot, vous allez pouvoir vous défouler sur les écrans verts de vos CPC avec European Superleague. Mais attention, il n'est pas question de s'amuser à courir après une babille. Il faut plutôt s'occuper du côté financier de la chose et se mettre dans la peau du manager d'une grande équipe européenne. Le choix des joueurs, les transferts, l'entraînement, les tactiques et toutes les opérations financières sont à votre charge, donc bon courage.



EXPOS

Vous ne connaissez pas le «Supergames show» ? Ce n'est pas étonnant, puisqu'il s'agit de la première édition de cette exposition (organisée par notre confrère Micronews). Pour découvrir les merveilles des jeux électroniques, il suffira de se présenter le 6 Décembre (enfin peut-être pas le 6 : ce sera une journée professionnelle) porte de Champerret pour entrer dans les lieux dédiés à l'électronique ludique.

Normalement, ce sera l'occasion de rencontrer les éditeurs et tous les acteurs du secteur micro-informatique à une période de l'année où les achats des fêtes ne sont pas encore effectués.

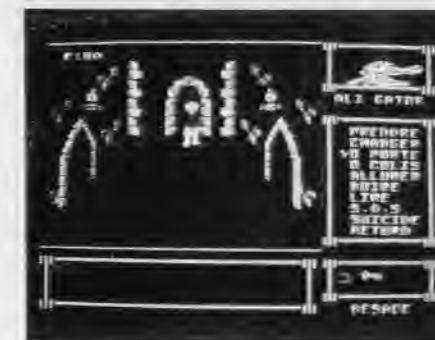
Dans un registre moins ludique, voici le salon Micro & Co. Sa vocation est de rassembler divers passionnés de micro. Ainsi, durant les 4 journées du salon, un thème différent sera abordé : Technique et bureau, arcade, l'espace créativité et la folie aux affaires. Ce salon est organisé avec le concours de la société Pressimage et aura lieu au Parc des expositions de Paris, Porte de Versailles, les 18, 19, 20 et 21 octobre 1991.

ACTUALITÉ



LANKHOR

Lankhor devient le spécialiste des jeux d'aventures sur CPC puisqu'après Alive et Le Fugitif, deux nouvelles histoires vont bientôt trouver place dans les douillets circuits de votre machine préférée. Le premier est intitulé : le trésor d'Ali Gator.



Les habitués des listings de CPC auront certainement reconnu ce nom et deviné alors l'auteur : il s'agit bien sûr de Claude Le Moullec. Dans un château mystérieux pétris de labyrinthes aux portes aléatoirement closes, vous devez retrouver un trésor en vous aidant de divers éléments, bouteilles, livres, clefs et autres objets.

Autre aventure bientôt disponible : Infernal House qui promet de vous donner des frissons dans une étrange demeure, à l'origine de nombreuses disparitions, dont celle de votre petite amie. Cela vous encouragera à mener vos recherches tambour battant parmi les diverses

TITUS

On retourne à l'ère préhistorique avec un personnage fruste et plein de poils (voir la couverture). La principale préoccupation de cette époque, et la mienne, est bien entendue la recherche de nourriture. Dans un décor luxuriant, le sympathique Grawagars, massue à la main, quête le moindre bout de graillon et bastonne accessoirement, mais d'une main décidée la faune pas câline. La version CPC de Préhistorik n'est pas encore prête mais elle promet beaucoup avec des graphismes colorés et une action «cartoonesque».

Puisque nous sommes dans le dessin animé, restons-y avec la bande à Plcsou, célèbre personnage Disneyen. La préoccupation actuelle de notre héros est de devenir le canard de l'année. Ce défi, lancé par son pire ennemi, Flairsou est à l'origine des aventures épiques de Plcsou et de ses neveux. Exploration de la jungle, safari photo et pilotage de biplan sont au programme.



plées et surtout les passages secrets que recèlent la maison.





DOMARK

Décidément, le foot revient en force avec Euro Football Champ de chez Taito. La compagnie japonaise a signé avec Domark pour l'adaptation du célèbre jeu d'arcade et vous devriez bientôt pouvoir vous exercer sur les pelouses électroniques.

La vitesse toujours et les joies nautiques en prime, tel est le menu alléchant de Hydra. Dans votre petit bateau, vous devez échapper à différents éléments hostiles. Pourquoi tant de haine ? Eh bien, voyez-vous, vous êtes une sorte de transporteur pour services secrets et un tas de

villains ne demandent pas mieux que de s'emparer de votre précieux chargement. Vous vous doutez donc que vos traversées seront loin d'être paisibles.



OCEAN

Autre adaptation de jeu d'arcade à succès : Narc. On se doute bien qu'avec ce titre, il ne s'agit pas de traiter de la cueillette des olives. Une fois de plus, les méfaits d'un caïd de la drogue vous forcent à sortir de votre tanière et à faire jouer vos arguments percutants : mitrailleuse lourde et muscles d'acier. Le tout possède des images digitalisées en présentation et une certaine dose d'action. Tous ces éléments seront indispensables pour venir à bout du chef de la pègre : Mr Big.



TRUCS POUR L'EXPLOITATION DE DISQUETTES A FORMATAGE SPECIAL EN LECTEUR A OU B

Valable pour CPC 464+DDI - 664 - 6128

Les possesseurs de CPC se sont vus de multiples fois proposer dans le commerce ou dans des revues des utilitaires de formatage pour augmenter la capacité des disquettes.

Le truc le plus simple est déjà d'utiliser les pistes 40 et 41. Un autre est d'avoir 10 secteurs par piste, ce qui donne pour 42 pistes (00-41) 210 KO de capacité. Le problème est que bien souvent, dès qu'il s'agit de faire tourner des programmes sur de telles disquettes, rien ne va plus. A coup sûr on obtient un «read fail», quand ce n'est pas un renvoi pour le lecteur A alors qu'on travaillait sur le B: «Drive A: disc missing». L'objet de cet article est d'aider à résoudre de tels problèmes.

Le lecteur attentif aura deviné la source de tous les maux : la réinitialisation du système disque. Celle-ci ne s'opère qu'en langage machine. Elle est facultative quand on travaille sous BASIC et indispensable quand on a lancé un programme en Langage Machine (qu'on désignera désormais par LM) avec RUN et qu'il doit accéder au disque.

QUAND ET POURQUOI REINITIALISER LE SYSTEME DISQUE ?

Quand on lance un programme en LM par RUN, le système d'exploitation du CPC fait un appel à une routine qui remet en quelque sorte l'ordinateur à zéro, si bien qu'il en «oublie» même qu'il pouvait y avoir un lecteur de disquette connecté. Pour qu'un lecteur puisse être à nouveau employé, il faut réinitialiser le système disque, c'est-à-dire en faire effectuer la reconnaissance de présence par le CPC. On parlera désormais pour système disque de DOS (de l'anglais Disk Operating System = «Système Opérateur de Disque», d'où le nom du DOS du CPC : AMSDOS pour Amstrad DOS), histoire d'utiliser un terme consacré. Un programme en LM chargé par un LOAD et exécuté par un CALL peut aussi nécessiter de réinstaller le DOS, car certains programmes procèdent à une remise à zéro du CPC. Le résultat est le même qu'avec un RUN. Un programme en BASIC peut avoir besoin de libérer ailleurs la partie de la mémoire vive réservée au DOS (un possesseur de CPC 464 avec lecteur un peu curieux aura remarqué que le HIMEM passe de 43903 sans lecteur à 42619 avec. Les 1284 octets manquants sont presque entièrement utilisés par le DOS). Dans ce cas il y a encore réinitialisation du DOS.

LES ROUTINES DU CPC EN CAUSE

On va ici examiner les routines du système d'exploitation du CPC, qui nous concernent, à savoir la remise à zéro du CPC et la réinitialisation du DOS. La terminologie employée est celle de la BIBLE du CPC, qui est celle standard. Ces routines ont appelées via des vecteurs de saut, à adresse fixe pour tous les CPCs, ce qui permet la compatibilité entre 464, 664 et 6128. C'est à l'adresse de ces vecteurs qu'il est fait référence dans la suite.

MC BOOT PROGRAM (&BD13) : remet à zéro le CPC. Cette routine est employée lors du lancement d'un

programme. Pour un programme en BASIC, la main est transmise à l'Interpréteur BASIC et le DOS n'est pas touché (sinon, même pour les formatages normaux, il faudrait réinitialiser le DOS après chaque chargement). Par contre pour un programme en LM, la main est transmise à la routine suivante, avec les conséquences que l'on va voir.

MC START PROGRAM (&BD16) : réinitialise le CPC et déconnecte toute périphérie (Attention : dans la terminologie propre au CPC, une initialisation est plus poussée qu'une remise à zéro. Celle-ci laisse certains paramètres intacts tandis que celle-là ramène comme à l'allumage de l'ordinateur). Il y a également saut à une adresse. Le fait important ici est que, la périphérie étant déconnectée et le lecteur de disquette en faisant partie, pour accéder de nouveau au disque, il faudra auparavant réinitialiser le DOS. Le fait gênant est que tous les vecteurs originaux sont restaurés (cf JUMP RESTORE).

KL INIT BACK (&BCCE) : Initialise une ROM (de l'anglais Read Only Memory = Mémoire Seulement Lisible, soit en français Mémoire Morte, que l'on peut trouver encore parfois dans la littérature sous l'abréviation MEM, opposée à MEV pour Mémoire Vive) de second plan. A l'origine le CPC n'a qu'une ROM de premier plan, la ROM BASIC. Les autres ROMs rajoutées par des extensions sont (en général) des ROMs de second plan, comme celle du lecteur de disquette. Les ROMS de second plan font partie de la périphérie. C'est à cette routine qu'il est le plus couramment fait appel pour réinitialiser le DOS.

KL ROM WALK (&BCCD) : trouve et initialise toutes les ROMs de second plan. A la différence de la précédente routine, celle-ci n'a pas besoin qu'on lui indique un numéro de ROM. Elle commence cependant sa recherche par la ROM du lecteur de disque. Comme très souvent elle est la seule ROM de second plan, le résultat est le même qu'avec un INIT BACK.

JUMP RESTORE (&BD37) : Initialise tous les vecteurs de sauts (dont font partie les 4 précédents), ce qui signifie que ce JUMP RESTORE restaure les valeurs originales des vecteurs de saut. Donc si on veut en détourner quelques-uns, et que l'on veut être à coup sûr prévenu contre ce JUMP RESTORE, il faut veiller à le détourner lui-même.

SYNTAXE DES ROUTINES

Il s'agit ici de la description d'utilisation des routines évoquées, quand c'est nécessaire, en langage d'assemblage Z80. A ceux qui n'y connaissent rien, cela restera abstrait, quant aux autres, cela fera une petite révision. Les nombres sont indiqués directement en hexadécimal, la seule base, avec le binaire, qui se respecte en LM.

MC BOOT PROGRAM : je ne l'ai jamais vu employé par quelqu'un d'autre que le système d'exploitation ou l'Interpréteur BASIC. Sa description est sans intérêt dans le cadre de cet article.

MC START PROGRAM: LD C,n

```
LD HL,dd
CALL BD16 (ou JP BD16)
```

- n est un code de configuration mémoire, qui détermine quelles RAMs ou ROMs sont sélectionnées. Presque toujours on utilise FF, ce qui fait que la totalité de la mémoire vive (que l'on désignera dorénavant par le terme anglais RAM - Random Access Memory = Mémoire à Accès Aléatoire ou Read And Modify Memory = Mémoire Lisible et Modifiable) est adressable de 0000 à FFFF.

- dd est l'adresse de saut après l'exécution de la routine.

```
KL INIT BACK: LD HL,dd1
              LD DE,dd2
              LD C,n
              CALL BCCE
```

- dd1 est le haut de la RAM disponible dont la ROM à installer peut disposer. Inversement dd2 est le bas de la RAM disponible.

- n est le numéro de la ROM. Pour le lecteur de disquette c'est 07.

Au retour de la routine, HL et DE contiennent les nouvelles valeurs pour la RAM disponible. En ce qui concerne le système disque, seuls les 1284 octets supérieurs sont utilisés, si bien que seul dd1 est modifié. En général, on indique 0040 pour dd2. On reviendra au chapitre suivant en détail sur les conséquences de l'appel de cette routine ou de la suivante.

```
KL ROM WALK: LD HL,dd1
              LD DE,dd2
              CALL BCCB
```

- dd1 et dd2 ont la même signification que pour KL INIT BACK et sont semblablement modifiés. Cette routine se contente en fait d'appeler la précédente, en ajoutant évidemment le numéro de la ROM. Il n'y a donc pas de surprise sur son fonctionnement dès qu'on a compris celui INIT BACK.

JUMP RESTORE : CALL BD37

- On ne peut pas rêver plus simple !

LA RAM AFFECTEE AU DOS

Le DOS a besoin d'un peu de place en RAM pour travailler. On ne va pas décrire ici l'ensemble de cette RAM dédiée au DOS mais seulement les parties nécessaires aux bidouillages pour exploiter des disquettes en formats spéciaux.

- Le DOS s'est vu attribuer par les concepteurs du logiciel une zone fixe en mémoire, qui est aussi la même sous CP/M (en fait l'AMSDOS a été conçu pour une compatibilité maximale avec le CP/M). Cette zone va de BE40 à BE7F. L'information qui nous intéresse ici est située en BE7D. Là est stockée l'adresse (donc sur 2 octets) de départ de la RAM dédiée au DOS (les 1284 octets dont on a déjà parlé). Pour abréger par la suite, on parlera de «RAMDOS», qu'on se le dise ! Pour résumer, en BE7D se trouve un pointeur sur le départ de la RAMDOS.

- La RAMDOS débute par 3 octets extrêmement intéressants, qui sont dans l'ordre :

- 1) le lecteur appelé (0 pour A et 1 pour B)
- 2) le user appelé
- 3) le lecteur actif

Lorsqu'une tentative de mise sur un lecteur a réussi, le lecteur actif est le même que l'appelé. Quand tout marche, c'est le cas et on fera donc comme si c'était le cas. Si on veut continuer à travailler avec un formatage spécial avec les lecteurs et users de son choix malgré une réinitialisation de DOS, il faudra veiller à rétablir ces 3 octets dans leur état avant l'appel de ROM WALK ou INIT BACK. En effet, en cas de réinitialisation normale du DOS, on se retrouve d'une part d'office en lecteur A et sous USER 0, et d'autre part, le DOS ne reconnaît plus que les formats habituels. Cette dernière constatation permet de faire la transition avec la suite.

- Pour l'exploitation d'un formatage quelconque de disquette, l'AMSDOS ne requiert que 25 octets seulement. Ces 25 octets, qui correspondent à 19 paramètres et que nous allons bientôt détailler, indiquent la capacité disc, la taille des blocs, celle du directory, le nombre d'entrées dans celui-ci, les numéros de secteurs, etc... Ils constituent le Bloc de Paramètres Disc, Disk Parameter Block en anglais, que l'on désignera par DPB. Comme l'AMSDOS peut travailler avec 2 lecteurs, il y a 2 DPBs, le DPB A et le DPB B. L'AMSDOS accède aux DPBs par l'intermédiaire de 2 pointeurs, qui indiquent donc les adresses des DPBs. Si la RAMDOS est située à l'adresse dd, alors on trouve le :

- pointeur de DPB A en dd+021A
- pointeur de DPB B en dd+022A

Lors d'une initialisation de DOS, les 2 DPBs sont remis aux paramètres standards du format System (recopiés depuis la ROM DISC). Pour exploiter une disquette en format spécial, il faut donc installer le DPB propre à ce format.

On a déjà tous les éléments en main pour imaginer une méthode d'exploitation de disquette en format spécial, malgré une réinitialisation de DOS. Rappelons d'abord les zones vitales de la RAM :

BE7D : pointeur sur le départ de RAMDOS ou son adresse, ad.

ad+0000 : lecteur appelé

ad+0001 : user appelé

ad+0002 : lecteur actif

ad+021A : pointeur sur le DPB de lecteur A

ad+022A : pointeur sur le DPB de lecteur B

La méthode choisie et qui sera mise en œuvre de diverses manières dans les programmes de démonstration, surtout selon les capacités de programmation de chacun, sera la suivante :

- 1) déterminer l'adresse de la RAMDOS
- 2) stocker le lecteur et le user courants
- 3) en fonction du lecteur actif, déterminer l'adresse de son DPB
- 4) stocker le DPB du lecteur actif
- 5) réinitialiser le DOS
- 6) déterminer la nouvelle adresse de la RAMDOS
- 7) restaurer à cette adresse les lecteur et user actifs
- 8) en fonction du lecteur actif, trouver le pointeur de son DPB
- 9) remplacer ce pointeur par l'adresse de stockage du DPB original

Il est plus rapide de modifier simplement un pointeur de DPB que de reloger un DPB. Cela réduit le nombre d'opérations. Autant s'épargner du travail inutile. On a maintenant les connaissances nécessaires et suffisantes à la réalisation de notre entreprise, ainsi que la méthode à employer. Avant de se lancer dans la description plus ou moins approfondie des différents programmes qui mettent tout ceci en application, on va s'offrir une petite récréation avec la description détaillée d'un DPB, ce qui ouvrira peut-être à certains de nouveaux horizons (pour concevoir des formatages personnalisés par exemple).

DESCRIPTION D'UN DPB

On a déjà évoqué la compatibilité maximale entre l'AMSDOS et le CP/M. C'est ainsi que les DPBs sont directement issus du monde du CP/M. Les DPBs de l'AMSDOS ne sont rien d'autre que des DPBs CP/M, pour les 15 premiers octets, qui correspondent à 9 paramètres. On peut donc rechercher leur description dans un livre dédié au CP/M ; c'est valable pour le CPC comme pour n'importe quel ordinateur dont le système d'exploitation est le CP/M. Leur fonction est purement logique et décrit l'organisation des données sur la disquette. Complémentairement, les 10 octets suivants, spécifiques des CPCs, décrivent la structure physique de la disquette : ils font partie des paramètres FDC (Floppy Disc Controller = Contrôleur de Disque Souple, c'est-à-dire disquette).

Voyons d'abord un tableau de ce DPB, avec la terminologie en vigueur :

adresse	NOM	taille	description très sommaire
0000	SPT	0002	nb de secteurs logiques par piste
0002	BSH	0001	décalage de bloc
0003	BLM	0001	masque de bloc
0004	EXM	0001	masque d'incrément
0005	DSM	0002	nb maxi de blocs
0007	DRM	0002	nb maxi d'entrées au catalogue
0009	AL0, AL1	0002	taille du catalogue en binaire
0008	CKS	0002	nb d'entrées catalogue à vérifier
000D	OFF	0002	décalage de piste
000F	FSC	0001	numéro physique du 1er secteur
0010	PST	0001	nb de secteurs par pistes
0011	GLS	0001	GAP 3 pour lecture/écriture
0012	GLT	0001	GAP 3 pour formatage
0013	FLB	0001	octet de remplissage pour formatage
0014	BPS	0001	taille de secteur codée
0015	RPS	0001	nb d'enregistrements par secteur
0016	BCT	0001	buffer piste actuelle (DOS)
0017	FT0	0001	Indicateur d'alignement (DOS)
0018	FLG	0001	Indicateur de sélection de format

Voyons maintenant plus en détail la signification de ces valeurs :

- SPT (Sectors Per Track = secteurs par piste) : c'est le nombre de secteurs logiques de 128 octets contenus sur une piste. Pourquoi 128 octets ? Pour des raisons historiques ! A l'origine le CP/M travaillait avec des secteurs physiques de 128 octets. Les modèles ultérieurs de lecteurs de disquette permettant d'utiliser des secteurs d'une autre taille, on a cependant con-

servé ce découpage virtuel en secteurs de 128 octets (qu'on appelle encore parfois enregistrements) d'une piste pour des raisons compréhensibles de compatibilité. Ceci explique la dénomination «secteur logique», par opposition à physique. Ainsi en format Data, on a 9 secteurs de 512 octets par piste, ce qui donne un SPT de $36 = 9 \cdot 512 / 128$. Après cette description des valeurs, on en donnera des tableaux pour les formats standards et d'autres plus exotiques. Les curieux de la disquette, qui se sont un jour intéressés à la structure du catalogue (ou directory en anglais, ou répertoire dans une traduction fidèle), ont pu remarquer en 16e position d'une entrée une valeur plafonnant à &80. C'est la taille du fichier de l'entrée en blocs de 128 octets. En standard une entrée correspond à 16 blocs de 1 KO au maximum. D'où on retrouve bien : $16 \cdot 1024 / 128 = 128 = &80$. Ceci est très important, car même en travaillant avec de blocs de 4 KO, on peut connaître la taille d'un fichier à moins de 128 octets près. Mathématiquement on a : $SPT = PST \cdot 2^RPS$

- BSH (Block Shift = décalage de bloc) : c'est (attention il faut être à première vue matheux) le logarithme en base 2 du nombre de secteurs logiques, donc de 128 octets, par bloc. Ceci sera beaucoup plus clair dans quelques instants. Le bloc, dont il a été question pour le SPT, est la taille minimale occupée par un fichier sur la disquette. Sous AMSDOS c'est 1 KO. Même pour sauver un seul octet, 2 secteurs de 512 octets sont employés. Soit donc TB la taille en octets d'un bloc. On a alors : $TB = 128 \cdot 2^BSH$. En format standard on a : $1024 = 128 \cdot 2^3$, donc le BSH vaut 3. Mathématiquement on a : $BSH = \log(TB/128)/\log(2)$

- BLM (Block Mask = masque de bloc) : c'est le nombre de secteurs logiques par bloc, moins 1. En standard, on a des blocs de 1 KO, soit 8 secteurs logiques, d'où le BLM est $7 = 8 - 1$. Mathématiquement on a : $BLM = 2^BSH - 1$

- EXM (Extend Mask = masque supplémentaire) : c'est une extension au masque fonction de la taille d'un bloc et du nombre de blocs sur la disquette. C'est égal au nombre maximum d'incrément sur un FCB moins un. Explications : un FCB est un File Control Block, c'est-à-dire un Bloc de Contrôle de Fichier. Il s'agit d'une entrée au catalogue, déjà évoquée pour le SPT. Une entrée, ou donc un FCB, est composée de 32 octets, les 16 premiers pour le user, le nom, un numéro d'extension pour les fichiers nécessitant plus d'une entrée, et la taille en blocs de 128 octets. Les 16 restant sont les numéros des blocs du disque occupés par le fichier. L'incrément est la taille maximale d'un fichier tenant sur un FCB, dans le cas le plus défavorable de formatage. Or le CP/M, ou l'AMSDOS, peut travailler avec des secteurs de 128 octets. Le 16e octet, qui donne la taille du fichier, nous limite donc à 16 KO, 32 KO n'étant pas exprimable sur un octet : $32 \cdot 1024 / 128 = 256 = &100$. L'incrément vaut donc 16 KO (N.B. : ceci n'est vrai en fait que pour les «petits» disques, pour lesquels on ne peut gérer que moins de 256 blocs. Au-delà on parle de «grands» disques et l'incrément est de 32 KO). En standard on a 16 KO par entrée au maximum, soit 1 incrément, donc le masque d'extension vaut $0 = 1 - 1$. Mathématiquement on a : $EXM = nb \text{ max} \text{ incrément} \text{ par FCB} - 1$.

