

**Q**  
numéro 19  
JS

Programmez votre CPC ou CPC+ avec

**QUASAR**

— ACTUS —

- ~Soundplayers
- ~projet ANA
- ~CPCNG

— DOSSIER —

- ~programmation  
des jeux

— ASSEMBLEUR —

- ~soft: déplombage
- ~hard: le Gate  
Array
- ~coding:  
optimisation  
des tables

— ELECTRONIQUE —

- ~montez un  
lecteur 3"1/2  
à la place  
du 3" interne





numéro  
19

# QUASAR

SOMMAIRE ————— PAGES

Editorial & Sommaire ..... 1

Actus ..... 2-4

Petites Annonces ..... 5

La Rubrique X ..... 6

Les Histoires Perpendiculaires .... 7

Assembleur : software ..... 8-10

Assembleur : hardware ..... 11-15

Assembleur : coding ..... 16-17

Dossier ..... 18-19

Electronique ..... 20-22

Helps ..... 23-25

## EDITORIAL

Non, vous ne rêvez pas ! C'est bel et bien le numéro 19 de Quasar CPC que vous tenez entre vos mains ! Que de retard ! Enfin, malgré tous les problèmes auxquels nous avons pu être confrontés cette fois-ci encore, nous sommes là !

Que dire, en dépit de tout le mal que certains peuvent en dire, notre petit CPC continue paisiblement sa vie grâce à la scène bien sûr, mais aussi par le biais de toute cette communauté qui se réforme peu à peu via l'Internet. Néanmoins, tout n'est pas rose. Certains acteurs nous quittent, le Cocoon System cesse ses activités en tant qu'association...

Enfin, ceci ne doit pas nous faire oublier les nombreux projets qui sont en cours ainsi que le grand retour des anciens, et même l'arrivée de nouveaux ! Qu'on le rejette ou qu'on s'y investisse au détriment du CPC réel, il n'en demeure pas moins que sans l'Internet nous n'aurions pas su attirer autant de monde. Un des événements importants est sans doute la mise en place du canal IRC CPC francophone grâce auxquels de nombreux CPCistes d'hier et de demain (!) peuvent entrer en contact avec la scène... pour les accueillir et les aiguiller, essayez de passer de temps à autre sur [irc.iftouch.net](http://irc.iftouch.net), port 6667, canal #CPC avec votre client IRC favoris !

Par ailleurs, dans un registre plus convivial que la froideur d'une discussion face à un moniteur d'ordinateur, une grande aventure se prépare pour cet été : La Klassentreffen ! Il va certainement s'agir du plus gros meeting CPC européen jamais organisé puisque près de 80 personnes y sont déjà attendues. Celui-ci se déroulera à Erlangen (en Allemagne) les 8, 9 et 10 Juin 2001. Pour plus de détails rendez-vous sur <http://www.kangaroo.cmo.de/klassentreffen>, le site officiel, ou écrivez-nous.

Avant de vous laisser entamer la lecture de cette 19ème édition de Quasar CPC, je tiens à vous préciser que vous allez découvrir un numéro pratiquement vide d'actus. En effet, comme nous ne pouvons pas raisonnablement vous proposer un fanzine de 50 pages, il a fallu faire des choix, et nous avons préféré donner la priorité aux sujets techniques. De plus, si vous êtes friands d'infos et de potins mondains, il y a l'excellent CPC Match... heu pardon, Another World, que toute la rédac' vous conseille vivement de lire !

Je vais finir avec une petite pensée pour nos expatriés, Tony (en Bulgarie) et SNN (aux Etats-Unis), qui continuent malgré tout à s'investir de leur mieux dans le CPC. Allez, trêve de bavardages : bonne lecture et rendez-vous dans le numéro 20 dès Août 2001 !

— OffseT —

REDACTEUR  
OffseT

### EDITORIAL ET SOMMAIRE

REDACTEUR  
OffseT

— QUASAR CPC numéro 19 — Diffusez-moi — Diffusez-moi — QUASAR CPC numéro 19 —

# ACTUS

Que de choses à dire en si peu de pages ! Allez hop, on commence sans tarder par une petite mise au point sur les Soundplayer... je laisse donc la parole à Zik.

## Dessine-moi une Soundplayer

La Soundplayer est une petite interface bien connue et répandue. Elle se branche sur le port imprimante du CPC ou CPC+ et fournit une sortie audio mono 8 bits. Elle est, dans sa fonction, équivalente à la Digiblaster, mais avec néanmoins une meilleure qualité sonore due au schéma électronique différent.

La Soundplayer+ ajoute un connecteur pour le réseau Virtual Net 96, elle cumule donc les deux fonctions. A noter que les deux fonctions sont incompatibles : c'est soit le son, soit le réseau.