- DSM (Disk Storage Mask = masque de stockage du disque) : C'est le numéro du dernier bloc du disque,

ou, la numérotation commençant à 0, la capacité de la disquette moins 1. En format SYSTEM, cette valeur est $170 = &AA$. Le bloc 0 est le premier bloc réservé au répertoire. Tous les blocs potentiellement en amont de ce bloc sont ignorés, comme pour les 2 pistes dites «système», réservées au CP/M. Voir à ce propos la description de OFF. Donc DSM est le numéro du dernier bloc ou la capacité totale de la disquette, directory y compris, moins 1.

- DRM (Disk Repertory Mask = masque de répertoire du disque) : c'est le nombre d'entrées au catalogue, moins 1, ou le numéro de la dernière entrée, en débutant à 0. On a déjà vu qu'une entrée est longue de 32 octets. Dans les formats standards on a 2 KO de directory, d'où un nombre d'entrées de $64 = 2^{10} / 32$. On obtient donc 63.

- AL0, AL1 (Allocation 0, Allocation 1) : ces deux valeurs expriment les blocs occupés par le catalogue. Il faut exprimer AL0 et AL1, dans cet ordre, en binaire sur 8 bits : 1 bit correspond alors à 1 bloc et la lecture se fait naturellement de gauche à droite ; le bit 7, soit de plus fort poids ou MSB en anglais (Most Significant Bit = Bit le Plus Significatif) de AL0 correspond au premier bloc et le bit 0, soit de plus faible poids ou LSB en anglais (Less Significant Bit = Bit le Moins Significatif), de AL1 au dernier bloc. Un bit mis (à 1) signifie que le bloc correspondant appartient au directory. En format standard on a 2 blocs de réservés pour le directory. Ecrivons le en binaire sur 8 bits : 1100 0000, 0000 0000 soit en hexadécimal C0,00.

AL0 , AL1

La taille TC du catalogue, exprimée évidemment en blocs, se déduit du nombre d'entrées et de la taille d'un bloc. En standard on a 64 entrées et des blocs de 1 KO, d'où $32 \cdot 64 / 1024 = 2$ KO de répertoire. Mathématiquement on a : $TC = (DRM + 1) / 4 \cdot 2^BSH$

- CKS (CHeked Sectors = secteurs vérifiés) : c'est le nombre de secteurs logiques du répertoire à vérifier avant toute opération d'écriture. Ainsi avant d'écrire un nouveau fichier, il faut vérifier qu'il reste assez de place sur le disque et qu'il n'existe pas de fichier portant déjà le même nom, auquel cas il faut se livrer à des opérations supplémentaires. En standard on a 64 entrées (de 32 octets), donc CKS vaut $16 = 64 \cdot 32 / 128$. Mathématiquement on a : $CKS = (DRM + 1) / 4$

- OFF (Offset = décalage) : il s'agit simplement du nombre de pistes réservées au système, ou bien à ignorer avant que ne débute la partie de la disquette exploitée par le DOS pour le stockage de données. En format System OFF vaut 2, les 2 premières pistes étant employées par le CP/M. On peut aussi utiliser ce paramètre pour diviser une disquette «physique» en plusieurs disquettes «logiques». Par exemple il existe des lecteurs 80 pistes, qui avec un formatage de 10 secteurs de 512 octets offrent par face physique 400 KO de capacité de stockage. Pour exploiter un tel volume on recourt à des blocs de 2 KO. Or l'AMSDOS ne gère pas correctement de tels blocs (alors que le CP/M sait le faire). Il en résulte que beaucoup de chargements aboutissent à des plantages. Il faut donc revenir à des blocs de 1 KO. Seulement le DOS ne peut travailler qu'avec au plus 255 blocs. L'astuce est de diviser une face physique de 400 KO en 2 faces logiques

de 200 KO. Pour la 1ère le décalage OFF vaut 00 tandis que pour la seconde il vaut 40. On commute aisément d'une face à l'autre en mettant OFF à la valeur convenable, ce par un simple POKE sous BASIC. Ce décalage est également très utile pour une disquette à format spécial, totalement incompatible avec les formats standards. Si toutes les pistes sont dans ce format, il faut lancer un utilitaire d'implantation du DBP depuis une autre disquette, ce qui est contraignant. L'astuce est de formater la première piste en Data et les autres comme on le veut. Sur les 2 KO disponibles sur cette première piste, on a largement la place pour installer un utilitaire d'implantation de DPB. Le décalage pour le reste des pistes en format spécial est de 1. Il en sera donné un exemple dans les tableaux de DPBs à la fin de ce chapitre.

Jusqu'ici les paramètres relevaient du monde du CP/M et les explications étaient valables pour toute machine CP/M. Désormais on a affaire à des paramètres du FDC, spécifiques au CPC.

- FSC (First SeCtor = premier secteur) : c'est le numéro physique du premier secteur de chaque piste. On use du qualificatif «physique» car il s'agit du numéro effectivement écrit sur la disquette pour identifier le premier secteur. En System c'est &41 et en Data &C1.

- PST (Physical Sectors per Track = secteurs physiques par piste) : c'est le nombre de secteurs physiques par piste, soit 9 en System ou Data.

- GLS (Gap Length for Sector = longueur de Gap pour secteur) : c'est la longueur du Gap pour les opérations de lecture et écriture. Qu'est-ce qu'un Gap sur une disquette ? C'est tout simplement un blanc (gap veut dire brèche en anglais). Il ne faut pas oublier que les lecteurs de disquettes sont avant toutes choses des machines mécaniques, donc soumises à des tolérances. Ainsi, à partir d'une certaine déclinaison, il n'y a pas deux lecteurs de disquette qui aient la même vitesse de rotation de disque. Un même lecteur peut voir sa vitesse varier en fonction des conditions ambiantes, sinon tout simplement de l'âge. Les Gaps sont là pour que ces variations, quoi qu'il en soit relativement infimes en cas de bon fonctionnement, ne créent aucun problème. Ils établissent des zones de sécurité, espèces de «no man's lands» entre les zones de données, en séparant les secteurs et les pistes. Le GLS est le Gap pris en compte en lecture ou écriture de secteur. Il est aussi appelé GAP 3 dans la littérature spécialisée. Il est situé après les secteurs. Pour les formats standards il vaut &2A.

- GLT (Gap Length for Track = longueur de Gap pour piste) : c'est la longueur du Gap employée lors du formatage d'une piste. Ce Gap est le même physiquement que le précédent. Seulement cette valeur n'est utilisée que lors du formatage d'une piste. Normalement le GLS doit être plus petit que ce GLT, afin que les nouveaux Gaps 3 dûs aux opérations d'écriture (de longueur GLS), ne «débordent» pas au-delà du Gap 3 résultant du formatage. On a d'autant moins de risques d'avoir des problèmes de recouvrement de données par un Gap 3, que la différence entre GLT et GLS est grande. Pour les formats standards le GLT vaut &52. Cela représente presque le double de GLS ; la marge de sécurité est excellente.

- FLB (FILL Byte = octet de remplissage) : c'est l'octet utilisé pour remplir les secteurs lors d'un formatage de piste. Pour les formats standards, il s'agit de &E5.

- BPS (Bytes Per Sectors = octets par secteurs) : c'est de façon «codée», afin que l'information tienne sur un octet, la longueur en octets d'un secteur. Pour connaître la longueur d'un secteur, il suffit d'employer cette formule, l étant la longueur : $l = 128 \cdot 2^BPS$. C'est cette valeur qui est en fait directement envoyée au FDC lors du formatage de piste pour indiquer la longueur d'un secteur. Pour les formats standards on a 512 octets par secteurs, or $512 = 128 \cdot 2^2$, donc BPS vaut 2.

- RPS (Records Per Sector = enregistrements par secteur) : c'est le nombre de secteurs logiques par secteur physique ; plus prosaïquement c'est la taille d'un secteur en octets divisée par 128. Mathématiquement on a : $RPS = 2^BPS$.

- BCT (Buffer for Current Track = mémoire pour piste courante) : c'est une sorte de mémoire pour stocker le numéro de la piste actuellement en service. C'est une des 2 valeurs variables du DPB sans intervention de l'utilisateur. Elle est utilisée exclusivement par le DOS et donc n'a que le mérite d'être signalée.

- FT0 (Flag for Track 0 = Indicateur pour piste 0) : c'est un indicateur pour faire savoir au DOS où se trouve la tête de lecture. Le FDC possède une fonction qui lui permet d'aller automatiquement à la recherche de la piste 0. Il interrompt sa recherche au bout d'un certain temps, et indique si elle a ou non atteint la piste 0. La piste 0 a été atteinte si FT0 vaut &FF, sinon il vaut 0. C'est la deuxième valeur variable du DPB du fait du FDC et comme BCT elle ne présente d'intérêt que pour le DOS.

- FLG (Flag for LoGIn = Indicateur d'accès) : c'est un indicateur destiné au DOS, pour lui faire savoir si, avant toute opération de lecture ou écriture, il doit accéder au disque afin d'en reconnaître le format. On parle couramment pour désigner cette opération de login. Si FLG vaut 0, il y a automatiquement login, s'il vaut &FF, il n'y a pas de login. Naturellement le DOS ne reconnaît que les formats standards, à savoir System, Data et IBM, qui sont implantés dans la ROM Disc. FLG est un paramètre important car si le login est autorisé, alors qu'on est sous un format spécial, il y a recherche du format de disquette. Cela débouche sur un «read fail», ou, quand le formatage peut tromper le DOS, comme en rajoutant un 10ème secteur, à un plantage. Il faut donc mettre cet indicateur à &FF pour éviter une telle déconvenue (en fait le problème du login ne se pose apparemment que si le DPB est situé dans la RAMDOS ; c'est une bizarrerie de l'AMSDOS).

Voilà ! On a enfin passé en revue le contenu d'un DPB, qui ne doit plus maintenant avoir de secret pour personne. On va voir maintenant quelques valeurs de paramètres pour ceux à qui cette récréation informatique aurait donné envie d'expérimenter des formats exotiques.

Commençons par un tableau de BSH, BLM et EXM, qui sont certainement les paramètres les plus énigmatiques de l'ensemble. Ils sont donnés en fonction de la taille du bloc. On rappelle qu'on parle de «petit disque» pour moins de 256 blocs et de «grand disque» pour plus de 255 blocs.

Taille de bloc en KO	BSH	BLM	EXM		
			petit disk	grand disk	
1	3	7	0	-	
2	4	15	1	0	
4	5	31	3	1	
8	6	63	7	3	
16	7	127	15	7	

Finissons par un tableau de DPBs entiers. On va y retrouver pour la forme les formats System et Data. J'ai appelé Data+ le format déduit du Data, avec 10 secteurs par piste (on a rajouté un secteur &CA), et en utilisant les pistes 40 et 41, ce qui offre 208 KO de stockage de données. Comme les 9 premiers secteurs sont identiques à ceux en Data, un programme situé en piste 0, qui installera le DPB du Data+, peut être chargé dès l'allumage du système sans aucune modification de DPB. Le format 202K est un format débutant en piste 1, la piste 0 étant en Data, afin d'y mettre l'utilitaire installant le DPB pour le 202K. D1-198K et D2-198K sont les 2 DPBs pour partager une face de disquette 80 pistes en deux. Ils ne se distinguent que par la valeur de OFF (0 pour l'un et 40 pour l'autre). La disquette est formatée comme en Data+. Toutes les valeurs sont en hexadécimal.

	System	Data	Data+	202K	D1-198K	D2-198K
SPT	24	24	28	28	28	28
BSH	03	03	03	03	03	03
BLM	07	07	07	07	07	07
EXM	00	00	00	00	00	00
DSM	AA	B3	D1	C9	C7	C7
DRM	3F	3F	3F	3F	3F	3F
AL0, AL1	C000	C000	C000	C000	C000	C000
CKS	0010	0010	0010	0010	0010	0010
OFF	02	00	00	01	00	28
FSC	41	C1	C1	C1	C1	C1
PST	09	09	0A	0A	0A	0A
GLS	2A	2A	1C	1C	1C	1C
GLT	52	52	2A	2A	2A	2A
FLB	E5	E5	E5	E5	E5	E5
BPS	02	02	02	02	02	02
RPS	04	04	04	04	04	04
BCT	-	-	-	-	-	-
FT0	-	-	-	-	-	-
FLG	00	00	FF	FF	FF	FF

Voilà qui conclut définitivement cette longue dégression et, je l'espère, récréation sur les DPBs. On va enfin aborder les programmes pour utiliser sans (trop de) problèmes les formats spéciaux.

PROGRAMMES POUR EXPLOITER DES FORMATS SPECIAUX

Il va s'agir ici de commenter les différents programmes proposés, bien que leurs listings soient déjà largement. Mais mieux vaut deux fois qu'une. Des explications parfois importantes sont données ici, qui ne figurent pas dans les commentaires des program-

mes. Lorsqu'il est fait explicitement appel à un format spécial, on a choisi le Data+, qui est le plus intéressant pour tout possesseur de système disque.

1) BOOT-DPB : est souvent appelé BOOT un petit programme qui installe un système d'exploitation ou qui lance un programme en faisant un peu de ménage avant, pour lui préparer la place. C'est la fonction de la routine du système d'exploitation MC BOOT PROGRAM.

Ce programme se contente d'installer un DPB et a surtout un intérêt pédagogique. Le DPB est mis en &BE00. C'est en effet une des rares zones de la mémoire qui soient inemployées tant par le système d'exploitation que par le BASIC, et par même beaucoup de programmes. Un rappel de l'emploi de la mémoire peut être utile, pour que chacun puisse loger un DPB. les RSXs (Resident System eXtension = extension au système résident) créées par le programme qui suivra, ou d'autres routines sensibles, à des endroits relativement sûrs :

0000I
saut dans le système d'exploitation pour remettre à zéro le CPC.
0007 Peut être employé sans risque.

X 0008
X série de restarts et de sauts ; strictement réservé
X 0040 au système d'exploitation. ZONE SENSIBLE

0041
RAM disponible sous BASIC
A6FB:

X A6FC
X RAM réservée à l'AMSDOS (RAMDOS)
X ABFF I ZONE SENSIBLE

X AC00I
X RAM réservée à la ROM BASIC. ZONE SENSIBLE
X BOFF I

X B100I
X RAM réservée au système d'exploitation. ZONE SENSIBLE
X B8FF I

X B900I
X ensemble de routines et de vecteurs de saut, ZONE SENSIBLE
X BDF3I

BDF4I
ESPACE LIBRE
BE3FI

X BE40I
X strictement réservé au DOS pour la gestion physique du lecteur de disquette. ZONE SENSIBLE
BE80I
ESPACE LIBRE, mais attention à la pile descendante du Z80 débutant en C000
BFFF

X C000I
X RAM VIDEO
X FFFF I

Il s'agit de la configuration sous BASIC. En LM, on libère la RAM réservée à la gestion du BASIC, on peut placer ailleurs la RAMDOS, ainsi que la RAM VIDEO... On constate donc que les zones libres et protégées sont rares : de BDF5 à BE3F et de BE40 à BF60 (la pile du Z80 s'étend très rarement en dessous de BF60).

Revenons sur le programme BOOT-DPB : l'adresse de placement du DPB peut être aisément modifiée. Il opère ainsi :

- 1) Installation du DPB
- 2) Lecture du pointeur de la RAMDOS
- 3) Lecture du lecteur actif (A ou B)
- 4) Remplacement du pointeur de DPB du lecteur actif par l'adresse d'implantation du nouveau DPB

On rappelle que, le format utilisé étant le Data+, ce programme peut se trouver sur la piste 0. A la mise en route, le DOS identifie le format comme du Data et BOOT-DPB peut être chargé. Ce dernier «réssuscite» alors en quelque le 10ème secteur, en installant le Data+. On se rend compte ici du danger à manipuler sans précautions des disquettes en Data et en Data+. Le DOS peut travailler avec une disquette en Data+ comme si elle était en Data, ce qui peut amener une altération de données en cas de sauvegarde. Prudence ! L'installation systématique sur toute disquette d'un premier fichier installant le DPB (et pourquoi pas faisant office aussi de menu - cf le programme MENU) est le meilleur moyen de se garder de tels désagréments.

2) LOADRUN : ce programme implante 2 RSXs pour charger des programmes en binaire. Ces RSXs sont I LOAD et I RUN et remplissent les mêmes fonctions que les commandes du même nom du BASIC. Elles sont loageables n'importe où dans la mémoire disponible. LOADRUN demandant une adresse de départ (une sauvegarde est effectuée automatiquement à la fin). Il est préférable d'indiquer la valeur en hexadécimal. Voyons d'abord les particularités de ces RSXs, sans quoi il n'y aurait pas de raison de remplacer les commandes originales :

- I LOAD : à la différence de la commande BASIC qui le nécessite, il n'y a pas besoin qu'il y ait 4 KO de mémoire disponible pour un buffer de transfert de données. La syntaxe est identique à celle en BASIC :

- I LOAD,«XXXXXX» pour charger à son adresse de chargement (I) le fichier XXXXXX. Les possesseurs de 464, à cause d'un bug de la ROM BASIC, doivent naturellement mettre le nom dans une variable et indiquer le pointeur de cette variable. Cela donne par exemple: a\$=>XXXXXX:I LOAD, aa\$. Ce sera cette dernière forme qui sera employée lorsqu'on aura affaire à cette instruction, ou à l'autre, pour des raisons évidentes de compatibilité. Si on oublie d'indiquer le nom de fichier, il y a retour immédiat au BASIC, de même si le nom est illégal, auquel cas on a droit au message d'erreur habituel de l'AMSDOS.

- I LOAD,«XXXXXX»:adr pour charger le fichier XXXXXX à l'adresse adr. La commande gère les erreurs qui peuvent survenir dans les routines de chargement: fichier absent ou bloc altéré. La main est rendue au

BASIC sans aucun problème. Le seul risque de planage est un chargement dans une des zones sensibles, signalées dans le plan de la mémoire, auxquelles il faut ajouter le programme BASIC lui-même ainsi que les 2 RSXs.

- I RUN : cette instruction fonctionne comme celle du BASIC. Il en résulte qu'elle ne peut être employée qu'une seule fois, la main étant passée à la fin de son exécution au programme qui a été chargé. En cas de problème de chargement (pas de nom fourni, nom illégal, fichier introuvable ou bloc altéré) il y a cependant retour au BASIC. La syntaxe est bien naturellement: I RUN,«XXXXXX» ou a\$=>XXXXXX:I RUN,aa\$. La différence majeure entre cette RSX et le RUN du BASIC est que cette routine ne fait pas appel à la routine du système d'exploitation MC BOOT PROGRAM, qui passe elle-même la main au MC START PROGRAM, avec les conséquences que l'on a déjà vu. Ainsi, grâce à cette RSX, il n'y a pas de JUMP RESTORE, c'est-à-dire que les vecteurs éventuellement détournés sont laissés intacts, et le système disque est toujours reconnu. C'est l'absence de JUMP RESTORE qui est toutefois important.

On dispose donc maintenant de 2 RSXs très utiles, tant pour l'objet de cet article que pour un usage personnel : avec I LOAD on peut se passer de buffer de 4 KO pour des chargements simples et avec I RUN on peut lancer des programmes binaires sans que le système disque soit «oublié» et sans que les vecteurs soient réinitialisés. On va donc pouvoir en détourner quelques-uns à nos fins. Ce sera mis en application dans le programme MENU. Mais auparavant on va enfin mettre en application la méthode décrite dans le chapitre LA RAM AFFECTEE AU DOS pour procéder à une réinitialisation de DOS sans perte des lecteur, user et DBB.

3) INITDOS1 : ce programme est encore un chargeur de routine, qui fonctionne comme le précédent. Il est destiné à être utilisé sous BASIC et opère un détournement de INIT BACK, ainsi d'ailleurs que de ROM WALK. La routine implantée est sauvegardée sous forme binaire en fin de programme. En queue de routine se trouve le véritable vecteur pour INIT BACK. Il est utile de rappeler que ce vecteur est spécifique à chaque type de CPC (464, 664 ou 6128). On ne pourra pas servir donc directement de la routine en binaire écrite



Voir bon de commande page 54

sur un type de CPC avec un autre type. Il faut la sauver sur ce dernier type. Un listing en langage d'assemblage suffisamment commenté est fourni en annexe, pour les curieux. Ceux qui n'y connaissent rien pourront cependant suivre les étapes.

Le chargeur fonctionne ainsi :

- demande de l'adresse d'implantation de la routine
- écriture de la routine
- écriture en fin de routine du vecteur original INIT BACK
- remplacement du vecteur INIT BACK par un saut à la routine
- remplacement du vecteur ROM WALK par un saut à la routine
- sauvegarde de la routine

On constate que pour rendre opérante cette routine, il faut détournner les vecteurs INIT BACK et ROM WALK. Cette manœuvre de détournement sera évidemment à répéter ultérieurement après tout chargement de la routine binaire. Quoi qu'il en soit, le chargeur est très simple et peut être aisément modifié selon sa convenance personnelle. Ce détournement de vecteur sera mis en œuvre très bientôt dans le programme MENU. Ajouté aux RSXs ILOAD et IRUN on obtiendra d'excellents résultats. Mais voyons auparavant une routine quasiment identique, à la différence près qu'elle ne requiert pas de détournement de vecteur : son usage reste en contrepartie réservé aux connaisseurs de l'assemblleur ou autres trifouilleurs d'octets.

4) INITDOS2 : encore une fois un chargeur BASIC, identique aux précédents, pour une routine à utiliser exclusivement sous assemblleur pour des programmes en LM. Comme indiqué en fin de paragraphe précédent, cette routine est uniquement destinée aux connaisseurs de langage machine. L'adresse d'implantation demandée par le chargeur BASIC n'a d'intérêt que pour la sauvegarde de la routine en binaire. Comme il apparaît à la lecture du listing en langage d'assemblage, elle est logeable telle quelle à n'importe quelle adresse, puisqu'elle n'effectue de saut à aucune adresse absolue, ni ne se modifie elle-même, comme le fait la routine de INITDOS1, pour que soient injectées dans HL et DE les valeurs désirées pour l'appel de INIT BACK. Il s'agit en fait de la routine proposée dans INITDOS1, mais un peu simplifiée. Le DPB est placé en BE00, qui est une zone très rarement utilisée. Les 3 octets précédents sont utilisés comme mémoire temporaire de lecteur et user. La routine a été ainsi conçue que l'adresse de ce buffer (mémoire tampon+DPB) n'est à indiquer qu'une fois, au début. On peut donc très rapidement le placer ailleurs.

Cette routine est à utiliser en sus et place du simple appel à INIT BACK. Si on dispose d'assez de place, soit 63 octets sans "RET", on n'a qu'à remplacer le INIT BACK par elle. Sinon il faut remplacer l'appel original à INIT BACK par un appel à la routine. Il faut veiller alors à injecter les valeurs correctes dans HL et DE avant l'appel du INIT BACK à l'intérieur de la routine. L'extraordinaire intérêt de la routine proposée est qu'on peut utiliser alors les programmes binaires qui en sont pourvus avec des disquettes de tous formats, standards ou non, et qu'en plus du INIT BACK les

lecteur et user actifs sont préservés. Il n'y a nul besoin de détourner un seul vecteur ou d'utiliser la RSX IRUN. On se sert du RUNXXXXXX du BASIC. Pour résumer, et en conclusion de ce paragraphe, grâce à cette routine, le lecteur, le user et le DPB du lecteur actifs avant la réinitialisation du système disque sont restaurés après.

5) MENU : enfin le programme attendu au moins par ceux qui ne jurent que par le BASIC, faute de mieux. Il met en application toutes les explications données : installation de DPB, RSXs, écriture d'une routine de remplacement pour INIT BACK, d'ailleurs détourné, et finalement modification de quelques vecteurs pour que le détournement effectué ne soit pas réduit à néant. Ce programme n'est pas exactement un menu, mais il prépare le terrain pour un véritable menu, comme il en sera donné un exemple avec MENU2, lancé par MENU. Je le redis encore une fois, mais mieux vaut trop que pas assez : MENU doit être le premier fichier sur la disquette, au sens physique du terme, c'est-à-dire résider en piste 0. On a 4.5 KO par piste en format Data. Les 2 premiers servant au directory, il en reste 2 pour un programme lisible indifféremment en Data ou Data+. MENU est très succinctement commenté pour rester dans la limite des 2 KO (il y a cependant encore de la marge pour des ajouts personnels). Hormis des initialisations de blocs et la modification de 2 vecteurs, tout a déjà été vu dans les programmes BOOT-DPB, LOADRUN et INITDOS1. Voyons de plus près le fonctionnement de MENU :

- Initialisations de blocs : c'est facultatif si on vient d'allumer l'ordinateur, sinon cela effectue une partie du travail du MC START PROGRAM, que l'on désactivera. On commence par restaurer les vecteurs originaux avec JUMP RESTORE (BD37). Cela verrouille sur le lecteur de cassette, ce qui explique ensuite l'instruction IDISC pour retourner au lecteur de disquette. On initialise les blocs :

- son : BCA7 - SOUND RESET
- gestion d'écran : BBFF - SCR INITIALISE
- écran graphique : BBBB - GRA INITIALISE
- écran de texte : BB4E - TXT INITIALISE
- clavier : BB00 - KM INITIALISE

- Installation du DPB : absolument rien de nouveau ici par rapport à ce qui est fait dans BOOT-DPB. On a seulement supprimé les commentaires et fait en sorte que cela prenne le moins de place possible. Le souci d'économie de place a été constant dans l'écriture de MENU, pour que l'utilisateur ait le maximum de liberté pour des apports dans la limite des 2 KO.

- Installation des RSX ILOAD et IRUN : se reporter à LOADRUN pour plus de détails. A noter l'utilisation d'un sous-programme commun à cette phase et à la suivante, toujours pour un gain d'espace. Comme on peut le constater à la lecture du listing, on peut très facilement loger à l'adresse que l'on désire les RSXs. Il n'y a qu'une variable dont il faut modifier la valeur, qui est celle d'implantation des RSXs.

- Ecriture du INITBACK détourné : se reporter à INITDOS1 pour plus de détails. Les remarques faites pour installer ILOAD et IRSX sont valables ici. On installe la routine remplaçant le INITBACK, dont on détourne le vecteur vers elle, ainsi que celui de ROM WALK.

- Modification de 2 vecteurs : il s'agit de MC START PROGRAM et de JUMP RESTORE. La nécessité de détourner ces routines a été expliquée dans le chapitre LES ROUTINES DU CPC EN CAUSE. Pour résumer, leur appel réinitialise les vecteurs, ce qui invalide tout détournement de vecteur. Examinons comment on opère :

- JUMP RESTORE : on annule tout simplement l'effet de cette routine en remplaçant le vecteur de saut par un retour immédiat. On fait par prudence un peu plus. En effet la plupart des vecteurs se décomposent ainsi : 1) 1 Restart (1 octet), soit un saut automatique à une des 7 premières adresses multiples de 8.
- 2) l'adresse sur 2 octets en ROM ou RAM de la routine. Aussi certains programmes, plutôt que d'appeler une routine via son vecteur, vont lire son adresse et exécuter directement eux-mêmes la routine en question. Ceci est très fâcheux dans notre cas. C'est pourquoi on remplace les 2 octets par une fausse adresse, sans conséquence. J'ai choisi l'adresse 000F, où se trouve l'instruction RET (RETOUR) : un CALL y est donc sans effet. A noter que c'est une des rares adresses de même valeur pour les 3 types de CPC. Est-il nécessaire de préciser pour conclure qu'on a remplacé le restart par un RET ?
- MC START PROGRAM : comme vu au chapitre SYNTAXE DES ROUTINES, cette routine, après avoir effectué son travail, saute à l'adresse contenue dans HL. On remplace donc le Restart par un saut direct à cette adresse, grâce à l'instruction JP (HL). Aux 2 octets suivants ont été substitués l'adresse 001E, où l'instruction est encore commune aux 3 types de CPC ; il s'agit d'un JP (HL), comme avaient pu le deviner les plus perspicaces.

• Lancement du menu : par un simple RUN» est lancé le véritable programme de MENU, MENU2, que l'on va voir maintenant.

6) MENU2 : ce programme de menu n'est là qu'à titre indicatif et se propose simplement de donner l'exemple pour que chacun puisse utiliser ses propres menus avec l'acquis dû à MENU. MENU2 est donc très simple. Sa partie la plus intéressante est la ligne de DATA. Examinons la :

DATA BASIC,BINAIRE,MIXTE,0,basic,1,binaire,0,mixte
On y trouve d'abord en majuscules les noms des 3 programmes au menu, tels qu'on veut les porter à la connaissance de l'utilisateur. Puis suivent en minuscules les noms des 3 fichiers correspondant dans l'ordre aux 3 programmes. Chaque nom de fichier est précédé d'un indicateur : 0 ou 1. Un 0 indique un fichier qu'on peut lancer avec l'ordre BASIC RUN» tandis qu'un 1 signale un fichier qui nécessite

l'emploi de la RSX IRUN. Cet indicateur donne en quelque sorte le type du fichier. D'où la ligne BASIC de lancement :

READ t,a\$:IF t=0 THEN RUN a\$ ELSE IRUN,a\$.

La procédure à suivre s'avère donc, pour ceux qui en auraient douté jusque là, enfantine. Pour exemple encore, on peut consulter le listing de MIXTE, qui se contente de lancer avec IRUN le programme en LM «BINAIRE», lequel charge un écran. On va enfin pouvoir conclure.

CONCLUSION

Tout d'abord bravo au lecteur qui aura tout lu et tout compris du premier coup. Je me suis efforcé de rendre le maximum d'information accessible au plus grand nombre et espère y avoir réussi. C'est pourquoi je n'ai pas hésité parfois à me répéter, au risque d'agacer les connaisseurs, pour que les notions fondamentales soient bien assimilées. Mon souhait le plus cher est que cet article et particulièrement les routines et trucs présentés puissent rendre de grands services, que les formatages spéciaux de disquette ne soient plus un problème. A une prochaine fois !

Yannick GOUR

BIBLIOGRAPHIE :

- LA BIBLE DU PROGRAMMEUR DE L'AMSTRAD CPC - Micro Application - 1985
- LE LIVRE DU LECTEUR DE DISQUETTE AMSTRAD - Micro Application - 1985
- CLEFS POUR AMSTRAD - Système de base - Editions du P.S.I. - 1986
- CLEFS POUR AMSTRAD - Système disque - Editions du P.S.I. - 1986
- CP/M APPROFONDI - Sybex - 1980

N'oubliez pas !!

**36 • 15
ARCADES**

Des centaines de logiciels pour CPC à télécharger avec le kit ARCADES.

Voir bon de commande page 24

TRUCS POUR L'EXPLOITATION DE DISQUETTES EN FORMAT SPÉCIAL POUR LECTEUR A OU B

BOOT-DPB.BAS

```

100 '*****  

110 '* Installation de DPB *  

120 '*****  

130 DEFINT A-Z  

140 '  

150 '--- IMPLANTATION DU DPB EN &BE00 ---  

160 AD=&BE00:FOR i=0 TO 24:READ i$:POKE AD+i,VAL("&"&i$)&FF  

):NEXT  

170 '  

180 '--- DETERMINATION DE L'ADRESSE ---  

190 '--- DE LA RAMDOS ---  

200 ramdos=UNT(PEEK(&BE7D)+256*PEEK(&BE7E))  

210 '  

220 '--- DETERMINATION DU LECTEUR ACTIF ---  

230 '--- PAR LECTURE EN DEBUT DE RAMDOS ---  

240 lecteur=PEEK(ramdos)  

250 '  

260 '--- REMPLACEMENT DU POINTEUR DE DPB ---  

270 '--- DU LECTEUR ACTIF PAR &BE00 ---  

280 adptr=ramdos+&21A+lecteur*&10  

290 POKE adptr,ad AND &FF:POKE adptr+1,INT(ad/256) AND  

&FF  

300 '  

310 '--- DPB FORMAT "Data+" ---  

320 DATA 28,00,03,07,00,01,00,3F,00,C8,00,10,00,00,00,C  

1,0A,1C,2A,E5,02,04,00,00,FF

```

INITDOS 1.BAS

```

100 '*****  

110 '* INIT BACK (& ROM WALK) DETOURNE *  

120 '* copyright Y. GOUR (#4/1991) *  

130 '*****  

140 '  

150 DEFINT a-z:MODE 1:CALL &BC02  

160 INPUT"adresse de INIT BACK detourne":a:ad=a  

170 '  

180 ' --- installe la nlie routine ---  

190 b=a-1:MEMORY MIN((b AND 255)+(INT(b/256) AND 255)*2)>HD

```

INITDOS 2.BAS

```

100 '*****  

110 '* INIT BACK AMELIORE *  

120 '* copyright Y. GOUR (#4/1991) *  

130 '*****  

140 '  

150 DEFINT a-z:MODE 1:CALL &BC02  

160 INPUT"ad de INIT BACK ameliore":a:ad=a  

170 '

```

```

180 ' --- installe la nlie routine --->RK
190 b=a-1:MEMORY MIN((b AND 255)+(INT(b/256) AND 255)*2)>HD
56,HIMEM)
200 WHILE a$<>"FIN">NM
210 READ a$>NF
220 IF a$=" ">HN
THEN READ a$:POKE a,(ad+VAL("&"&a$)) AND
255:POKE a+1,INT((ad+VAL("&"&a$))/256) AND 255:a=a+2:GO
TO 240
230 IF a$<>"FIN" THEN POKE a,VAL("&"&a$):a=a+1>MZ
240 WEND>CE
250 IF a<>(ad+&40) THEN PRINT"Erreur":END>LH
260 '  

270 '--- sauve INIT BACK ameliore --->RK
280 SAVE"initdos2",b,ad,&40>YM
290 END>TE
300 '  

310 DATA 2A,7D,BE,11,FD,BD,1,3,0,7E,D5,ED,B0,1,17,2,87,>YU
28,2,E,27,C5,9,4E,23
320 DATA 66,69,1,18,0,ED,B0,3E,FF,12,21,F8,AB,11,40,0,E>ZD
,7,CD,CE,BC,E1,E3,ED
330 DATA 58,7D,BE,1,3,0,ED,B0,EB,C1,9,73,23,72,C9>QM
340 DATA FIN>CK

```

MENU.BAS

```

100 'RAZ et DPB >RB
110 DEFINT a-z:RESTORE 210:READ a$:WHILE a$<>"FIN":CALL >YJ
VAL("&"&a$):READ a$:WEND:UDISC:d=&BE00:FOR i=0 TO 24:R
EAD a$:POKE d+i,VAL("&"&a$):NEXT:a=UNT(PEEK(&BE7E)*256+
PEEK(&BE7D)):a=PEEK(a)*16+&21A+a:POKE a,d AND 255:POKE
a+1,INT(d/256) AND 255
120 'RSXs et INIT BACK detourne >RD
130 a=&A600:MEMORY a-1:GOSUB 170:CALL ad=a:&BF00:RESTOR >BH
E 310:GOSUB 170:POKE a,PEEK(&BCCF):POKE a+1,PEEK(&BCD0)
:POKE &BCCE,&C3:POKE &BCCB,&C3:b=ad AND 255:POKE &BCCF,
b:POKE &BCCC,b=b=INT(ad/256) AND 255:POKE &BCD0,b:POKE
&BCCD,b
140 'modifie START PROGRAM et JUMP RESTORE >RF
150 a=&BD16:POKE a,&E9:POKE a+1,&1E:POKE a+2,&0:a=&BD37:>TB
POKE a,&C9:POKE a+1,&F:POKE a+2,&
160 RUN"menu2 >LE
170 ad=a:a="" WHILE a$<>"FIN":READ a$:IF a$=" ">FQ
EAD a$:b=ad+VAL("&"&a$):POKE a,b AND 255:POKE a+1,INT(b
/256) AND 255:a=a+2 ELSE IF a$<>"FIN" THEN POKE a,VAL("
&"&a$):a=a+1
180 WEND:RETURN >MK
190 '  

200 'remise a zero >RC
210 DATA BD37,BCA7,BBFF,BB4E,BB00,FIN >FG
220 '  

230 'DPB >RF
240 DATA 28,0,3,7,0,D1,0,3F,0,C0,0,10,0,0,C1,A,1C,2A, >MT
E5,2,4,0,0,FF
250 '  

260 'LOAD et RUN >RJ
270 DATA 21,0,9,1,0,D,C3,D1,BC,0,0,0,0,15,C3,0,1D,C3, >YA
$54,4C,4F,41,C4,52,55,CE,0,B7,C8,3D,28,B,DD,66,1,DD,6E
,0,E5,DD,23,DD,66,1,DD,6E,0,46,23,56,23,56,21,0,C
0,EB,F5,CD,77,BC,38,5,F1,B7,C8,E1,C9,F1,B7,28,1,D1,EB,C
D,B9,BC,DC,7A,BC,C9
280 DATA 3D,C0,DD,66,1,DD,6E,0,46,23,56,21,0,C0,E >CA
B,CD,77,BC,D0,EB,CD,83,BC,D0,E5,CD,7A,BC,D8,E1,C9,FIN
290 '  

300 ' INIT BACK detourne >RD
310 DATA 22,0,2B,ED,53,0,2E,2A,7D,BE,11,FD,BD,1,3,0,7E, >FH
D5,ED,B0,1,17,2,B7,28,2,E,27,C5,9,4E,23,56,69,1,18,0,ED
,0,B0,3E,FF,12,21,F8,AB,11,40,0,E,7,CD,$47,E1,E3,ED,5B,7
D,BE,1,3,0,ED,B0,EB,C1,9,73,23,72,C9,CF,FIN

```

LOADRUN.BAS

```

10 '*****>LA
15 '* RSX LOAD ET RUN: *>LF
20 '* _LOAD sans Buffer *>LB
25 '* _RUN sans MC BOOT PROGRAM *>LG
30 '* copyright Y. GOUR (#4/1991) *>LC
35 '*****>LH
40 '  

45 DEFINT a-z:MODE 1:CALL &BC02>YE
50 INPUT"adresse des RSXs":a:ad=a>HP
55 b=a-1:MEMORY MIN((b AND 255)+(INT(b/256) AND 255)*25>GX
6,HIMEM)
60 WHILE a$<>"FIN">ME
65 READ a$>HG
70 IF a$=" ">EC
THEN READ a$:POKE a,(ad+VAL("&"&a$)) AND 2>EC
55:POKE a+1,INT((ad+VAL("&"&a$))/256) AND 255:a=a+2:GOT
0,B0
75 IF a$<>"FIN" THEN POKE a,VAL("&"&a$):a=a+1>LU
80 WEND>XB
85 IF a<>(ad+&75) THEN PRINT"Erreur":END>KP
90 SAVE"loadrun",b,ad,&75>XX
95 CALL ad>MB
100 END>RE
105 '  

110 DATA 21,0,9,1,0,D,C3,D1,BC,0,0,0,0,15,C3,0,1D,C3, >QR
$54,4C,4F,41,C4
115 DATA 52,55,CE,0,B7,C8,3D,28,B,DD,66,1,DD,6E,0,E5,DD >ZN
,23,DD,23,DD,66,1,DD
120 DATA 6E,0,46,23,5E,23,56,21,0,C0,EB,F5,CD,77,BC,38, >XD
5,F1,B7,C8,E1,C9,F1
125 DATA B7,28,1,D1,EB,CD,83,BC,DC,7A,BC,C9,3D,C0,DD,66 >YW
,1,DD,6E,0,46,23,5E
130 DATA 23,56,21,0,C0,EB,CD,77,BC,D0,EB,CD,83,BC,D0,E5 >XX
,C0,7A,BC,D8,E1,C9
135 DATA FIN>DB

```

MENU 2.BAS

```

10 MODE 1:LOCATE 19,6:PRINT"MENU":FOR i=1 TO 3:LOCATE i >RU
5,7+2*i:READ a$:PRINT i;a$:NEXT
20 i=VAL(INKEY$):IF i>4 OR i=0 THEN 20 ELSE k=i:FOR i=1 >VV
TO i:READ t,a$:NEXT:IF t=0 THEN RUN a$ ELSE URUN,aa$>VW
30 DATA BASIC,BINAIRE,MIXTE >XV

```

MIXTE.BAS

```

10 CLS:PRINT"Je suis un programme en BASIC":PRINT"Je me >UV
contente de lancer le programme binaire avec URUN":FOR
i=0 TO 3000:NEXT:a$="BINAIRE":URUN,aa$>VW

```

chronique du TELECHARGEMENT

FRANCOIS Ier

QUESTION : Qui fut le vainqueur de la bataille de Marignan ?

Turin

Rocroi

Milan

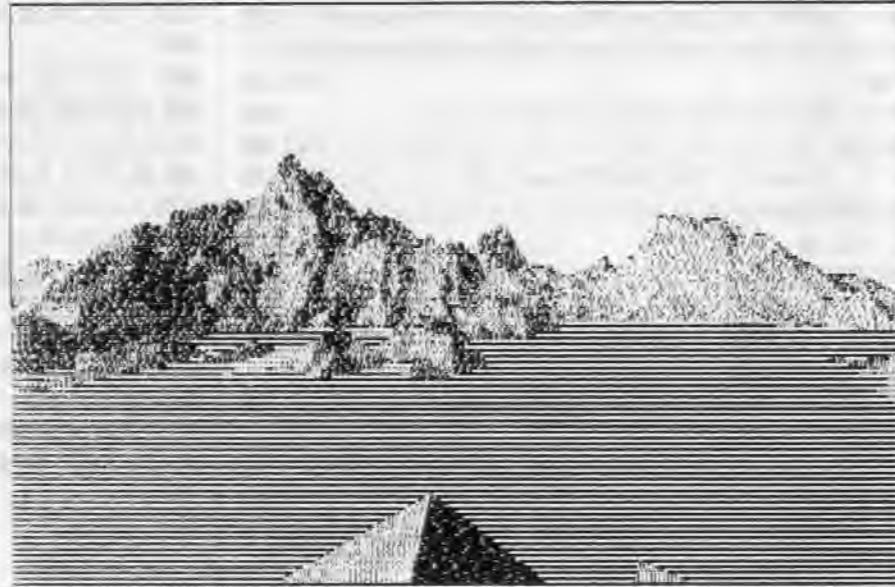
Marignan

Rome

HISTOIRE

Tout d'abord, il faut souligner l'utilité des remarques sur le serveur puisqu'ainsi, vous pouvez signaler les problèmes rencontrés lors du téléchargement. Pour Desktop, il manque effectivement le fichier Easy.doc. Ce dernier sera ajouté et vous pourrez le télécharger séparément. Pour Omicron Démô, il n'y a pas de fichier «Menu», celui-ci étant un reliquat du fanzine duquel il a été extirpé. Il suffit de supprimer cette référence dans le premier fichier (Omicron.bln).

Au programme aujourd'hui, voici Histoire, un logiciel éducatif envoyé spontanément par son auteur pour venir grossir la quantité de programmes disponibles. Le programme est constitué de séries de questions sur plusieurs périodes de notre histoire. En répondant correctement à toutes les séries, vous obtiendrez un code donnant accès à un jeu. Mais il faudra auparavant répondre et ce n'est pas facile, car, à chaque erreur, il faut reprendre à partir du début. Mais normalement, à la fin de votre «apprentissage» vous aurez acquis pas mal de connaissances historiques.