Je suis en train de mettre au point la version suivante qui apporte les mêmes services mais avec un son encore meilleur et une qualité beaucoup plus uniforme d'une interface à l'autre. Le prix est légèrement augmenté et les dimensions de l'interface diminuent un peu. Cette interface portera le nom de SoundplayerNG (NDLR : et il s'agira vraisemblablement de la dernière version qui sera mise au point).



Passons maintenant à la longue histoire de la Soundplayer2. Cette interface a comme vocation première de permettre l'échantillonnage audio sur CPC. Elle se connecte sur le port expansion du CPC. C'est au départ une modeste carte faite pour mon usage propre. Je l'ai présentée à différents meetings où elle a intéressé plusieurs personnes, et j'ai donc accepté des commandes. Celles-ci n'ont toujours pas été satisfaites à ce jour (excuse-moi Shap !). Je n'ai évidemment pas encaissé les chèques que certains m'avaient donnés, mais je garde les noms des personnes intéressées.

Pourquoi trainer autant ? Eh bien, mis à part le peu de temps que j'ai à lui consacrer, cette carte est beaucoup plus complexe à réaliser que la Soundplayer et surtout, je ne cesse de l'améliorer. Ce que je veux éviter, c'est de distribuer une interface qui sera ridicule et obsolète face à la version suivante pour un prix comparable. Il est vrai qu'en attendant rien ne

sort...

Donc, j'ai maintenant figé les caractéristiques de la Soundplayer2 (un nom un peu impropre), elles seront à peu près les suivantes :

- entrée et sortie stéréo ou mono échantillonnée en 8 bits à une fréquence jusqu'à 50kHz,
- possibilité de générer des interruptions,
- une entrée de mixage stéréo pour le son du CPC,
- une entrée de mixage mono pour le son d'une Soundplayer éventuellement branchée sur le port imprimante.

Grâce aux entrées de mixage, toutes les sources audio seront regroupées sur une seule sortie. Ces caractéristiques sont nettement supérieures à celles de la version initiale mais malgré cela le prix sera du même ordre. Ceci grâce à l'utilisation d'un circuit spécialisé (inutile de réinventer la roue). Encore faut-il réussir à se le procurer...

Je vous le dis tout de suite, n'espérez pas voir cette interface avant l'été 2001 !

NDLR : attention, la Soundplayer2 n'est pas compatible Digiblaster ! Il s'agit d'une carte son entièrement nouvelle, si vous voulez utiliser les logiciels exploitant les avantages de la Digiblaster il vous faut donc toujours une Soundplayer (la NG par exemple). Toutefois, lors de sa sortie, la Soundplayer2 sera bien entendu distribuée avec les logiciels (et le device ANA) permettant de l'exploiter.

Merci à Zik pour tous ces éclaircissements ! Et comme nous sommes dans le hardware, je vous propose une brève présentation du CPCNG.

## Le CPCNG

Il s'agit en fait d'un projet un peu fou qui consiste tout bonnement à créer une machine entièrement nouvelle ! Une équipe internationale assez conséquente travaille sur ce projet depuis maintenant plusieurs mois... Mais que ce cache-t-il vraiment derrière ces cinq lettres... Christophe GUELF, l'initiateur du projet, nous en dit quelques mots :

"Le premier prototype contiendra un CRT6845 avec 128k de RAM vidéo gérant les modes CPC ou plus. Les I/O du CPC seront interceptés pour correspondre à ceux du NG. En mode CPC, les 128k de base seront de la RAM

vidéo, le FPGA aura les fonctions vidéos du Gate Array et gèrera l'accès à la mémoire. Il y aura des interfaces clavier et souris ainsi qu'un contrôleur de disque dur IDE. Il faudra prévoir une alimentation PC. Peu d'infos sur l'OS qui aura une GUI. On retrouvera un langage semicompilé Basic capable d'exécuter des codes d'autres Basic et pouvant être intégré à l'OS. Rien n'est décidé pour le son. On prévoit à la fin un GPU additionnel pour de meilleurs graphismes (haute résolution, blitter, bitplanes, copper list), ainsi qu'une sortie TV, de l'USB, un système de ROMs, une alimentation." Christophe GUELF, 6209 Résidence Universitaire du Saulcy, 57000 METZ. <http://cpcng.free.fr>.

Merci à Christophe pour ces quelques informations... Mais ce projet ne doit pas nous faire oublier les interfaces en route pour CPC.

## Les interfaces pour CPC

Outre la Soundplayer2 dont nous venons de parler, de nombreux projets matériels pour CPC sont en cours. Bien sûr il y a toujours la fabuleuse CPC-ISAcard de Siou ([siou.free.fr](http://siou.free.fr)) mais également les Ramcards, Memcard, et futuramente IDEcard de Ram7... sans compter tous les autres projets qui restent un peu dans l'ombre. Et qui dit superbes interfaces, dit nouveau système d'exploitation ! Eh oui, ANA est toujours en course !