FRACTAL LANDSCAPES



qu'en modifiant tous les paramètres un peu au hasard, il est rare d'obtenir un bon résultat. De toutes façons, il est nécessaire d'être patient lorsque l'on a demandé un calcul avec ombrage et facettes pleines puisque cela peut aller jusqu'à plusieurs heures. Pour ceux qui voudront avoir des exemples, il suffira de télécharger la démo ainsi que les matrices utilisées pour la démo. Fractal Landscapes ne fonctionne que sur 6128 puisqu'il utilise les 64 Ko supplémentaires. Il faudra d'abord lancer Bankman, puis taper Symbol After 256 et ensuite lancer Fracrun.

Pour rester dans le domaine (merveilleux) des programmes de CPC Infos, voici la fournée des pro-

grammes du mois correspondant à 2 numéros (31 et 32). Easy Screen est un utilitaire permettant de composer des pages avec des menus et des sous-menus. En fait, Easy Screen est un générateur de programme. Une fois définies les caractéristiques de vos pages et les liens entre les menus, un programme basic est créé que vous pourrez utiliser ensuite dans vos propres créations. Ce programme est lui-même constitué de menus et possède toutes les explications nécessaires.

Oxysudis est un jeu de plate-formes se déroulant sur des planètes inconnues, avec un temps très limité et des tas de pièges. Sur l'ensemble des niveaux, ce jeu vous donnera du fil à retordre.

Un programme graphique, à paramètres multiples et qui permet de réaliser des motifs géométriques et multicolores, c'est Automate cellulaire. Comme son nom l'indique, il travaille au niveau du pixel et d'une «semence» aléatoire qui lui permet de réaliser des écrans très rapidement (routine en assembleur).

Le jeu que beaucoup de personnes attendent : Axys est enfin sur le téléchargement. Au moment où j'écris, je ne sais pas si ce program-

CREATION MATRICE	AFFICHAGE	CHARGEMENT	SAUVEGARDE	QUITTER
AFFICHAGE				
OK	OK	OK	OK	ECLAIRAGE
Lancement	X = 0	X = 0	X = 0	Eclairage
Observateur	Y = 0	Y = 0	Y = 0	X = 0
Eclairage	Dir = 0	Dir = 0	Dir = 0	Z = 0
Altitude zones planes = 0				
Facteur d'échelle = 1				
Angle cone de vision = 50				
Terplaine				
PaysageCarte				
SqueletteFacettesOmbrages				
Densité fill				
NOTICE				
X : entre -4000 et 4000				
Y : entre -4000 et 4000				
Dir : entre 0 et 359	-4000	0	4000	X

observateur
direction regardée
soleil

FRACTAL LANDSCAPES

me sera compacté avec CRUNCH20, étant donné sa taille mais vous le saurez facilement puisqu'il y aura alors un Z dans l'extension du nom de fichier.

Claude Le Moullec vous présente un autre jeu de son crû : Charlie Bond Sound. Il s'agit d'un jeu musical, où vous devez recomposer un air de musique sur 3 volets.

Pour les démos vous trouverez les démos de AST System : Joke demo, Short demo II, Demo3, Joke demo II, Equinox, Demo 6



allo
CPC infos

Attention, notez bien ce qui suit :
Dès le mois de septembre, tous les mercredis de 14 h à 17 h, la rédaction se tiendra à votre disposition au 99 52 98 11 pour tous renseignements techniques. En dehors de ce créneau les réponses techniques se feront par courrier (n'oubliez pas de mentionner votre adresse et joignez un timbre pour la réponse) ou bien par Minitel sur le 3615 code : MHZ.



BON DE COMMANDE

A expédier à : Editions SORACOM
La Haie de Pan - 35170 BRUZ

DESIGNATION	Prix unitaire	Quantité	Port	Montant
Pour chaque article, entourez le ou les n° choisis.				
ANCIENS Nos AMSTAR & CPC N° 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 35 - 36 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47	25 F (unité) 26 F (unité)		Franco	
DISQUETTES AMSTAR & CPC N° 20 - 21 - 22 - 23 - 24 - 25 - 26 - 27 - 28 - 29 - 30 (une disquette réunit 2 numéros consécutifs d'Amstar & CPC)	Abonné Non abonné	110 F (unité) 140 F (unité)	Franco	
ANCIEN No CPC Infos N° 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33		26 F (unité)	Franco	
DISQUETTES CPC Infos - MEGA SOUND - N° 31 (Am'Star 48 & CPC Infos 24) - N° 32 (CPC Infos 25 & 26) - N° 33 (CPC Infos 27 & 28) - N° 34 (CPC Infos 29 & 30) - N° 35 (CPC Infos 31 & 32) - N° 36 (CPC Infos 33 & 34)		Abonné : 110 F (unité) Non abonné : 140 F (unité)	Franco	
ANCIENS Nos CPC N° 11 - 13 - 14 - 18 - 23 - 24 - 25 - 26 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 36 - 37		25 F (unité)	Franco	
ANCIENS Nos HORS-SERIE CPC N° 7 - 8 - 9 - 10 - 12 - 13 - 14 - 15 - 16 N° 17 - 18 - 19 - 20 - 21 - 22 - 23		21 F (unité) 23 F (unité)	Franco	
DISQUETTES CPC ET DISQUETTES HORS-SERIE CPC N° 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 (une disquette réunit 2 numéros consécutifs de CPC) HS1 - HS2 - HS3 - HS4 - HS5 - HS6 - HS7 - HS8 - HS9 - HS10 - HS11 HS12 - HS13 - HS14 - HS15 - HS16 - HS17 - HS18 - HS19 - HS20 - HS21 - HS22 - HS23		Abonné : 110 F (unité) Non abonné : 140 F (unité)	Franco	
ABONNEMENT & DISQUETTES (non rétroactif)		600 F	Franco	
DISQUETTES "ARCADES" Téléchargement AMSTRAD CPC CABLE pour Téléchargement (Téléchargement non compatible avec 6128+ et 464+)		60 F	5 F	
LOGICIELS EDUCATIFS - LIVRES - DIVERS		89 F	5 F	
DESIGNATION	REFERENCE			
<i>Forfait port + 10 F par logiciel et 10 % pour livres</i>				
Pour tout envoi par avion : prendre contact avec le service commercial		<input type="checkbox"/> Facultatif : recommandé + 10 F par commande		
		10 F		
		MONTANT GLOBAL		
Je joins mon règlement : <input type="checkbox"/> chèque bancaire <input type="checkbox"/> chèque postal <input type="checkbox"/> mandat				
CPC Infos 34				
PAYEZ PAR CARTE BANCAIRE				
Nom : _____ Prénom : _____				
Adresse : _____				
Date d'expiration _____ Signature _____				
(inscrire les numéros de la carte, la date et signer)				
Date et signature _____ Signature _____				
Afin de faciliter le traitement des commandes, nous remercions notre aimable clientèle de ne pas grafer les chèques, et de ne rien inscrire au dos.				
Commande : La commande doit comporter tous les renseignements demandés sur le bon de commande (désignation ou référence si celle-ci existe). Toute absence de précision est sous la responsabilité de l'acheteur. La vente est conditionnée à l'acceptation du bon de commande par notre société, sur les articles disponibles uniquement. Les prix : Les prix indiqués sont valables du jour de la parution du catalogue jusqu'au mois suivant ou au jour de parution du nouveau catalogue, sauf erreur dans la libellé de nos tarifs au moment de la fabrication du catalogue, et de variation de prix importante des fournisseurs. Livraison : La livraison intervient après le règlement. Les délais de livraison étant de 10 à 15 jours environ, SORACOM ne pourra être tenue pour responsable des retards dus aux transporteurs ou aux grèves des services postaux. Transport : La marchandise voyage aux risques et périls du destinataire. La livraison se faisant par code postal ou par transporteur. Les prix indiqués sur le bon de commande sont valables sur toute la France métropolitaine, + 20 F par article pour Outre-Mer par avion et au-dessus de 5 kg nous réservons la possibilité d'ajuster le prix de transport en fonction du coût réel de celui-ci. Pour bénéficier d'un récu possible nous invitons notre aimable clientèle à opter pour l'envoi en recommandé. À réception des paquets, toute déclaration doit être signalée. Réclamation : Toute réclamation doit intervenir dans les dix jours suivants la réception des marchandises.				

SORACOM SOLDE A SES LECTEURS

Lots de jeux divers cassettes ou disquettes

Par 5 jeux **40 F** le lot de cassettes

Par 10 jeux **60 F** le lot de cassettes

Par 5 jeux **100 F** le lot de 5 disquettes

Par 10 jeux **150 F** le lot de 10 jeux



ATTENTION:

Il s'agit de lots en solde. Les jeux ne sont ni repris ni échangés.
Merci de ne pas demander au téléphone la liste de ces titres.

JUSQU'À EPUISEMENT DES STOCKS

Bon de commande

Je commande

- 5 jeux K7 pour 40 F
- 10 jeux K7 pour 60 F
- 5 jeux disc pour 100 F
- 10 jeux disc pour 150 F

Je joins mon règlement chèque bancaire
 chèque postal mandat

Nom : _____ Prénom : _____

Adresse : _____

Code postal : _____ Ville : _____

Date : _____ Signature : _____



JE REGLE PAR CARTE BANCAIRE

Date d'expiration _____ Signature _____

EASY SCREEN

Guy DUBUS

Valable pour CPC 664 - 6128

Avec ce générateur de menus, vous allez pouvoir créer des pages écran ayant une présentation professionnelle. Reportez-vous au numéro précédent pour avoir toutes les explications et les débuts des listings.

S U I T E

```

***:PRINT:PRINT"CREER/COMPLETER UNE SERIE DE PAGES (Editeur de pages)":PRINT:PRINT" * OPTION TEXTE";" - MENU : l'emplacement de la dernière option est r(servi au programme, qui";
390 PRINT" y inscrit ";CHR$(34);"QUITTE";CHR$(34);" >FN pour le menu principal (le premier) ou ";CHR$(34);"RETOUR";CHR$(34);" pour les";" suivants.":PRINT" Dans l'option ";CHR$(34);"Pointe sur S/Programme Code...";CHR$(34);", vous pouvez entrer ";
400 PRINT" - un label mnémonique pour la routine @ >XC appeler (dans VOTRE programme), ";" - le numéro de la première ligne de cette routine (précidé de";CHR$(34);%;"#";CHR$(34);");";
410 PRINT" - un numéro de page (type Message avec >PR réponse O/N attendue, ou Saisie).";" Dans les 3 cas, NE PAS OUBLIER D'AJOUTER UN RETURN EN FIN DE ROUTINE OU DE";" PAGE !"
420 PRINT" - Utilisez la commande ";CHR$(34);"Fin de >GQ texte";CHR$(34);" (CTRL+F) dans le type Message, si vous";" voulez une fenêtre vide (avec ou sans titre) : fenêtre de travail de votre";
430 PRINT" programme, ou pour incrustation d'une autre fenêtre. L'écriture du programme";" générée en sera altérée, mais cette commande est irréversible, sauf si vous";
440 PRINT" décidez de refaire la page.":PRINT:PRINT" >RR * OPTION COULEURS";" Pour chaque page, définissez de 4 INK (@ à 3) ; INK 0 est affecté au fond";
450 WHILE k<40:GOTO 320:WEND
460 PRINT CHR$(23);CHR$(0);:CLEAR
470 '
480 MODE 2:INK 0,15:INK 1,1:LOCATE 35,1:PRINT"*** AIDE >BR

```



on de rang, que vous devez indiquer";"dans l'(diteur de pages lorsqu'il vous est demandé (Pointe sur S/Menu Page...)."PRINT:PRINT"COMMANDES DANS LE PROGRAMME D'APPLICATION":PRINT

870 PRINT" Les mêmes que dans EASY SCREEN."PRINT" >CA Messages n'attendant pas de réponse : taper une touche quelconque pour";"passer @ la suite.";" Fenêtres de saisie : entrée alphanumérique ou numérique, en mode insertion.";

880 PRINT" Corrections :";CHR\$(242);"/";CHR\$(243);"/CLR/ >BK DEL. Passage au champ précédent/suivant et validation : ";CHR\$(240);"/";CHR\$(241);"Fin de saisie : ENTER ; impossible de sortir si tous les champs ne sont pas";" saisis."

890 ----- >TH

900 help\$(2,1)=" Sélection:";CHR\$(241);"/";CHR\$(243);"/" >QD "+CHR\$(240);"/";CHR\$(242);" Validation:ENTER":m\$(0)=" Tapez un nom de fichier, puis ENTER":m\$(1)=" ou ENTER pour choisir sur répertoire":m\$(2)=" Annulation de la commande : COPY"

910 help\$(3,0)=" Réponse : o/O/n/N":help\$(1,1)=" Corrections:DEL Validation:ENTER":m\$(3)="Choix du numéro: "+CHR\$(240);"/";CHR\$(241);" Validation:ENTER"

920 help\$(3,1)=help\$(1,1):help\$(4,0)=help\$(2,1):help\$(4 >ND ,1)=" Sortie d'option : COPY":help\$(5,0)="Changer de fenêtre:tapez un No valide":help\$(5,1)="de 1 @ 4. Correctlon:retapez un numéro"

930 help\$(5,2)=help\$(4,1):help\$(6,0)="Curseur=coin haut >GW /gauche de la fenêtre":help\$(6,1)="D(placements du curseur":CHR\$(241);"/";CHR\$(243);"/";CHR\$(240);"/";CHR\$(242);"help\$(6,2)="Apparition de la fenêtre minimum:ENTER"

940 help\$(7,0)="Déplacer la fenêtre:";CHR\$(241);"/";CHR\$(243);"/";CHR\$(240);"/";CHR\$(242);"help\$(7,1)="Changer les dimensions:SHIFT+";CHR\$(241);"/";CHR\$(243);"/";CHR\$(240);"/";CHR\$(242);"help\$(7,2)="Centrer en largeur/en hauteur:C/CTRL+C"

950 help\$(7,3)="Modifier marges:G/D+";CHR\$(242);"/";CHR\$(243);" Sortie:COPY":help\$(8,0)=" Tapez un No en hexadécimal (1 @ F)":help\$(8,1)=" Changer de cadre : tapez un autre No":help\$(8,2)=help\$(4,1)

960 help\$(9,0)=" Caractère précédent/suivant:";CHR\$(242);"TD "+"/";CHR\$(243);"help\$(9,1)=" Corrections:CLR/DEL Valide la ligne:ENTER":help\$(9,2)="Ligne précédent./suiv.:";CHR\$(240);"/";CHR\$(241);" Effacer:CTRL+E":help\$(9,3)=" Fin de fenêtre:CTRL+F"

970 help\$(9,4)=help\$(4,1):help\$(10,0)=" Options séparées par un seul espace":help\$(10,1)=help\$(9,0):help\$(10,2)=help\$(9,1):help\$(10,3)=help\$(4,1):help\$(11,0)=" 2 chiffres maximum":help\$(11,1)=help\$(1,1)

980 help\$(12,0)="10 caractères alphanumériques maxi :";>YB help\$(12,1)=" * Label, débutant par une lettre":help\$(12,2)=" * No de ligne appellée, précédent de #":help\$(12,3)=" * No de page (2 chiffres maxi)":help\$(12,4)=help\$(1,1)

990 help\$(13,0)=" (A)lphanumérique ou (N)umérique":help\$(13,1)=help\$(1,1):help\$(14,0)="6 caractères alphanumériques maxi":help\$(14,1)="commençant par une lettre":help\$(14,2)="Code de type (\$,!,%)" facultatif"

1000 help\$(14,3)="Variable numérique = réelle par défaut":ME



t":help\$(14,4)=help\$(1,1):help\$(15,0)=" Selection et validation:";CHR\$(241)+"/"+CHR\$(243)+"/"+CHR\$(240)+"/"+CHR\$(242):help\$(15,1)="Corrections:DEL Sortie d'option COPY":GOSUB 2550:CLS:LOCATE 23,1
1010 PRINT"QUELQUES CONSEILS EN VRAC POUR FINIR":PRINT" * Un menu doit toujours étre appell par un autre menu, directement ou par;"une page intermédiaire terminée par un GOTO, pour que le retour automatique";
1020 PRINT"fonctionne correctement.":PRINT" * Si vous faites pointer une option d'un menu sur un sous-programme de;"votre cru par un label alphanumérique, n'oubliez pas, dans l'application, de";
1030 PRINT"remplacer ce label par le No de ligne correct.";
* La sauvegarde n'est effective que lors que vous retournez au Menu;"Principal (travail terminé, ou pause). Si vous avez lieu de craindre une coupure";
1040 PRINT"de courant, demandez périodiquement @ intérrompre le travail (dans ";CHR\$(34);";RETOUR";CHR\$(34);";);";vous serez sauvé, et vos données aussi...";
* Une astuce pour examiner une page déjà créée : ";CHR\$(34);";Modifier Page...";
1050 PRINT CHR\$(34);";sans";ajouter de fenêtres, et alors cher tout de suite @ ";CHR\$(34);";COULEURS";CHR\$(34);"; qui affiche l'ensemble;"de la page), puis ";CHR\$(34);";RETOUR";CHR\$(34);". NDTA : les No de page et de rang ne sont pas alors";
1060 PRINT"précisés dans l'éditeur.":PRINT" * Fene tructures ";CHR\$(34);";Message";CHR\$(34);"; sans cadre : si vous remplissez complètement la;"dernière ligne, le texte remonte. Validez par ENTER, sortez par COPY et revenez";
1070 PRINT"dans ";CHR\$(34);";TEXTE";CHR\$(34);": vous constaterez que tout est en ordre. Si alors vous vous levez;"corriger un caractère, effacez-le d'abord par CLEAR ou DEL (vous êtes en mode;"insertion)."
1080 PRINT" * Si une fenêtre incrustée doit, @ un moment donné, laisser la place @ une autre plus petite ou décalée, vous devez ajouter, au bon endroit du programme";
1090 PRINT"initial : CLSF si le fond de la fenêtre environnante est de la même couleur,";sinon PAPER#f,x :CLSF, x (tant le No de PAPER de la fenêtre environnante.)"
1100 GOSUB 2550:fes=1:ON BREAK CONT >DL
1110 "-----" >XF
1120 nom\$="":ON ERROR GOTO 2620:MODE 1:BORDER 0:INK 0,0:DLR :INK 1,4:INK 2,2:INK 3,15:PAPER#1,1:PEW#1,2:PAPER#3,0:w f=1:g=2:cd=39:ch=1:cb=13:cad=5:GOSUB 2570:WINDOW#1,3,3,8,2,12
1130 LOCATE#1,1,2:PEW#1,3:PRINT#1,"";m1\$(1):PRINT#1:PR NT#1,"";m1\$(2):PRINT#1:PRINT#1,"";m1\$(3):PRINT#1:PRI NT#1,"";m1\$(4):PRINT#1:PRINT#1,"";m1\$(5)
1140 PEW#3,3:GOSUB 2510:PEW#3,2:LOCATE#3,1,2:PRINT#3,he >FW ip#(2,1):GOSUB 2290:r=0:WHILE r>13:GOSUB 2530:GOSUB 290:FOR i=5 TO 9:IF r<>com(i) THEN NEXT i
1150 ON i-5 GOSUB 2300,2310,2300,2310:GOSUB 2290:WEND:i >RF i=5 THEN CALL &
1160 PAPER#2,3:PEW#2,1:wf=2:cg=4:cd=37:ch=15:cb=19:cad= >RQ

3:GOSUB 2570
1170 PEH#2,2:WINDOW#2,5,36,16,1B:CLS#2:PRINT#2," Nom du fichier de données ?":GOSUB 1250
1180 CLS#3:IF i>1 THEN PRINT#3,help\$(2,1);m\$(2):GOSUB 2360:GOTO 1260 ELSE PRINT#3,m\$(0);m\$(1);m\$(2);
1190 GOSUB 2530:IF r=127 THEN IF POS#(2)=13 THEN PRINT CHR\$(7); ELSE PRINT#2,CHR\$(24);CHR\$(8);CHR\$(16);CHR\$(24)
);:r\$="":GOTO 1240
1200 IF r=13 THEN IF nom\$="" THEN CLS#3:PRINT#3,help\$(2,1);m\$(2):GOSUB 2360:GOTO 1260 ELSE GOSUB 1250:GOTO 1260
1210 IF r=224 THEN i=1:GOSUB 2490:GOSUB 2290:GOTO 1140
1220 IF r<45 AND (r<48 OR r>57) AND (r<65 OR r>90) THE BG N PRINT CHR\$(7):GOTO 1190
1230 IF POS#(2)=21 THEN PRINT CHR\$(7):GOTO 1190
1240 nom\$=LEFT\$(nom\$,POS#(2)-13)+r\$:PRINT#2,CHR\$(24);r\$>BN ;CHR\$(24):GOTO 1190
1250 LOCATE#2,13,3:PRINT#2,CHR\$(24);SPACE#(8);STRING\$(8,0,XL ,8);nom\$:STRING\$(LEN(nom\$),8);CHR\$(24);:RETURN
1260 IF nom\$="" THEN i130 ELSE PRINT CHR\$(21);file\$=no >FX m\$+".NDX":OPENIN file\$:INPUT#9,pp,pmax,rmax:PRINT CHR\$(6);:CLOSEIN
1270 ON i GOTO 1310,1370,1420,2260 >ZR
1280 "-----" >YD
1290 * Crier/Compliter une série >YE
1300 "-----" >XG
1310 IF pp THEN IF pp<pmax THEN fidx=1:fsav=1:CHAIN"CRE ECRAN.BAS" ELSE CLS#2:LOCATE#2,9,1:PRINT#2,"Série compilée !":LOCATE#2,6,3:PRINT#2,"Appuyez sur une touche":CALL &BB06:i=3:GOSUB 2490:IF fdir=1 THEN fdir=0:CLS#1:GOTO 1130 ELSE GOSUB 2290:GOTO 1140
1320 CLS#2:LOCATE#2,4,1:PRINT#2,nom\$;".DAT n'existe pas >HY ";:LOCATE#2,4,2:PRINT#2,"Voulez-vous le créer O/N ?":;C LS#3:PRINT#3,help\$(3,0);help\$(1,1);m\$(2):LOCATE#2,15,3 :fup=2:GOSUB 2330
1330 IF rr THEN fsav=0:CHAIN"CREECRAN.BAS" ELSE fan=0:G QR OSUB 2490:GOSUB 2290:GOTO 1140
1340 "-----" >YA
1350 * Crier le programme d'application >YB
1360 "-----" >YC
1370 IF pp=pmax THEN CHAIN"CREBASIC.BAS" >KF
1380 GOSUB 2480:IF rr THEN fidx=1:fsav=1:CHAIN"CREECRAN.BAS" ELSE fan=0:i=5:GOSUB 2490:IF fdir=1 THEN fdir=0:C LS#1:GOTO 1130 ELSE GOSUB 2290:GOTO 1140
1390 "-----" >YF
1400 * Modifier/Mettre @ jour une série >XH
1410 "-----" >XJ
1420 MODE 1:BORDER 0:INK 0,0:INK 1,15:INK 2,9:INK 3,2:W DF INDO#1,1,13,1,11:PAPER#1,1:WINDOW#2,14,40,1,15:PAPER#2 ,3:WINDOW#4,1,40,17,19:PAPER#4,2:PEW#4,0:WINDOW#5,1,12,14,14:PAPER#5,1:PEW#5,0
1430 CLS#1:PEW#1,2:PRINT#1,CHR\$(150);STRING\$(11,154);CHR\$(156);:FOR i=1 TO 2:PRINT#1,CHR\$(149);STRING\$(11,9);CHR\$(157);:FOR i=1 TO 6:PRINT#1,CHR\$(149);STRING\$(11,9);CHR\$(149);:NEXT i
1440 PRINT#1,CHR\$(147);STRING\$(11,154);CHR\$(158);:PEW#1 >FW ,3:LOCATE#1,2,2:PRINT#1,"MISE A JOUR":LOCATE#1,5,3:PRI NT#1,"PAGES":WINDOW#1,3,11,5,10:CLS#1:FOR i=1 TO 6:PRI

NT#1,m2\$(i);:NEXT i:CLS#5:PRINT#5,nom\$;".DAT";
1450 CLS#2:PEW#2,0:PRINT#2,CHR\$(150);STRING\$(7,154);CHR\$(158);STRING\$(7,154);CHR\$(158);CHR\$(156);:FOR i=1 TO 2:PRINT#2,CHR\$(149);STRING\$(7,9);CHR\$(149);CHR\$(149);:NEXT i
1460 PRINT#2,CHR\$(151);STRING\$(7,154);CHR\$(159);STRING\$(7,9);CHR\$(159);CHR\$(155);STRING\$(9,154);CHR\$(157);:FOR i=1 TO 10:PRINT#2,CHR\$(149);STRING\$(7,9);CHR\$(149);STRING\$(7,9);CHR\$(149);STRING\$(9,9);CHR\$(149);:NEXT i
1470 PRINT#2,CHR\$(147);STRING\$(7,154);CHR\$(155);STRING\$(7,154);CHR\$(155);CHR\$(153);:PEW#2,1:LOCATE#2,2,2:PRINT#2,"NUMERO";STRING\$(2,9);"NUMERO";STRING \$(2,9);"NOMBRE DE";
1480 LOCATE#2,2,3:PRINT#2,"DE RANG";CHR\$(9);"DE PAGE";C PY HR\$(9);"FENETRES":CLS#4:PEW#3,1:GOSUB 2510
1490 PEW#3,3:WINDOW#2,17,35,5,14
1500 OPENIN file\$:INPUT#9,pp,pmax,rmax:FOR i=1 TO pmax:MX INPUT#9,index(i,0),index(i,1):NEXT i:CLOSEIN:DIM tri(99):FOR i=1 TO pmax:tri(index(i,0))=i:NEXT i:i=it=i:PEW#2,0 :LOCATE#2,1,it:FOR ir=1 TO MIN(i,rr):GOSUB 2000:NEXT ir:ir=i:iup=1
1510 PRINT#3,help\$(2,1):PRINT#3,m\$(3);m\$(2); >MQ
1520 "-----" >YA
1530 PAPER#1,2:PEW#1,0:GOSUB 2020:r=0:WHILE r<13:GOSUB >HW 2530:PAPER#1,1:PEW#1,3:GOSUB 2020:FOR i=5 TO 9:IF r<c om(i) THEN NEXT i
1540 ON i-5 GOSUB 2030,2040,2030,2040:PAPER#1,2:PEW#1,0 >UN :GOSUB 2020:WEND
1550 IF iup=6 THEN GOSUB 1960:GOTO 1120 >FN
1560 PAPER#1,1:PEW#1,3:GOSUB 2020:PAPER#2,2:LOCATE#2,1, >HZ it:GOSUB 2000:PAPER#2,3:PRINT#4,";com\$(iup)":GOSUB 20 60:GOSUB 2170
1570 IF fan THEN fan=0:LOCATE#2,1,it:GOSUB 2000:CLS#4:G KQ OTO 1530
1580 IF iup>2 AND index(tri(ir),1)=0 THEN LOCATE#4,2,2: >EU PEN#4,3:PRINT#4,"Page vide ! ";:PEN#4,0:PRINT#4,"Appuyez sur une touche":CLEAR INPUT:CALL &BB06:CLS#4:LOCATE #2,1,it:GOSUB 2000:GOTO 1530
1590 ON iup GOTO 1630,1680,1750,1810,1940 >GK
1600 "-----" >XX
1610 * Ajouter >YA
1620 "-----" >YB
1630 pmax=pmax+1:rmax=rmax+1:IF num=rmax-1 THEN tri(rma x)=pmax ELSE FOR i=rmax TO num+2 STEP -1:tri(i)=tri(i-1):NEXT i:tri(i)=pmax
1640 GOSUB 2150:CLS#4:GOTO 1530 >YQ
1650 "-----" >YE
1660 * Déplacer >YF
1670 "-----" >YG
1680 r1=num:LOCATE#4,5,2:PRINT#4,"APRES LA PAGE DE RANG >VR No ";:GOSUB 2060:GOSUB 2170:IF fan THEN GOTO 1570 ELSE r2=num:stri=tri(r1)
1690 IF r2>r1 THEN FOR i=r1 TO r2-1:tri(i)=tri(i+1):NEXT i:tri(i)=stri
1700 IF r1>r2 THEN FOR i=r1 TO r2+2 STEP -1:tri(i)=tri(i-1):NEXT i:tri(i)=stri
1710 IF r1=r2 THEN PRINT CHR\$(7);:LOCATE#2,1,it:GOSUB 2 >XL 000:CLS#4:GOTO 1530 ELSE GOSUB 2150:CLS#4:GOTO 1530
1720 "-----" >YC



```

2110 IF ir>rmax THEN RETURN >XX
2120 IF it<10 THEN LOCATE#2,1,it:GOSUB 2000:it=it+1:ir= >HD
ir+i:PAPER#2,2:LOCATE#2,1,it:GOSUB 2000:PAPER#2,3:GOSUB
2000:RETURN >XX
2130 LOCATE#2,1,1:sir=ir+i:FOR ir=sir-9 TO sir-1:GOSUB >KY
2000:NEXT ir:PAPER#2,2:GOSUB 2000:PAPER#2,3:ir=sir:GOSU
B 2060:RETURN
2140 '----- >XX
2150 sir=ir:i=ir-it+1:FOR j=1 TO rmax:index(tri(j),0)=j >HH
:NEXT j:LOCATE#2,1,1:FOR ir=i TO MIN(i+9,rmax):GOSUB 20
00:NEXT ir:ir=sir:RETURN
2160 '----- >YB
2170 CLEAR INPUT:CALL &BB06:WHILE INKEY(18)=-1 AND INKE >JD
Y(6)=-1
2180 IF INKEY(0)>-1 THEN GOSUB 2000 >BZ
2190 IF INKEY(2)>-1 THEN GOSUB 2110 >BW
2200 IF INKEY(9)>-1 THEN fan=1:RETURN >FD
2210 WEND:RETURN >MZ
2220 '----- >XJ
2230 '----- >XX
2240 ' Supprimer un fichier de donnees >YA
2250 '----- >YB
2260 CLS#2:LOCATE#2,5,2:PRINT#2,"Confirmez par ";CHR$(3) >UJ
41;"o";CHR$(34);"; ou ";CHR$(34);";"O";CHR$(34);:CALL &BB0
6:IF INKEY(34)>-1 THEN file$=nom$+"NDX":ERA,file$;fil
e$=nom$+".DAT":ERA,file$ >YD
2270 i=4:GOSUB 2490:fdir=0:CLS#1:GOTO 1130 >KP
2280 '----- >YE
2290 MOVE 40,404-(32*i),3,1:DRAWR LEN(m1$(1))*16+16,0:D >HD
RAWR 0,-26:DRAWR -(LEN(m1$(1))*16+16),0:DRAWR 0,26:RETU
RN
2300 i=1-5*(i=1)-1:RETURN >UA
2310 i=1+(i=5)+1:RETURN >UB
2320 '----- >XX
2330 GOSUB 2530:IF rr#"0" THEN rr=1:PRINT#fmp,"OUI"; EL >CD
SE IF rr#"N" THEN rr=0:PRINT#fmp,"NON"; ELSE IF rr=224 T
HEN rr=0:fan=1:RETURN ELSE PRINT CHR$(7);:GOTO 2330
2340 GOSUB 2530:IF rr=13 THEN RETURN ELSE IF rr=224 THEN >FE
rr=0:fan=1:RETURN ELSE IF rr=127 THEN rr=""":FOR i=1 TO 3
:PRINT#fmp,CHR$(8);CHR$(16);:NEXT i:GOTO 2330 ELSE PRIN
T CHR$(7);:GOTO 2340 >YC
2350 '----- >YD
2360 fdir=1:WINDOW SWAP 0,1:CLS:PEN 3:LOCATE 1,1:DIR," >NY
.DAT":xd=1:yd=4:GOSUB 2450
2370 IF nom#" " THEN nom$="":CLS#2:PRINT#2," >ZY
Aucun fichier de donnees ! ";:LOCATE#2,6,3:PRINT#2,"App
uyez sur une touche":CALL &BB06:WINDOW SWAP 0,1:fdir=0
:CLS#1:GOSUB 2490:RETURN
2380 LOCATE xd,yd:IF INKEY(8)=0 AND xd=1:PRI >WP
NT n$;:GOSUB 2450 >FR
2390 IF INKEY(1)=0 AND xd=1 THEN LOCATE 16,yd:IF COPYCH >BJ
R$(#0)" " THEN xd=16:LOCATE 1,yd:PRINT n$;:GOSUB 2450 >FH
2400 IF INKEY(0)=0 AND yd>4 THEN yd=yd-1:PRINT n$;:GOSU
B 2450 >FU
2410 IF INKEY(2)=0 THEN LOCATE xd,yd+1:IF COPYCHR$(#0)>XP
" " THEN LOCATE xd,yd:PRINT n$;:yd=yd+1:GOSUB 2450 >FX
2420 IF INKEY(18)=0 OR INKEY(6)=0 THEN WINDOW SWAP 0,1: >GZ
RETURN >FQ
2430 IF INKEY(9)>-1 THEN WINDOW SWAP 0,1:fdir=0:CLS#1:G >XQ

```

```

2440 CALL &BB06:GOTO 2380 >TA
2450 LOCATE xd,yd=n$=""":FOR i=xd TO xd+11:LOCATE i,yd:r >WD
$=COPYCHR$(#0):n$=n$+r$:NEXT i:LOCATE xd,yd:PAPER 3:PEN
0:PRINT n$;STRING$(12,8);:PAPER 1:PEN 3:nom$=LEFT$(n$,
8):GOSUB 1250:RETURN >YE
2460 '----- >YE
2470 CLS#2:LOCATE#2,5,1:PRINT#2,nom$".DAT n'existe pas >PF
":LOCATE#2,6,3:PRINT#2,"Appuyez sur une touche":CALL
&BB06:nom$=""":RETURN >PC
2480 CLS#2:PRINT#2," Srie incomplte ! ":LOCATE#2,P2
2,2,2:PRINT#2,"Voulez-vous la compiler O/N ? ";:CLS#3:P
8INT#3,help$(3,8);help$(1,1);m$(2);:LOCATE#2,15,3:fmp=2
:GOSUB 2330:RETURN >P2
2490 WINDOW#2,4,37,15,19:PAPER#2,0:CLS#2:nom$=""":RETURN >CT
2500 '----- >XX
2510 wf=3:cg=1:cd=40:ch=21:cb=25:cad=2:GOSUB 2570:WINDO >PC
W#3,2,39,22,24:CLS#3:RETURN >PC
2520 '----- >YB
2530 CLEAR INPUT:r$=""":WHILE r$=""":r$=UPPER$(INKEY$):WE >ER
ND:r=ASC(r$):RETURN >PC
2540 '----- >YD
2550 LOCATE 30,25:PRINT CHR$(24);"APPUYEZ SUR UNE TOUCH >VK
E":CHR$(24);:CALL &BB06:RETURN >PC
2560 '----- >YF
2570 hb=cb-ch-1:gd=cd-cg-1:WINDOW#wf,cg,cd,ch,cb:CLS#wf >DU
2580 PRINT#wf,CHR$(box(cad,1));STRING$(gd,box(cad,2));C >T
HR$(box(cad,3));
2590 FOR i=1 TO hb:PRINT#wf,CHR$(box(cad,4));STRING$(gd )QE
,9);CHR$(box(cad,5));:NEXT i >PC
2600 PRINT#wf,CHR$(box(cad,6));STRING$(gd,box(cad,7));C >LL
HR$(box(cad,8));:RETURN >PC
2610 '----- >YB
2620 PRINT CHR$(6);:IF DERR=146 THEN CLOSEIN:pp=0:pmax= >EA
#1:rmax=0:RESUME NEXT >PC
2630 ON ERROR GOTO # >ML
2640 '----- >YE
2650 DATA 5,6,16,127,13,240,241,242,243,224,3,67,71,68, >PA
244,245,246,247 >PC
2660 DATA "CREER/COMPLETER UNE SERIE DE PAGES","CREER L >MV
E PROGRAMME D'APPLICATION","MODIFIER/METTRE A JOUR UNE
SERIE","SUPPRIMER UN FICHIER DE DONNEES","QUITTER LE PR
OGRAMME" >PC
2670 DATA "AJOUTER ","DEPLACER ","SUPPRIMER ","MODIFIER >NV
","REFAIRES ","RETOUR "
2680 DATA "AJOUTER 1 PAGE APRES LE RANG No ","DEPLACER >KB
LA PAGE DE RANG No ","SUPPRIMER LA PAGE No ","MODIFIER
LA PAGE No ","REFAIRES LA PAGE No "
2690 DATA 135,131,139,133,138,141,140,142 >FR
2700 DATA 150,154,156,149,149,147,154,153 >FH
2710 DATA 250,251,252,253,219,254,220,255 >FU
2720 DATA 229,234,235,236,237,239,248,249 >FX
2730 DATA 221,222,223,224,225,226,227,228 >FQ
2740 DATA 42,42,42,42,42,42,42,42 >YM
2750 DATA 238,238,238,238,238,238,238,238 >PG
2760 DATA 203,203,203,203,203,203,203,203 >FN
2770 DATA 230,230,230,230,230,230,230,230 >PP

```

```

2780 DATA 231,231,231,231,231,231,231,231 >FZ
2790 DATA 232,232,232,232,232,232,232,232 >FJ
2800 DATA 233,233,233,233,233,233,233,233 >FJ
2810 DATA 128,244,128,247,246,128,245,128 >FN
2820 DATA 207,207,207,207,207,207,207,207 >FV
2830 DATA 127,127,127,127,127,127,127,127 >FE
2840 DATA 2,1,2,4,0,7,7,33,10,15,3,1,2,3,14,4,2,"NOMBRE >WG
DE PAGES-ECRAN ?"
2850 DATA 1,1,2,1,0,2,4,37,1,23,8,1,3,2,0,0,0,"PAGE No >KH
RANG No"
2860 DATA 1,1,0,4,0,0,5,36,3,5,6,1,3,2,27,2,1,"NOMBRE D >PG
E FENETRES ?"
2870 DATA 1,4,2,1,0,3,5,36,5,22,2,2,1,0,0,0,0," >FB
FENETRE No","","TYPE :"
2880 DATA 2,4,2,3,0,5,12,30,7,17,3,3,0,2,0,0,0,"MESSAGE >QK
","MENU HORIZONTAL","MENU VERTICAL","SAISIE"
2890 DATA 3,1,0,1,1,0,6,35,11,11,1,2,2,1,0,0,0,"VOULEZ- >AG
VOUS UN CADRE O/N ?"
2900 DATA 3,2,0,1,1,0,6,35,13,14,1,2,2,1,0,0,0,"TITRE O >AJ
U EN-TETE EN","PREMIERE LIGNE O/N ?"
2910 DATA 3,2,0,1,1,0,6,35,16,17,1,2,2,1,0,0,0,"REPONSE >VN
PAR OUI OU NON","ATTENDUE O/N ?"
2920 DATA 3,1,0,1,1,0,6,35,19,19,1,2,2,1,0,0,0,"LIGNES >RB
CENTREES O/N ?"
2930 DATA 3,1,0,1,1,0,6,35,21,21,1,2,2,1,0,0,0,"LIGNES >VT
JUSTIFIEES O/N ?"
2940 DATA 5,6,0,2,0,0,1,40,25,25,1,0,2,3,0,0,0,"No ", "F >LB
ENETRE","CADRE","TEXTE","COULEURS","RETOUR"
2950 DATA 5,1,0,1,0,0,1,40,25,25,1,1,2,3,0,0,0,"POINTE >BH

```

SUR"
2960 DATA 7,2,0,2,0,0,12,40,25,25,1,1,2,3,0,0,0,"S/MENU >RV
","S/PROGRAMME"
2970 DATA 7,1,0,4,0,0,12,40,25,25,1,1,2,3,16,1,2,"S/MEN >LW
U PAGE No "
2980 DATA 7,1,0,4,0,0,12,40,25,25,1,1,2,3,18,1,10,"S/PR >QA
OGRAMME CODE"
2990 DATA 7,1,0,4,0,0,12,25,25,1,1,2,3,10,1,2,"LONGUE >BF
UR"
3000 DATA 7,1,0,4,0,0,13,19,25,25,1,1,2,3,6,1,1,"TYPE:" >WR
3010 DATA 7,1,0,4,0,0,20,40,25,25,1,1,2,3,10,1,6,"VARIA >CG
BLE"
3020 DATA 7,2,2,1,1,1,13,28,11,14,2,1,2,3,0,0,0,"PAGE T >XA
ERMINEE?"," O/N ?"
3030 DATA 7,4,2,1,0,1,9,31,10,16,3,1,2,3,0,0,0,"Vous >KR
devez dfinir","toutes les fenetres","Pressez une tou
che"
3040 DATA 7,2,2,1,0,7,10,31,9,17,8,1,2,3,0,0,0,"VOUL >GU
EZ-VOUS"
3050 DATA 7,2,0,3,0,0,11,30,12,18,2,1,2,3,0,0,0,"MODIFI >UC
ER LA PAGE ?","REFAIRES LA PAGE ?"
3060 DATA 7,3,0,1,1,0,11,30,12,16,2,1,2,3,0,0,0,"EFFACE >QR
R L'ECRAN AU","CHANGEMENT DE PAGE"," O/N ?"
3070 DATA 7,3,0,1,1,0,11,30,12,16,2,1,2,3,0,0,0," INTE >AU
RROMPTE LE"," TRAVAIL O/N?","
3080 DATA 7,2,2,1,0,2,11,30,12,16,6,1,2,3,0,0,0,"SAU >GG
VEGARDE"
3090 DATA 1,3,2,1,0,3,4,37,10,15,3,2,3,1,0,0,0,">SPM
CHARGEMENT DU FICHIER","Ah si j'avais un accs direct!"

GESTION BANCAIRE 6128

LA GESTION DE COMPTE BANCAIRE INDISPENSABLE POUR VOTRE CPC

Quelques caractéristiques :

- Gère jusqu'à 10 comptes (banque, épargne, caisse...).
- Codes secrets possibles pour chacun des comptes.
- Saisie des opérations très simple, avec aide en ligne.
- Fonction archivage, vous permettant de stocker année par année vos opérations.
- Fonction TRIER, pour obtenir des listes d'opérations par dates croissantes.
- A l'aide de POINTER, vous pouvez effectuer la liaison avec votre relevé de banque 'officiel'.

NOMBREUSES POSSIBILITES DE SORTIES :

- Recherche particulière répondant à 1 ou plusieurs critères parmi les suivants : -DATES DE DEBUT ET DE FIN
-MONTANTS MINIMUM ET MAXIMUM
-UN LIBELLE PARTICULIER
- Liste de chèques pouvant répondre aux critères suivants : -DATES DE DEBUT ET DE FIN
-NUMEROS DE DEBUT ET DE FIN
- Relevé complet -ENTRE DATES
-AFFICHAGE SOLDE REEL OU SOLDE OPERATIONS POINTEES.
- Sorties sur ECRAN, IMPRIMANTE ou DISQUETTE.
- UTILITAIRES IMPRIMANTE, ECRAN, COPIE D'ECRAN...
- UTILISE LES 128 Ko DE VOTRE CPC 6128.

SUPPORT / SORTIE		CRITÈRE	
C	DATE	TYPE	N°
1	01/01/82	RECHERCHE	VERIFICATION INITIAL
2	01/01/82	CHQUE	100000
3	01/01/82	CHQUE	CONSULTATION MEDICAM
4	01/01/82	CHQUE	PHARMACIE
5	01/01/82	CHQUE	REINFORCEMENT S.S.
6	01/01/82	CHQUE	TELEPHONE
7	01/01/82	CHQUE	REINFORCEMENT
8	01/01/82	CHQUE	REINFORCEMENT
9	01/01/82	CHQUE	REINFORCEMENT
10	01/01/82	CHQUE	REINFORCEMENT
11	01/01/82	CHQUE	REINFORCEMENT
12	01/01/82	CHQUE	REINFORCEMENT
13	01/01/82	CHQUE	REINFORCEMENT
14	01/01/82	CHQUE	REINFORCEMENT
15	01/01/82	CHQUE	REINFORCEMENT
16	01/01/82	CHQUE	REINFORCEMENT
17	01/01/82	CHQUE	REINFORCEMENT
18	01/01/82	CHQUE	REINFORCEMENT
19	01/01/82	CHQUE	REINFORCEMENT
20	01/01/82	CHQUE	REINFORCEMENT
21	01/01/82	CHQUE	REINFORCEMENT
22	01/01/82	CHQUE	REINFORCEMENT
23	01/01/82	CHQUE	REINFORCEMENT
24	01/01/82	CHQUE	REINFORCEMENT
25	01/01/82	CHQUE	REINFORCEMENT
26	01/01/82	CHQUE	REINFORCEMENT
27	01/01/82	CHQUE	REINFORCEMENT
28	01/01/82	CHQUE	REINFORCEMENT
29	01/01/82	CHQUE	REINFORCEMENT
30	01/01/82	CHQUE	REINFORCEMENT
31	01/01/82	CHQUE	REINFORCEMENT
32	01/01/82	CHQUE	REINFORCEMENT
33	01/01/82	CHQUE	REINFORCEMENT
34	01/01/82	CHQUE	REINFORCEMENT
35	01/01/82	CHQUE	REINFORCEMENT
36	01/01/82	CHQUE	REINFORCEMENT
37	01/01/82	CHQUE	REINFORCEMENT
38	01/01/82	CHQUE	REINFORCEMENT
39	01/01/82	CHQUE	REINFORCEMENT
40	01/01/82	CHQUE	REINFORCEMENT
41	01/01/82	CHQUE	REINFORCEMENT
42	01/01/82	CHQUE	REINFORCEMENT
43	01/01/82	CHQUE	REINFORCEMENT
44	01/01/82	CHQUE	REINFORCEMENT
45	01/01/82	CHQUE	REINFORCEMENT
46	01/01/82	CHQUE	REINFORCEMENT
47	01/01/82	CHQUE	REIN

MODBASE.BAS

```

1 '
2 'MODBASE.BAS
3 '
10 SYMBOL AFTER 219
20 SYMBOL 219,38,38,38,38,38,38,38,38,38
30 SYMBOL 220,0,0,255,0,0,255,255,0
40 SYMBOL 221,0,55,72,72,55,72,72,72
50 SYMBOL 222,0,219,36,36,219,0,0,0
60 SYMBOL 223,0,236,18,18,236,18,18,18
70 SYMBOL 224,72,72,48,72,72,48,72,72
80 SYMBOL 225,18,18,12,18,18,12,18,18
90 SYMBOL 226,72,72,72,55,72,72,55,0
100 SYMBOL 227,0,0,0,219,36,36,219,0
110 SYMBOL 228,18,18,18,236,18,18,236,0
120 SYMBOL 229,0,127,64,73,86,72,72,80
130 SYMBOL 234,0,255,0,153,102,0,0,0
140 SYMBOL 235,0,254,2,146,106,18,18,10
150 SYMBOL 236,80,72,72,80,80,72,72,80
160 SYMBOL 237,10,18,18,10,10,18,18,10
170 SYMBOL 239,80,72,72,86,73,64,127,0
180 SYMBOL 248,0,0,0,102,153,0,255,0
190 SYMBOL 249,10,18,18,106,146,2,254,0
200 SYMBOL 250,0,63,127,96,96,103,100,100
210 SYMBOL 251,0,255,255,0,0,255,0,0
220 SYMBOL 252,0,252,254,6,6,230,38,38
230 SYMBOL 253,100,100,100,100,100,100,100
240 SYMBOL 254,100,100,103,96,96,127,63,0
250 SYMBOL 255,38,38,230,6,6,254,252,0
260 DEFINT a-z:del#=CHR$(8)+CHR$(16):DEF FNx=-(1+fcad/2) >AH
)(i=0)-mg*(i>0):DEF FNy=-2*(i=0)-(1+fcad/2/(ftype<>1)+
ftitre+(i-1)*(1-(ftype<>1)))*(i>0)
270 DIM I$(12),com(8),hpos(9),rep$(10),cx(10),cy(10),lc >MH
(i10),fc(10)
280 DATA 13,240,241,242,243,16,127,5 >BP
290 FOR i=1 TO 8:READ com(i):NEXT i:MODE 1:GOTO 2010 >VU
300 DATA 135,131,139,133,138,141,148,142 >EQ
310 DATA 150,154,156,149,149,147,154,153 >ER
320 DATA 250,251,252,253,219,254,228,255 >EC
330 DATA 229,234,235,236,237,239,248,249 >EF
340 DATA 221,222,223,224,225,226,227,228 >EZ
350 DATA 42,42,42,42,42,42,42,42 >XE
360 DATA 238,238,238,238,238,238,238,238 >FG
370 DATA 203,203,203,203,203,203,203,203 >EX
380 DATA 230,230,230,230,230,230,230,230 >EY
390 DATA 231,231,231,231,231,231,231,231 >EH
400 DATA 232,232,232,232,232,232,232,232 >ET
410 DATA 233,233,233,233,233,233,233,233 >EX
420 DATA 128,244,128,247,246,128,245,128 >ED
430 DATA 207,207,207,207,207,207,207,207 >EN
440 DATA 127,127,127,127,127,127,127,127 >RK
450 **** AFFICHAGE DES FENETRES **** >RK
460 ERASE 1$:DIM I$(12):READ f,11,ftype,fcad,ftitre,fre >WB
p,fcentre,cad,cg,cd,ch,cb,mg,a,b,c:IF ftitre=3 THEN REA
D I$(0)
470 IF ftype=2 THEN READ sp:ERASE hpos:DIM hpos(9) >VA

```

```

1490 '
1500 ' Interrogation Clavier >YG
1510 ' >XJ
1520 CLEAR INPUT:r$="" WHILE r$="" :c$=UPPER$(INKEY$):WE >EP
ND:r=ASC(r$):RETURN
1530 ' >YB

```

MODSAISI.BAS

```

1150 '
1160 ' MODSAISI.BAS >YA
1170 ' >YB
1180 WINDOW SWAP 0,f:CURSOR 1,1:i=1 >DR
1190 PEN b:LOCATE cx(i),cy(i):PRINT STRING$(lc(i),CHR$(>BM
210));:IF rep$(i)="" THEN rep$(i)="" ELSE PEN c:LOCAT
E cx(i),cy(i):PRINT MID$(rep$(i),1-(LEFT$(rep$(i),1)="
"));
1200 PEN b:LOCATE cx(i),cy(i):flag=0 >EK
1210 r$=INKEY$:IF r$="" THEN 1210 ELSE r=ASC(r$) >PA
1220 FOR j=1 TO 8:IF r<>com(j) THEN NEXT j >HV
1230 ON j GOSUB 1330,1360,1370,1380,1390,1400,1420,1430 >EU
:IF j<9 THEN ON flag GOTO 1250,1260:GOTO 1210
1240 tc=fe(i)+1:ON fc GOSUB 1270,1290:GOTO 1210 >NY
1250 GOTO 1190 >MG
1260 CURSOR 0:WINDOW SWAP 0,f:RETURN >FF
1270 IF r>31 AND r<126 THEN GOSUB 1310 >CA
1280 RETURN >FE
1290 IF r>47 AND r<58 THEN GOSUB 1310 >CZ
1300 RETURN >EH
1310 IF LEN(rep$(i))=lc(i) THEN PRINT CHR$(7)::RETURN >YM
1320 x=POS(#0):rep$(i)=LEFT$(rep$(i),x-cx(i))+r$+MID$(r >KD
ep$(i),x-cx(i)+1):PEN b:LOCATE cx(i),cy(i):PRINT rep$(i
);:LOCATE x+1,cy(i):RETURN
1330 IF l1=1 THEN IF rep$(i)="" THEN PRINT CHR$(7)::LOC >GD
ATE cx(i),cy(i):RETURN ELSE PEN c:LOCATE cx(i),cy(i):PR
INT rep$(i);:flag=2:RETURN
1340 PEN c:LOCATE cx(i),cy(i):PRINT MID$(rep$(i),1-(LEF >VA
T$(rep$(i),1)=" "));:flag=1:FOR i=1 TO l1:IF rep$(i)=""
THEN PRINT CHR$(7)::LOCATE cx(i),cy(i):RETURN ELSE NEX
T i
1350 flag=2:RETURN >NG
1360 IF l1=1 THEN RETURN ELSE PEN c:LOCATE cx(i),cy(i): >NP
PRINT MID$(rep$(i),1-(LEFT$(rep$(i),1)=" "));:i=~|j|(j=
1)-(i-1)*(l>1):flag=1:RETURN
1370 IF l1=1 THEN RETURN ELSE PEN c:LOCATE cx(i),cy(i): >MH
PRINT MID$(rep$(i),1-(LEFT$(rep$(i),1)=" "));:i=~(l-1)
-(i+1)*(l>1):flag=1:RETURN
1380 IF POS(#0)=cx(i) THEN PRINT CHR$(7)::RETURN ELSE P >BY
PRINT CHR$(8)::RETURN
1390 IF POS(#0)=LEN(rep$(i))+cx(i) OR VPOS(#0)=cy(i)+1 >CV
THEN PRINT CHR$(7)::RETURN ELSE PRINT CHR$(9)::RETURN
1400 IF POS(#0)=cx(i)+lc(i) OR VPDS(#0)=cy(i)+1 THEN PR >ZF
INT CHR$(7)::RETURN
1410 PRINT CHR$(16)::xx=POS(#0):rep$(i)=LEFT$(rep$(i),x >TM
-x-cx(i))+MID$(rep$(i),xx-cx(i)+2):PEN b:LOCATE cx(i),cy
(i):PRINT rep$(i);CHR$(210)::LOCATE xx, cy(i):RETURN
1420 IF POS(#0)=cx(i) THEN PRINT CHR$(7)::RETURN ELSE P >TV
PRINT CHR$(8)::GOSUB 1410:RETURN
1430 rep$(i)=""::LOCATE cx(i),cy(i):PRINT STRING$(lc(i), >EE
CHR$(210));:LOCATE cx(i),cy(i):RETURN

```

MODMENUH.BAS

```

880 '
890 ' MODMENUH.BAS >TH
900 ' >RK
910 iop=0:GOSUB 1460:hpos(l)=mg: lign$=I$(1) >MD
920 FOR j=2 TO l1:lign$=lign$+SPACE$(sp):hpos(j)=LEN(l1 >RP
gn$)+mg: lign$=lign$+I$(j):NEXT j: PEN#f,c:LOCATE#f,mg,1:
PRINT#f,lign$;
930 PEN#f,b:GOSUB 980: PEN#f,c >YR
940 r=0: WHILE r<>13:GOSUB 1520:GOSUB 980 >HC
950 FOR j=1 TO 5:IF r<>com(j) THEN NEXT j >GW
960 ON j GOSUB 1480,1450,1460,1450,1460,1470 >JZ
970 PEN#f,b:GOSUB 980: PEN#f,c:WEND:choix=iop:RETURN >AR
980 LOCATE#f,hpos(iop),1:PRINT#f,I$(iop)::RETURN >WZ

```

MODMENUV.BAS

```

1060 '
1070 ' MODMENUV.BAS >YA
1080 ' >YB
1090 iop=l:gxy=i6*(cg+mg)-38:gyv=16*(25-ch-fcad/2-ftitr >HT
e)+38 >HE
1100 GOSUB 1140: r=0: WHILE r<>13:GOSUB 1520:GOSUB 1140 >XC
1110 FOR j=1 TO 5:IF r<>com(j) THEN NEXT j >HP
1120 ON j GOSUB 1480,1450,1460,1450,1460,1470 >KA
1130 GOSUB 1140: WEND:choix=iop:RETURN >HC
1140 long=12+LEN(I$(iop))*16:MOVE gxy,gyv-32*iop,c,1:DR >JH
AWR long,0:DRAWR 0,-26:DRAWR 0,26:RETURN■

```

MODREP.BAS

```

990 '
1000 ' MODREP.BAS >XD
1010 ' >XE
1020 GOSUB 1520:IF r=79 THEN PRINT#f,"OUI";:rr=r ELSE I >PR
F r=78 THEN PRINT#f,"NON";:rr=r ELSE PRINT CHR$(7)::GOT
O 1020
1030 GOSUB 1520:IF r=13 THEN RETURN ELSE IF r=127 THEN >QK
r$="":FOR i=1 TO 3:PRINT#f,del$::NEXT i:GOTO 1020
1040 PRINT CHR$(7)::GOTO 1030 >XH
1050 ' rr=79 : OUI rr=78 : NON >XJ

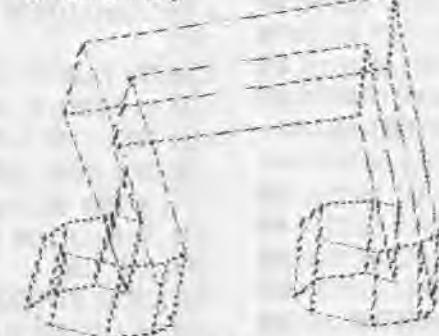
```

MUSIC MACHINE

Ayant rencontré de grandes difficultés pour faire fonctionner la Music Machine (impossibilité de saisir un échantillon (pardon de "sampler" un son), impossibilité de faire exécuter le morceau pré-programmé sur mon clavier, impossibilité de jouer à partir de mon clavier les "samples" préenregistrés ???), ceci n'étant pas faute d'avoir compris comment fonctionnait l'interface, ni d'avoir tout essayé, je vous livre quand même mes impressions de musicien amateur sur la Music Machine.

N'ayant travaillé qu'au casque, je dois avouer qu'il manque quelque chose de très utile : un potentiomètre de réglage de volume d'écoute. Je crois avoir perdu environ 10 dB à chaque oreille. Ou alors il aurait fallu fixer d'usine un niveau sonore un peu moins... DESTROY ! Deuxième point fort désagréable : le problème de ronflement résiduel. Il faut s'amuser à déplacer la Music Machine par rapport à l'écran, aux câbles d'alimentation pour trouver un emplacement plus ou moins acrobatique pour l'interface afin que celle-ci ne capte trop de parasites.

(le seul fait d'approcher la main provoque un ronflement terrible. Une vraie chatte, on peut passer des heures à la caresser et l'écouter ronronner).



Bon, le ton est donné, vous vous en doutez : critique sévère. Un bon point quand même pour la présentation. Les écrans sont bien faits, le passage d'un écran à l'autre est psychédélique. Mais gros problème : Peut-on vraiment parler de Music Machine quand on voit les limitations données par le manuel ? Tout d'abord en éditeur de mesure : On ne peut avoir que deux notes alignées verticalement tout en sachant que la Music Machine ne pourra en jouer qu'une.



Si vous voulez entendre l'autre, il vous faudra connecter un instrument MIDI qui, lui, pourra jouer les deux notes simultanément (même avec un instrument MIDI connecté). Il faut avouer qu'un maximum de deux notes simultanées c'est un peu léger pour jouer une partition. La plage de programmation des notes sur la portée est de quatre octaves...

La partie éditeur de rythmes vous permettra quant à elle de programmer des arrangements de vos "samples" en mémoire. Là, vous pourrez superposer au maximum trois "samples" parmi un choix de huit mémorisables. C'est un peu mieux que l'éditeur de mesures. Grâce à cet éditeur, vous pourrez mémoriser votre partition rythmique. Il faut savoir que vous pouvez enregistrer un maximum de 1.22 secondes de son "samplé" en tout. Comme vous pouvez mémoriser huit "samples" au maximum, cela donne une moyenne de 0.15 seconde par "sample". Quant à la qualité des sons de percussion préenregistrés, on peut dire qu'elle est assez bonne. Un bon point aussi pour l'éditeur de "sample" qui possède en plus une fonction REVERSE qui peut s'avérer intéressante.

Il faut savoir aussi que la Music Machine ne pourra pas restituer à la fois la musique et les percussions simultanément. Il faudra choisir l'une ou l'autre. Par contre, il vous sera possible d'envoyer la partition musicale vers un synthétiseur MIDI, de faire jouer la partie percussion en synchronisation par la Music Machine. On pourra même, si on le désire, superposer à cela un accompagnement en le jouant directement sur les touches du synthétiseur. Voici un petit résumé des choses que vous pourrez faire en association avec un instrument MIDI :

- Jouer un des "sample" en mémoire dans la Music Machine à partir d'un clavier ou tout autre instrument MIDI.

- Envoyer au synthétiseur MIDI la partition musicale programmée et superposer en le jouant sur votre clavier un des "sample" en mémoire ou bien une des voies que votre clavier MIDI peut générer.

- En utilisant le microphone, vous pourrez "sampler" n'importe lequel des sons générés par votre synthétiseur MIDI et l'éditer en utilisant l'éditeur de "sample".

- Si vous connectez un clavier MIDI sur votre Music Machine, votre clavier sera en mesure de restituer jusqu'à deux notes simultanément (alors que la Music Machine seule ne peut en reproduire qu'une en mode exécution d'une partition musicale).

Il est clairement spécifié dans la notice que la Music Machine est tout ce qu'on veut sauf un séquenceur du fait que le logiciel est trop gourmand de mémoire, et que si l'on est assez riche pour posséder un sé-

quenceur MIDI, on peut l'utiliser comme matériel principal pour piloter à la fois un clavier externe, un synthétiseur de sons de percussion externe et la Music Machine pour la génération de sons "samplés".

En résumé, je pense que le principal intérêt de la Music Machine est de pouvoir capturer un son - voire le "SAMPLER" comme ils disent

autre Manche - et de le restituer en association avec un clavier MIDI sur toute la gamme accessible par celui-ci. Donc grâce à la Music Machine, vous pourrez faire ce que ne peut faire votre synthétiseur c'est-à-dire jouer des sons "SAMPLES".

Thierry Manglon



Participez à CPC Infos

L'E PROGRAMMEUR

Nom : _____ Prénom : _____

Adressé complète : _____

Tél. : _____ Age : _____

Attestation sur l'honneur

Je soussigné _____ déclare être l'auteur du programme ci-joint et ne l'avoir jamais proposé à une autre revue.

Le _____ A _____

* Ne pouvant être tenu pour responsable de l'acheminement des paquets postaux, nous conseillons à notre aimable clientèle de choisir l'envoi en recommandé.

Remplissez soigneusement ce coupon et joignez-le à votre programme, sur cassette ou disquette.
Envoyez le tout à CPC Infos - La Haie de Pan - 35170 BRUZ

L'E PROGRAMME

Nom : _____

Catégorie Jeu Utilitaire Educatif

Taille : _____

Périphériques utilisées : _____

Support Cassette Disquette

Comptabilité (testée) avec : _____

464 664 6128 PCW 8256

Signature



RECTIFICATIF

JACQUETTE POUR 464

Aussi incroyable que cela puisse paraître, le célèbre All Gator est un inconditionnel du CPC 464. Cela est peut-être dû au fait qu'il a été l'un des tout premiers en

```

62 RESTORE 57:FOR H=150 TO 165:READ A$:POKE H,VAL(A$)+ASC(X):NEXT
63 DATA 00,5E,00,00,66,02,00,75,BB,CD,60,BB,32,08,00,09

640 CALL 150,X,Y:LPM=PEEK(200):IF CHR$(LPM)<>"." THEN 690
650 MM=MNP2:FOR H=0 TO 15:Z=(X-B)+H:CALL 150,Z,Y:LPM=PEEK(200)+MNP*(MM,F)+NUMP*(F)+CHR$(LPM)+MNP:LOCATE Z-B,Y:PRINT CHR$(24)+MNP*(N
MM=F+MM+F):MM
660 POKS MM,41=LEFT$(NAME(MM,41),LEN(MM,41)-10,32)+" "+RIGHT$(N
MM,41-11,21)+" K"
670 Y=Y+1:IF Y>Y THEN 690
680 GOTO 650
690 Y=4:R=4+20:IF R>71 THEN 700 ELSE 650
700 FOR H=1 TO 3:CALL 150,H,RF+1:LM=PEEK(200)+RF*(F)+RS*(F)+DRES(LPM):NEXT F:RF

```

Routine assembleur remplaçant la fonction COPYCHR\$ inexistante sur 464.

THE ENERGY OF WOK

```

270 ...:POKE &9048,0:GOTO 250
280 ...:POKE &904B,0:GOTO 250
290 ...:POKE &904D,40:GOTO 250

```

Il faut rajouter une ligne :

```

55 POKE &9048,&C8:POKE &904B,&C8:POKE
&904E,&C8:FOR A=&6180 TO &6180+255:POKE
A+9945,PEEK(A):NEXT

```

Si le personnage devient invisible à un niveau, il faut éteindre l'ordinateur, recharger le jeu et passer au tableau suivant en modifiant la ligne 240:POKE &901A,tableau:CALL &9023:GOTO 260.

Dans ce listing, une erreur rendait impossible l'accès aux niveaux 2, 3 et 4. Voici les quelques lignes à modifier (les instructions à rajouter sont en gras).

OXYSDUDIS

La présentation d'Oxydudis est certes facultative mais encore faudrait-il avoir le listing correspondant si l'on choisit l'option «écran de présentation». Ce listing était absent du numéro précédent, le voici donc :

```

10 MODE 0:PAPER 0:BORDER 0:INK 1,26:INK 2,24:IN 0FH
K 3,15:INK 4,6:INK 5,11:INK 6,19:INK 7,3:INK 8,23:INK 9
,13:INK 10,18:INK 11,9:INK 12,2:INK 13,1:INK 14,24,6:IN
K 15,9:CLS:DEFINT a-z
20 MODE 2:PAE 1:LOCATE 17,1:PRINT /*CREATION DE LA PAGE >VX
DE PRESENTATION D'OXYSUDIS*/
30 FOR t=1 TO 550:NEXT t:MODE 0 >V
40 FOR i=1 TO 800:x=i+INT(RND*639):y=1+INT(RND*399):PLD 20A
T x,y,1:NEXT i
50 RESTORE 60:FOR w=1 TO 4:READ x,y,r,cpl,c1(i),c1 >BH
21,c1(3):st=1:GOSUB 70:NEXT w:GOTO 190
60 DATA 370,150,110,290,350,1,2,3,170,300,70,130,160,2, >ZG
3,4,470,350,40,440,460,8,5,12,300,350,28,290,300,13,6,9

70 DEG:ORIGIN 0,0:ORIGIN x,y:FOR i=0 TO 360:PLOT COS(i) >UZ
*x,SIN(i)*r,15:NEXT i:ORIGIN 0,0
80 a=(x-r)*4:b=y:r=a:g=b >UE
90 IF TEST(h,g)<>15 THEN PLOT h,g,c1(st) >HP
100 g=g-2:IF TEST(h,g)<>15 THEN PLOT h,g,c1(st):GOTO 10 >AW
0
110 g=b >PJ
120 g=g+2:IF TEST(h,g)<>15 THEN PLOT h,g,c1(st):GOTO 12 >AY
0
130 g=b >QA
140 h=h+2 >CJ
150 IF h=cpl THEN cpl=0:st=st+1:GOTO 170 >HT
160 IF h=cp2 THEN cp2=0:st=st+1 >ZK
170 IF TEST(h,g)<>15 THEN GOTO 90 >AA
180 RETURN >ZE
190 INK 16,0 >VA
200 SAVE"OXYSUDIS.PRE",b,&C000,64000 >ET
210 END >RG

```

TRUCS EN VRAC



Christian CHANTREAU

Voici un mini listing permettant (encore UN) d'obtenir sur l'imprimante, le catalogue ou le Directory complet de tous les users (y compris ceux situés entre 16 et 255). Et ce d'une façon très simple, sans pokes et en moins de 10 lignes.

```

10 /* Impression du Catalogue ou Directory COMPLET d'une disquette de tous les
USERS : numé de 16 à 255
20 MODE 2:INPUT "Numéro de la disquette :";nomdisk:INPUT "Face :";face
30 a=1:INPUT "Entrez le numéro d'user à éditer ";u$(a):u$(a)=VAL(u$(a)):IF u$(a)
=>16 THEN 40 ELSE 30
40 num=1:FOR a=1 TO nomdisk:POKE 1470,u$(a);DAT=GOSUB 60:PRINT#rep,acar (la papie
t pour imprimer le prochain CAT à la suite ou changer de page PUIS UNE TPOUCHE
:CALL 4816:NEXT
50 CLS:PRINT "Impression terminée. Un autre traitement :ON":CALL 4816:IF INKEY(
3A)=0 THEN 20 ELSE 60
60 PRINT#0,CHR$(15):PRINT#0," DISQUE ";nomdisk;" FACE ";face:ph=0:pv=1
70 ph=ph+1:IF ph=80 THEN 80 ELSE 90
80 PRINT#0,CHR$(15):PV=PV+1:IF PV<15 THEN RETURN ELSE ph=0:GOTO 70
90 UPDATE ph,pv=1:POKE#0,(40):PRINT#0,a$(a):GOTO 70

```

BONNES VACANCES DE LA PART DE TOUTE L'ÉQUIPE DE GIGA CPC INFOS !

DISQUE WORDSTAR FACE 1

Drive A: user 0

ADR	,DATA	1K	MAILMSG.DVR	8K	TOT	,MAC	1K	WSMS555	,DVR	31K
ACST	,MAC	8K	TELFILE.BAK	4K	NS	,COM	16K	WSOVLY1	,DVR	34K
LT	-	2K	TELFILE.MAC	8K	NS	,INS	39K			

2k free

DISQUE FACE

Drive A: user 229

IAP	,BSG	1K	ZAP	,BIN	8K				
-----	------	----	-----	------	----	--	--	--	--

2k free

CPC
jeu

CHARLIE BONDSOUND

Claude LE MOULLEC

Valable pour CPC 464 - 664 - 6128

Une balle sautillante doit capturer des couleurs (cubes colorés) afin d'allumer des projecteurs (cubes grillagés).

CHARLIE.BAS

```

10 REM ::::::::::::::::::::
20 REM ::::::::::::::::::::
30 REM : CHARLIE BOND SON :
40 REM : by :
50 REM :
60 REM : CLAUDE LE MOULLEC :
70 REM : 83 RUE CURIE :
80 REM : 22420 PLOUARET :
90 REM : TEL 96 38 94 24 :
100 REM ::::::::::::::::::::
110 MEMORY &8800:LOAD "ICHALIBIN",&8820
120 REM ::::::::::::::::::::
130 REM :
140 REM : VARIABLES DE BASE :
150 REM :
160 REM ::::::::::::::::::::
170 DEFINT a-z:MODE 0:BORDER 1
180 RESTORE 190:FOR h=0 TO 15:READ a:INK h,a:NEXT
190 DATA 0,18,26,9,24,12,18,7,8,16,1,12,3,15,3,6
200 DIM sp(30):FOR h=0 TO 25:sp(h+1)=&8820+(100*h):NEXT >YR
210 DIM PY(20):RESTORE 230:FOR h=1 TO 9:READ a$ >NA
220 PY(h)=VAL("0"+a$):PY(h+8)=VAL("0"+a$):NEXT >PV
230 DATA BFFB,E09B,C18B,E22B,C31B,E38B,C4AB,E54B,C63B >XH
240 DEF FN PO(X,Y)=PY(Y)*(X#5) >YR
250 DEF FN PK(X,Y)=PEEK1&9E5F+(Y*16)+X >GJ
260 DEF FN LT(X,Y)=&BFFE+(Y-1)*80+(X-1)*2 >HK
270 ENT 4,10,-1,3:ENV 4,10,-1,3:ENT 3,20,3,5:ENV 14,1,5 >PE
,1,10,1,4,5,-3,1 >DEB
280 ENV 1,15,-1,1:ENT 2,10,-1,0,10,1,1:ENV 2,1,-1,4:ENV >HR
,10,15,-1,1 >VA
290 ENV 15,1,5,1,10,1,4,5,-3,1:ENT 14,40,-10,1:ENT 15,5 >MM
,-20,1,60,10,1 >TG
300 GOSUB 280 >QC
310 VIE=4:SC=0:REC=0:TA=1 >GU
320 A$="" SCORE LIFES LEVEL HIGHT":ZL=1:YL >VG
,24:GOSUB 350 >KE
330 A$="" 00000 000 000 00000":ZL=1:YL >JY
,25:GOSUB 350 >WX
340 GOTO 430 >XX
350 A$=UPPER$(A$):FOR T=1 TO LEN(A$):PS=(ASC(MID$(A$,T,1)))-48 >YB
360 IF PS<0 THEN PS=43 >TB
370 DATA 0,18,26,9,24,12,18,7,8,16,1,12,3,15,3,6 >MZ

```

orsque tous les projecteurs d'une même couleur sont allumés, un disque apparaît dans la partie contraire de celle où se trouve la balle à ce moment.

Ce jeu comprend 2 fichiers.
CHARLIE : fichier basic, corps principal du jeu.

CHALIBIN : fichier binaire (sprites, tableaux et routines LM) ;

DATACHAR est un fichier basic de data qui, une fois lancé, donne CHALIBIN.

```

370 CALL &8ECC,FN Lt(ZL,YL)+(T*2),&BEEC+(PS*16):NEXT:RE >GD
TURN >XG
380 REM :::::::::::::::::::: >XB
390 REM : >LC
400 REM : BRANCHEMENT TAB >VR
410 REM : >KF
420 REM :::::::::::::::::::: >XW
430 DN ta GOTO 440,450,460,470,480,490,500,510,520,530, >JV
540,550,560 >XV
440 RESTORE 2710:CALL &A2A8,&91B0:GOTO 570 >JE
450 RESTORE 2720:CALL &A2A8,&92C0:GOTO 570 >JJ
460 RESTORE 2730:CALL &A2A8,&93D0:GOTO 570 >JN
470 RESTORE 2740:CALL &A2A8,&94E0:GOTO 570 >JT
480 RESTORE 2750:CALL &A2A8,&95F0:GOTO 570 >JX
490 RESTORE 2760:CALL &A2A8,&9700:GOTO 570 >JC
500 RESTORE 2770:CALL &A2A8,&9810:GOTO 570 >JX
510 RESTORE 2780:CALL &A2A8,&9920:GOTO 570 >JB
520 RESTORE 2790:CALL &A2A8,&9A30:GOTO 570 >JN
530 RESTORE 2800:CALL &A2A8,&9B40:GOTO 570 >JH
540 RESTORE 2810:CALL &A2A8,&9C50:GOTO 570 >JM
550 RESTORE 2820:CALL &A2A8,&9D60:GOTO 570 >JR
560 TA=1:GOTO 430 >LC
570 READ GJ,GM,GR,X,Y:POR=0:EC=1:CALL &A1EC:mu=0 >UN
580 fin=0:gan=0:GOSUB 1840:GOSUB 1910:GOSUB 1980:GOSUB 2010 >ER
590 CUB=0:tt=100:READ M,N:IF M=1 THEN GOSUB 2430 >QU
600 CALL &A1EC:CALL &A230,FN PO(X,Y),SP(15):D7=0:INK 12 >AW
,3 >XX
610 REM :::::::::::::::::::: >KJ
620 REM : >YT
630 REM : ROUTINE PRINCIPALE: >LA
640 REM : >XB
650 REM :::::::::::::::::::: >PT
660 IF FIN=1 THEN 2100 >DC
670 IF INKEY(74)*(INKEY(8)=0 THEN 1000 >DZ
680 IF INKEY(75)*(INKEY(1)=0 THEN 1110 >CW
690 IF INKEY(72)*(INKEY(0)=0 THEN 740 >CV
700 IF INKEY(73)*(INKEY(2)=0 THEN 870 >XD
710 PA=PA+1:IF PA>50 THEN 1220 >ZH
720 GOTO 660 >VY
730 REM :: VERS LE HAUT :::: >UZ
740 IF Y=9 AND EC=2 THEN 830 >RX
750 ADD=FN PO(X,Y):CALL &A25A,ADD:CALL &A0BB,ADD >OK
760 CALL &A264,SP(16):FOR T=1 TO tt:NEXT:CALL &A28C >UM
770 CALL &A264,SP(17):FOR T=1 TO tt:NEXT:CALL &A28C >UN
780 CALL &A264,SP(17):FOR T=1 TO tt:NEXT:CALL &A28C >DG
790 IF MU=0 THEN SOUND 4,150,35,10,4,4 >UE
800 CALL &A264,SP(16):FOR T=1 TO tt:NEXT:CALL &A28C >NM
810 CALL &A264,SP(15):FOR T=1 TO 50:NEXT:Y=Y-1 >CT
820 IF Y=9 AND EC=2 THEN 830 ELSE 1370 >TU
830 IF M=1 THEN GOSUB 2420 >PA=0:GOTO 1380 >ZT
840 CALL &A1EC:PY(9)=&C63B:CALL &A230,FN PO(X,Y),SP(15) >ZY
850 EC=1:GOTO 1370 >MH
860 REM :: VERS LE BAS :::: >UG
870 IF Y=9 AND EC=1 THEN 960 >UG
880 ADD=FN PO(X,Y):CALL &A25A,ADD:CALL &A0BB,ADD >RB
890 CALL &A295,SP(16):FOR T=1 TO tt:NEXT:CALL &A28C >UU
900 CALL &A295,SP(17):FOR T=1 TO tt:NEXT:CALL &A28C >UL
910 CALL &A295,SP(17):FOR T=1 TO tt:NEXT:CALL &A28C >UM
920 IF MU=0 THEN SOUND 4,150,35,10,4,4 >OB
930 CALL &A295,SP(16):FOR T=1 TO tt:NEXT:CALL &A28C >UN
940 CALL &A295,SP(15):FOR T=1 TO 50:NEXT:Y=Y+1 >NU
950 IF Y=9 AND EC=1 THEN 960 ELSE 1370 >CA
960 IF N=1 THEN GOSUB 2510 >TZ
970 CALL &A21E:PY(9)=&BFFB:CALL &A230,FN PO(X,Y),SP(15) >ZW
980 EC=2:GOTO 660 >LB
990 REM :: A GAUCHE :::: >QE
1000 ADD=FN PO(X,Y):CALL &A0BB,ADD:CALL &A230,ADD-L,SP(16) >CJ
16) >CJ
1010 FOR T=1 TO tt:NEXT >RQ
1020 CALL &A0BB,ADD-1:CALL &A230,ADD-2,SP(17) >ML
1030 FOR T=1 TO tt:NEXT >RT
1040 CALL &A0BB,ADD-2:CALL &A230,ADD-3,SP(17) >MQ
1050 FOR T=1 TO tt:NEXT:IF MU=0 THEN SOUND 4,150,35,10,4,4 >BK
1060 CALL &A0BB,ADD-3:CALL &A230,ADD-4,SP(16) >MU
1070 FOR T=1 TO tt:NEXT >RX
1080 CALL &A0BB,ADD-4:CALL &A230,ADD-5,SP(15) >MX
1090 FOR T=1 TO 50:NEXT:X=X-1:GOTO 1370 >GL
1100 REM :: A DROITE :::: >RF
1110 ADD=FN PO(X,Y):CALL &A0BB,ADD:CALL &A230,ADD+1,SP(16) >CJ
16) >CJ
1120 FOR T=1 TO tt:NEXT >RT
1130 CALL &A0BB,ADD+1:CALL &A230,ADD+2,SP(17) >MJ
1140 FOR T=1 TO tt:NEXT >RV
1150 CALL &A0BB,ADD+2:CALL &A230,ADD+3,SP(17) >MN
1160 FOR T=1 TO tt:NEXT:IF MU=0 THEN SOUND 4,150,35,10,4,4 >BM
1170 CALL &A0BB,ADD+3:CALL &A230,ADD+4,SP(16) >MR
1180 FOR T=1 TO tt:NEXT >RZ
1190 CALL &A0BB,ADD+4:CALL &A230,ADD+5,SP(15) >MV
1200 FOR T=1 TO 50:NEXT:X=X+1:GOTO 1370 >GB
1210 REM :: SUR PLACE :::: >TU
1220 ADD=FN PO(X,Y):CALL &A0BB,ADD:CALL &A230,ADD,SP(16) >AD
1230 FOR T=1 TO tt*2:NEXT >UF
1240 CALL &A0BB,ADD:CALL &A230,ADD,SP(17) >HH
1250 FOR T=1 TO tt*2:NEXT >UH
1260 CALL &A0BB,ADD:CALL &A230,ADD,SP(17) >HK
1270 FOR T=1 TO tt*2:NEXT:IF MU=0 THEN SOUND 4,150,35,10,4,4 >DT
1280 CALL &A0BB,ADD:CALL &A230,ADD,SP(16) >HL
1290 FOR T=1 TO tt:NEXT >RB
1300 CALL &A0BB,ADD:CALL &A230,ADD,SP(15) >HC
1310 PA=0:GOTO 1380 >ME
1320 REM :::::::::::::::::::: >XD
1330 REM : >RE
1340 REM : TESTS DES CASES : >VM
1350 REM : >RG
1360 REM :::::::::::::::::::: >XH
1370 IF CUB=1 THEN 1430 >QH

```

1380 A=FN PK(X,Y)+1:IF a>14 THEN 660 >BP
 1390 ON A GOTO 1400,1410,1420,1420,1460,1600,1670, >HK
 1690,1730,1730,1400,1400
 1400 FIN=1 >MJ
 1410 GOTO 660 >FD
 1420 IF POR<>0 THEN 660 ELSE POR=A-1:X1=X:Y1=Y:CUB=1:EC >UP
 RA=EC:GOTO 660
 1430 IF EC=ECRA THEN CALL &A00D, FN PO(X1,Y1), SP(1) >TG
 1440 POKE &9E5F+(Y*16)+X1,0:SC=SC+5:GOSUB 1840 >QC
 1450 CALL &A00D,&E706,SP(POR+1):CUB=0:GOTO 660 >PU
 1460 IF POR<2 OR POR>4 THEN 660 >XE
 1470 CALL &A0BB, FN PO(X,Y):CALL &A00D, FN PO(X,Y), SP(POR >DB
 +1)
 1480 CALL &A230, FN PO(X,Y), SP(6):CALL &A230, FN PO(X,Y), >FX
 SP(15)
 1490 POKE &9E5F+(Y*16)+X,13:POR:PP=POR-1:POR=0 >RB
 1500 POKE &A01F,&AF:CALL &A00D,&E706,SP(1):POKE &A01F,&AM
 1A
 1510 ON PP GOTO 1520,1530,1540 >WZ
 1520 GJ=GJ-1:IF GJ=0 THEN 1550 ELSE 660 >EE
 1530 GM=GM-1:IF GM=0 THEN 1550 ELSE 660 >EQ
 1540 GR=GR-1:IF GR=0 THEN 1550 ELSE 660 >EH
 1550 IF ec=1 THEN 1580 >PL
 1560 XD=INT(RND*16)+1:YD=INT(RND*8)+1:IF FN PK(XD,YD)<> >NL
 1 THEN 1550
 1570 POKE &9E5F+(YD*16)+XD,8+PP:SOUND 3,20,100,15,0,3:G >GM
 OTO 660
 1580 XD=INT(RND*16)+1:YD=INT(RND*9)+9:IF FN PK(XD,YD)<> >NB
 1 THEN 1580
 1590 GOTO 1570 >NF
 1600 IF ec=1 THEN 1640 >PD
 1610 IF M=1 THEN GOSUB 2420 >UH
 1620 X=INT(RND*16)+1:Y=INT(RND*8)+1:IF FN PK(X,Y)<>1 TH >HL
 EN 1620
 1630 EC=1:py(9)=&C63B:CALL &A1EC:CALL &A230, FN PO(X,Y), >TZ
 SP(15):GOTO 660
 1640 IF N=1 THEN GOSUB 2510 >UM
 1650 X=INT(RND*16)+1:Y=INT(RND*9)+9:IF FN PK(X,Y)<>1 TH >HC
 EN 1650
 1660 EC=2:py(9)=&BFFB:CALL &A21E:CALL &A230, FN PO(X,Y), >TX
 SP(15):GOTO 660
 1670 IF POR<>10 THEN 660 >RX
 1680 gan=1:fin=1:GOTO 660 >UZ
 1690 IF POR<10 OR POR>12 THEN 660 >ZJ
 1700 INK 12,3,24:D7=D7+1:POR=0 >XP
 1710 POKE &A01F,&AF:CALL &A00D,&E706,SP(1) >JJ
 1720 POKE &A01F,&IA:GOTO 2560 >WW
 1730 IF POR<>0 THEN 660 ELSE SOUND 1,50,21,13,3,4 >PY
 1740 POR=A:CALL &A00B, FN PO(X,Y):CALL &A00D, FN PO(X,Y), >FR
 SP(2)
 1750 CALL &A230, FN PO(X,Y), SP(15):POKE &9E5F+(Y*16)+X,1 >YN
 1760 IF POR=10 THEN SS=10 ELSE IF POR=11 THEN SS=12 ELS >ER
 E SS=11
 1770 CALL &A00D,&E706,SP(SS):GOTO 660 >EB
 1780 REM :::::::::::::::::::: >YF
 1790 REM : >TE
 1800 REM : GESTION COMPTEURS : >YM
 1810 REM : >RH

1820 REM :::::::::::::::::::: >YA
 1830 REM :::: SCORE :::: >PY
 1840 IF sc=0 THEN a\$="#0000":ZL=3:YL=25:GOSUB 350:RETUR >BV
 N
 1850 a\$=STR\$(sc):A\$=RIGHT\$(A\$,LEN(A\$)-1) >HT
 1860 IF sc<10 THEN ZL=7:YL=25:GOSUB 350:RETURN >QQ
 1870 IF sc<100 THEN ZL=6:YL=25:GOSUB 350:RETURN >RG
 1880 IF sc<1000 THEN ZL=5:YL=25:GOSUB 350:RETURN >TZ
 1890 ZL=4:YL=24:GOSUB 350:RETURN >GJ
 1900 REM ::: RECORD ::: >QJ
 1910 IF REC=0 THEN a\$="#0000":ZL=34:YL=25:GOSUB 350:RET >DB
 URN
 1920 a\$=STR\$(REC):A\$=RIGHT\$(A\$,LEN(A\$)-1) >JA
 1930 IF REC<10 THEN ZL=36:YL=25:GOSUB 350:RETURN >TA
 1940 IF REC<100 THEN ZL=37:YL=25:GOSUB 350:RETURN >UU
 1950 IF REC<1000 THEN ZL=36:YL=25:GOSUB 350:RETURN >VN
 1960 ZL=35:YL=25:GOSUB 350:RETURN >CZ
 1970 REM ::: VIE ::: >ML
 1980 a\$=STR\$(vie):A\$=RIGHT\$(A\$,LEN(A\$)-1) >JT
 1990 ZL=14:YL=25:GOSUB 350:RETURN >CZ
 2000 REM ::: TAB ::: >MF
 2010 a\$=STR\$(TA):A\$=RIGHT\$(A\$,LEN(A\$)-1) >HE
 2020 IF TA<10 THEN ZL=29:YL=25:GOSUB 350:RETURN >RA
 1 THEN 1550
 2030 ZL=28:YL=25:GOSUB 350:RETURN >CP
 2040 REM :::::::::::::::::::: >XD
 2050 REM : >RE
 2060 REM : FIN DE PARTIE : >TD
 2070 REM : >RG
 2080 REM :::::::::::::::::::: >XH
 2090 ::: GAGNE ::: >LB
 2100 FOR H=>0 TO 3:MU=REMAIN(H):NEXT >DU
 2110 WHILE INKEY\$<>"":WEND:IF gan=0 THEN 2230 >MN
 2120 SOUND 1,0,15,15,10,,15:FOR h=L TO 25:sc=sc+10:GOSU >EJ
 B 1840
 2130 SOUND 1,0,15,15,10,,15:FOR t=1 TO 50:NEXT t,h >QR
 2140 ADD=FN PO(X,Y):CALL &A0BB,ADD:CALL &A230,ADD,SP(16) >AF
)
 2150 FOR T=1 TO 260:NEXT:CALL &A0BB,ADD:CALL &A230,ADD, >FG
 SP(17)
 2160 FOR T=1 TO 260:NEXT:CALL &A0BB,ADD:CALL &A230,ADD, >FH
 SP(17)
 2170 FOR T=1 TO 260:NEXT:CALL &A0BB,ADD >FM
 2180 POKE &A01F,&AF:CALL &A00D,&E706,SP(1):POKE &A01F,&AT
 1A
 2190 FOR t=1 TO 300:NEXT t >UM
 2200 TA=TA+1:IF SC>REC THEN REC=SC:GOSUB 1910 >NJ
 2210 GOTO 430 >EH
 2220 REM ::: PERDU ::: >PW
 2230 BORDER 26:INK 0,26:SOUND 4,1500,50,7,0,2,10 >PH
 2240 SOUND 4,1200,0,0,14,14:SOUND 4,0,20,2,,10 >ME
 2250 FOR h=1000 TO 1500 STEP 100:SOUND 4,h,30,0,15,15:N >AE
 EXT
 2260 ADD=FN PO(X,Y):CALL &A0BB,ADD:CALL &A230,ADD,SP(16) >AJ
)
 2270 FOR T=1 TO 260:NEXT:BORDER 0:INK 0,0 >HR
 2280 CALL &A0BB,ADD:CALL &A230,ADD,SP(17) >HN
 2290 FOR T=1 TO 260:NEXT >RE
 2300 CALL &A0BB,ADD:CALL &A230,ADD,SP(17) >HF

2310 FOR T=1 TO 260:NEXT >RX
 2320 CALL &A0BB,ADD:POKE &A01F,&AF:CALL &A00D,&E706,SP(>VR
 L):POKE &A01F,&IA
 2330 vie=vie-1:GOSUB 1900:IF vie=0 THEN 2350 >LX
 2340 FOR T=1 TO 260:NEXT T:ta=ta-1:GOTO 2200 >MB
 GOTO 2200
 2360 REM :::::::::::::::::::: >YA
 2370 REM : >RK
 2380 REM : S/PROG DIVERS : >UU
 2390 REM : >TB
 2400 REM :::::::::::::::::::: >XD
 2410 REM ::: MONSTRE TAB 1 ::: >WL
 2420 POKE ADX3,1:POKE ADX4,1:POKE ADX5,1 >HD
 2430 X3=INT(RND*16)+1:Y3=INT(RND*8)+1:IF FN PK(X3,Y3)<> >MP
 1 THEN 2420
 2440 ADX3=&9E5F+(Y3*16)+X3:POKE ADX3,L2 >GK
 2450 X4=INT(RND*16)+1:Y4=INT(RND*8)+1:IF FN PK(X4,Y4)<> >MZ
 1 THEN 2450
 2460 ADX4=&9E5F+(Y4*16)+X4:POKE ADX4,L2 >GR
 2470 X5=INT(RND*16)+1:Y5=INT(RND*8)+1:IF FN PK(X5,Y5)<> >MH
 1 THEN 2470
 2480 ADX5=&9E5F+(Y5*16)+X5:POKE ADX5,L2 >GY
 2490 RETURN >FJ
 2500 REM ::: MONSTRE TAB 2 ::: >WM
 2510 COUP=0 >ZB
 2520 XB=INT(RND*16)+1:YB=INT(RND*9)+9:COUP=COUP+1:IF CO >AB
 UP>10 THEN RETURN
 2530 IF FN PK(XB,YB)<>1 THEN 2510 >ZX
 2540 POKE &9E5F+(YB*16)+XB,13:RETURN >EE
 2550 REM ::: MUSIQUE ::: >RT
 2560 IF D7=1 THEN EVERY 15,1 GOSUB 2610:tt=tt-10:GOTO 6 >AE
 60
 2570 IF D7=2 THEN EVERY 60,3 GOSUB 2600:tt=tt-15:GOTO 6 >AN
 60
 2580 IF D7=3 THEN RESTORE 2640:EVERY 15,2 GOSUB 2620:PO >MC
 R=100:MU=1:tt=tt-15:GOTO 660
 2590 GOTO 660 >GD
 2600 DI:SOUND 2,0,15,15,1,0,1:SOUND 2,0,15,15,1,0,5:E1: >GZ
 RETURN
 2610 DI:SOUND 2,0,15,15,1,0,5:E1:RETURN >HJ
 2620 DI:READ s:IF s=0 THEN E1:RETURN ELSE IF s=-1 THEN >NY
 RESTORE 2640:GOTO 2620 ELSE 2630
 2630 SOUND 1,s,30,14,0,2:SOUND 4,s*2,30,15,0,2:E1:RETUR >BH
 N
 2640 DATA 451,0,379,0,301,0,0,451,379,0,301,301,451,0,3 >BK
 79,0,301,0,0,451,379,0,301,301,338,0,284,0,225,0,0,338,
 284,0,225,225,338,0,284,0,225,0,0,338,284,0,225,225
 2650 DATA 451,0,379,0,301,0,0,451,379,0,301,301,451,0,3 >KJ
 79,0,301,0,0,451,379,0,301,301,338,0,301,338,338,379,0,
 451,-1
 2660 REM :::::::::::::::::::: >YD
 2670 REM : >TC
 2680 REM :PARAMETRES TABLEAUX: >AP
 2690 REM : >TE
 2700 REM :::::::::::::::::::: >XG
 2710 DATA 3,2,2,6,6,0,0 >PV
 2720 DATA 3,4,3,6,3,0,0 >PW
 2730 DATA 2,2,2,2,2,0,0 >PM

2740 DATA 4,4,2,6,6,1,1 >PD
 2750 DATA 6,6,2,6,6,0,1 >PH
 2760 DATA 4,4,3,2,2,0,0 >PW
 2770 DATA 3,3,2,2,2,0,1 >PV
 2780 DATA 3,3,2,5,6,1,1 >PE
 2790 DATA 3,3,2,4,6,1,0 >PD
 2800 DATA 2,2,2,8,2,1,1 >PU
 2810 DATA 3,2,3,7,2,0,0 >PU
 2820 DATA 2,1,3,2,7,1,1 >PV
 2830 REM :::::::::::::::::::: >YC
 2840 REM : PRESENTATION : >TB
 2850 REM : >UM
 2860 REM : >TD
 2870 REM :::::::::::::::::::: >YG
 2880 INK 3,0,16:INK 10,16,0:SPEED INK 25,25:RESTORE 264 >BN
 0:EVERY 15,2 GOSUB 2620
 2890 EVERY 15,1 GOSUB 2610:EVERY 60,3 GOSUB 2600 >PE
 2900 ZX=0:ZY=0:E8=10:E9=3:GOSUB 2910:ZX=188:ZY=0:E8=3:E >TB
 9=18:GOSUB 2910:ZX=376:ZY=0:E8=10:E9=3:GOSUB 2910:GOTO
 3010
 2910 ORIGIN ZX,ZY:FOR h=60 TO 76 STEP 2:PLOT 100,h,4:DR >QV
 AW 164,h:NEXT
 2920 FOR h=84 TO 160 STEP 2:PLOT h,156:DRAW 164,60:NEXT
 2930 FOR h=160 TO 164 STEP 2:PLOT h,156:DRAW h,76:NEXT: >QU
 PLOT 126,156,14:DRAW 100,76:DRAW 164,76:DRAW 138,156:PL
 OT 96,156:DRAW 132,76:DRAW 168,156
 2940 FOR h=80 TO 164 STEP 4:x=INT(RND*8):PLOT h,166,1:D >LM
 RAW h,146+x:NEXT:PLOT 80,169,14:DRAW 184,160
 2950 PLOT 80,166:PLOT 184,166:PLOT 80,158:PLOT 184,158 >XG
 2960 FOR G=146 TO 178 STEP 16:XT=G:YT=176:P=8:EN=E8:GOSUB >TH
 UB 3000:XT=G:YT=200:P=8:EN=E9:GOSUB 3000:NEXT
 2970 XT=198:YT=170:P=8:EN=E8:GOSUB 3000:XT=198:YT=194:P >CR
 =8:EN=E9:GOSUB 3000
 2980 FOR G=88 TO 120 STEP 16:XT=G:YT=176:P=8:EN=E9:GOSU >QP
 B 3000:XT=G:YT=200:P=8:EN=E8:GOSUB 3000:NEXT
 2990 XT=68:YT=170:P=8:EN=E9:GOSUB 3000:XT=68:YT=194:P=8 >AR
 :EN=E8:GOSUB 3000
 3000 PLOT 700,700,EN=P2=P:ORIGIN XT+ZX,YT+ZY:FOR H=0 >AJ
 TO P*0.71 STEP 2:J=SQR(P2-H*H):MOVE H,J:DRAW H,-J:MOVE
 J,H:DRAW J,-H:MOVE -H,J:DRAW -H,-J:MOVE -J,H:DRAW -J,-H
 :NEXT:ORIGIN 0,0:RETURN
 3010 B\$="CHARLIE":FOR H=1 TO 7:A\$=MID\$(B\$,H,1):INK 8,0 >WB
 3020 PEN 8:LOCATE 1,23:PRINT A\$:X=H*48+30:Y=1:E1=2:E2=0 >YN
 :GOSUB 3060:NEXT H
 3030 B\$="BONDSOUND":FOR H=1 TO 9:A\$=MID\$(B\$,H,1) >QN
 3040 PEN 8:LOCATE 1,23:PRINT A\$:X=H*48+90:Y=2:E1=15:E2= >ZV
 4:GOSUB 3060:NEXT H
 3050 LOCATE 1,23:PRINT " " :INK 8,0:GOTO 3100 >KG
 3060 FOR I=32 TO 48 STEP 2:FOR L=1 TO 32 STEP 4 >KN
 3070 A=TEST(L,I):IF A=0 THEN 3090 >ZV
 3080 PLOT X-4+L,310-(Y*48-(I*2)),E1:PLOT X-4+L,312-(Y*4 >LT
 B-(I*2)),E2
 3090 NEXT L,I:RETURN >QD
 3100 FOR H=1 TO 5:MOVE 30,H:DRAW 610,H,4:NEXT:FOR H=6 T >QT
 0 20:MOVE 30,H:DRAW 610,H,8:NEXT
 3110 FOR H=20 TO 22:MOVE 30,H:DRAW 610,H,1:NEXT:FOR H=1 >RZ
 TO 8:MOVE H,26:DRAW H,374,4:NEXT
 3120 FOR H=8 TO 20:MOVE H,28:DRAW H,374,8:NEXT:FOR H=20 >UD

DATACHAR.BAS

5 MEMORY &881F:MODE 1
7 LOCATE 1,10:PRINT " DATAS DE LA LIGNE CORRECTS":PEN 2

```

10 A=1&8820:F=&A2C0:L=100:WHILE A<=F:FOR A=A TO A+15:READ C$:K=VAL(C$):S=S+K+65536*(S+K>32767):IF A<=F THEN POKE A,K
20 NEXT:READ D$:T=VAL("&"&D$):(IF T>5 THEN PRINT CHR$(7);"Erreur ligne";L:END ELSE L=L+5
25 LOCATE 20,10:PRINT L-5:WEND
100 DATA 20,5E,3C,5A,B4,B4,1E,F0,2D,7B,F0,2D,F0,B4,5A,B4,B8D0

```


ECRITURE ARABE

Michel la ROSA

Valable pour CPC 464 - 664 - 6128

Voici un petit programme utilitaire qui pourrait intéresser tous nos amis de langue et de culture arabe, mais aussi tous ceux qui souhaitent comme moi s'y initier.

J e l'ai appelé Gerbert, en hommage au premier lettré français qui est allé au Xème siècle renouveler notre science balbutiante en Catalogne, haut lieu de culture arabe à l'époque (et dans le Midi, c'étaient souvent les juifs, polyglottes de tradition, qui apprenaient l'arabe aux français !). Il faut savoir que Gerbert, après avoir contribué à mettre un prince illétré (comme les autres !) sur le trône de France (Hugues Capet), a fait de son fils Robert (et de plusieurs princes de sa génération) un lettré accompli. Enfin c'est lui le pape de l'An Mil. Voilà pour l'évocation.

Ce programme utilise le clavier français pour transcrire la calligraphie arabe, en suivant un code représenté dans la figure 1. En l'utilisant, vous découvrirez quelques signes supplémentaires, par exemple les parenthèses, transformables en guillemets de début et de fin de citation, en tapant COPY avant le signe parenthèse.

Le curseur se déplace au moyen des 4 flèches de direction, de ESPA-

CE et de DEL. Attention ! Il efface tout sur son passage !

CLR permet de recommencer la page à 0.

TAB avance le curseur au milieu de la page, s'il était en deçà.

On peut noter la façon d'obtenir les 4 formes de chaque lettre arabe suivant sa place dans le mot :

- Deux fois COPY puis la lettre si elle est en position isolée (1)

- COPY puis la lettre en position finale (2)

- La lettre seule si elle est en position (3)

- SHIFT puis la lettre en position initiale (4)

On peut aussi noter :

- Les 4 positions du hamza et le moyen de les obtenir, quelques variantes calligraphiques courantes dans l'arabe écrit, les signes en surimpression (hamza, shadda, fatha, damma, kasra), l'utilisation de la barre de soulignement pour prolonger la ligne d'écriture dans le cours d'un mot si on le désire.

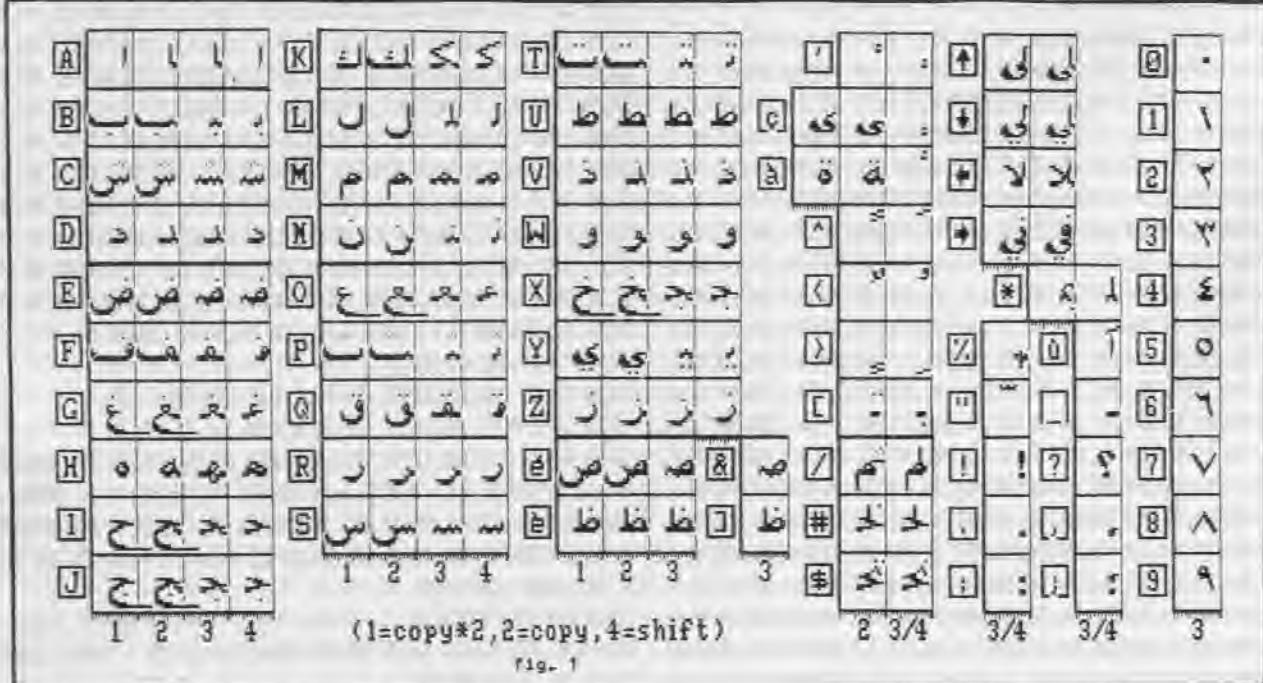
Pour ceux que cela intéresse, il est possible de modifier l'allure des

lettres, que j'ai choisies en pleins et en déliés, parce qu'elle passe mieux à l'écran et qu'on la trouve souvent ainsi dans la presse arabe (les titres notamment). Il est également possible de modifier les datas pour obtenir des caractères complémentaires.

J'ai passé plus de 3 mois à ce travail, mais j'ai conscience qu'il peut être amélioré : le choix du MODE 2 donne une plus belle lisibilité à l'écran, mais diminue la clarté à l'impression (en tapant CONTROL puis COPY).

Enfin ce programme est bilingue : on peut passer du français à l'arabe et réciproquement en tapant COPY puis RETURN. Vous verrez que l'usage du tabulateur TAB permet, au choix, de placer les textes arabes à droite et français à gauche (le plus pratique, surtout si le texte est long) ou l'inverse (vocabulaire ou conjugaisons et leur traduction, par exemple).

Taper et sauver les fichiers ARABE1.PRG et ARABE2.PRG, qui génèrent les fichiers ARABE1.BIN et ARABE2.BIN, puis taper et lancer le programme GERBERT.



```

18 ' #####          >LA 6,1744
15 '             >LF
28 ' ECRITURE ARABE >LB
25 '             >LG
38 ' par           >LC
35 '             >LH
48 ' Michel LA ROSA >LD
45 '             >LJ
58 ' ->55.74.26.12 >LE
55 '             >LK
68 ' Juin 1991    >LF
65 '             >MA
78 ' #####          >LG
88 '             >LH
98 ' Préparation Imprimante >LJ
100 '             >RB
110 MODE 2:MEMORY &9AE2 >RY
120 ad=&9AE3:nl=170:RESTORE 170 >Z2
130 FOR a=1 TO 22:b=0:FOR c=1 TO 16:READ a$ >HC
140 POKE ad,VAL("&"&a$):b=b+PEEK(ad):ad=ad+1 >MW
150 NEXT:READ e:IF e<>b THEN PRINT"Erreur DATA ligne":nl >MR
:END
160 nl=nl+10:NEXT:CALL HIMEM+1 >ZJ
170 DATA 21,E1,E9,22,30,00,F7,E8,21,49,01,19,4E,23,46,7 >DD
9,1491
180 DATA B0,28,15,E5,60,69,19,E5,4E,23,46,60,69,19,44,4 >DP
D,1475
190 DATA E1,71,23,70,E1,23,18,E4,21,49,00,06,81,0E,00,1 >DY
1,1269
200 DATA 2F,B0,CD,D7,BC,C9,C5,D5,E5,F5,09,CD,1E,BB,2 >FD
8,2273
210 DATA 0A,3E,17,CD,1E,BB,28,03,CD,59,00,F1,E1,D1,C1,C >EM
9,1923
220 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0 >YT
0,0
230 DATA DD,21,52,00,DD,36,01,00,DD,36,02,00,DD,36,03,9 >DP
0,1311
240 DATA DD,36,04,01,DD,36,05,00,DD,36,00,00,3E,1B,CD,C >EG
C,1333
250 DATA B0,05,E5,CD,2B,BD,3E,41,CD,2B,BD,3E,06,CD,2B,B >FZ
D,2135
260 DATA CD,2E,BD,38,FB,3E,1B,CD,2B,BD,3E,4B,CD,2B,BD,3 >GY
E,1909
270 DATA 40,CD,2B,BD,3E,01,CD,2B,BD,DD,36,00,00,DD,36,0 >EY
6,1557
280 DATA 00,DD,CB,00,26,DD,5E,01,DD,56,02,DD,6E,03,DD,6 >EX
6,00
600 POKE &45A0,ecr-256+[NT(ecr/256):POKE &45A1,INT(ecr/ >EA

```


Digitized by srujanika@gmail.com

malum pungens (L.) L.

ABONNEZ-VOUS !

11 numéros
210 F

43 F
D'ECONOMIE



5% de remise
sur les produits
du catalogue
SORACOM



OUI, je désire m'abonner à CPC *infos* pour
un an au prix de 210 F (11 numéros)

Nom : _____ Prénom : _____

Adresse : _____

Code postal : _____ Ville : _____

Date : _____ Signature
obligatoire

ATTENTION !
SEUL CE BULLETIN
D'ABONNEMENT
EST VALABLE



CPC N°

- Les abonnements ne sont pas rétroactifs
- Envoi par avion + 165 F

je désire payer avec une carte bancaire



Date d'expiration



Envoyez votre bulletin accompagné du règlement à : Editions SORACOM - Service abonnement - BP 88 - 35170 BRUZ

**LE SPECIALISTE DE
L'EDUCATIF**

**notre CATALOGUE
des éducatifs**



**Les détails des logiciels
les tarifs**

catalogue gratuit

Je désire recevoir le catalogue des éducatifs, possédant
un ordinateur(s) de
marque(s) _____,
de type(s) _____,
à l'adresse ci-contre Nom: _____ Prénom: _____

Adresse: _____
Code postal: _____ Ville: _____

A envoyer à SORACOM Editions - BP 88 - 35170 BRUZ
Joindre deux timbres à 2,30 F

SORACOM éditions

Media Box Cassettes

Vidéo (L)

222x135
x348 mm
Réf. 310.540.0

155 F + port

pour 9 cassettes
VHS, V2000, Betamax



Media Box Mini Cassettes

148x91x348
Réf. 310.503.3

95 F + port

pour 16 mini
cassettes



Media Box Disquettes 3"-3 1/2

221x135x348 mm
Réf. 310.506.4

175 F + port

pour 150 disquettes



Media Box Cassettes Vidéo VHS-C

148x91x348 mm
Réf. 310.505.7

102 F + port

pour 8 cassettes

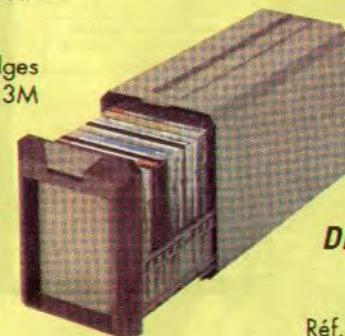


Media Box Data Cartridge

222x135x348 mm
Réf. 310.518.7

160 F + port

pour 11 Data Cartridges
type 3M



Media Box Compact Disc Multi

148x177
x348 mm
Réf. 100.525.0

pour 23 CD simples
ou 11 CD doubles

Media Box Photo

222x135
x348 mm
Réf. 310.501.9

155 F + port

pour 140 tirages
format maximum 13x18 plus



Module serrure

60 F + port

(bien indiquer la boîte)



Port + 25 F par article
port + 40 F par article pour l'étranger

Media Box Disquettes 5"1/4

182x178x348 mm
Réf. 310.501.1

180 F + port

pour 70 disquettes, livré
avec séparations et index



Media Box Compact Disc

148x135x348 mm
Réf. 310.502.6

125 F + port

pour 13 compacts discs simples

Nouveau ! les mêmes en noir



Mini cassettes
Réf: 310 . 620 . 7
95 F + PORT



Vidéo VHS (L)
Réf: 310 . 623 . 8
160 F + PORT



Compact Disc
réf: 310 . 621 . 4
125 F + PORT



C D Multi
Réf: 310 . 622 . 1
190 F + PORT