## ANA Ouroboros

L'architecture du futur OS de nos bons vieux CPC est à présent clairement posée. Comme prévu, ANA est une mise à jour logicielle du système d'exploitation du CPC qui s'appuie entièrement sur le principe du Firmware Amstrad ; et qui n'introduira donc aucune incompatibilité. Elle est multitâche et fonctionne entièrement en 32 bits linéaires pour la gestion mémoire. Une structure modulaire des "ressources mémoire" permettra aisément d'ajouter à peu près n'importe quel type de RAM dans l'adressage d'ANA ; par défaut, les ressources correspondantes aux ROMs, à la mémoire centrale, à la mémoire bank (dont Memcard), à la Multiface II et aux cartouches des CPC+ seront incluses.

Ce choix de la modularité vaut aussi pour la gestion des périphériques via des "devices" ce que facilitera le développement et la gestion homogène des drivers associés à toutes les cartes d'extension ainsi qu'au hard interne du CPC ou du CPC+. Une fois le système mis en place, des modules seront peu à peu mis au point pour remplacer les routines Amstrad par nos propres vecteurs système pour ne garder au final que le système de démarrage du firmware.

Pratiquement, ANA pourra tourner sur n'importe quel CPC équipé de 128ko de RAM (voire 64ko), toutefois,

l'utilisation de la version ROM, qui se présentera sous la forme d'une ROM de premier plan (donc bootable), sera fortement conseillée.

Du point de vue de l'utilisateur, une fois ANA lancée, elle se substitue complètement au Basic en offrant à la place un système de shell tel que le font déjà des logiciels comme Protex ou Rodos.

Enfin, au niveau graphique, si ANA gèrera la mémoire vidéo, avec notamment la notion d'écrans indépendants, elle n'inclura pas réellement d'interface graphique... BOB, une surcouche offrant des classes de fenêtres et de gadgets sera développé dans un second temps.

En outre, du fait de la gestion déportée de tout ce qui n'est pas rigoureusement l'OS, il est envisageable de porter ANA sur n'importe quel système à base de 280, le CPCNG par exemple !

En ce qui concerne le développement, il est ouvert à toutes les bonnes volontés... un dossier détaillé sur ANA et BOB va être rédigé et diffusé auprès de la scène dans les semaines à venir.

Contact développement : Offset (noyau, ordonnanceur et devices), Madram (noyau, ressources mémoire).

## Les fanzines

Le désormais célèbre Another World en est à son 23ème numéro. Rien à redire, il est toujours aussi complet pour tout ce qui est des actus de la scène CPC. Je ne peux que vous le conseiller pour absolument tout savoir sur tout, de l'excellent boulot. Je vous renvoie à l'adresse de Hicks dans la Rubrique X pour tout commander !

Enfin, selon les dires de Madram, Amslive ne devrait plus trop tarder à reprendre du service... nous l'espérons tous !

## Brèves et potins

Tout d'abord, pour les internautes, voici quelques adresses sympas à visiter :

<http://www.inicon.de> (les concepteurs de la RamRomBox)  
<http://phenixinformatique.free.fr> (le portail CPC)  
<http://grees.nps.free.fr> (le site du NPS)  
<http://briconique.multimania.com> (pour les bidouilleurs)  
<http://gfxzone.planet-d.net> (bientôt une section CPC)

Et puis pour finir quelque rumeurs... Bientôt du nouveau chez Power System, ALN nous prépare quelque chose, mais chut, c'est top secret ! De même, Ast et Overflow nous préparent un retour sur CPC avec une (des ?) demo pour CPC+... la place ne manque, désolé pour les brèves et infos que j'ai du retirer : lisez Another World !!!

**CROCO CHANEL N 2**

**DU VENDREDI 27 NOVEMBRE AU LUNDI 31 OCTOBRE 2000 S'EST DEROULEE LA DEUXIEME EDITION DU CROCO CHANEL ORGANISE PAR MORTEL ET OVERLANDERS A CATENOY (DANS L'OISE). VOICI POUR VOUS QUELQUES IMPRESSIONS PRISES SUR LE TAS... ON N'EST PAS CHERS PAYS MAIS QU'EST-CE-QU'ON RIGOLE !**

Une Super Ambiance, on y apprend énormément, j'veux pas m'arrêter j'veux encore faire mumuse. GEGERE

J'ai beaucoup aimé, surtout que cette année il y avait une baignoire. A noter qu'il n'y a eu aucune coupure de courant : services à la hausse, mériterait 2 étoiles AU MICHELIN. RoUdOuDu

Un meeting super sympa, je n'ai jamais autant ri, de plus d'avoir rencontré des personnes jusqu'alors inconnues fut plus que sympathique (j'aime bien les "sympa" c'est un bon mot). Vivement l'année prochaine pour un autre Croco tout aussi délirant.

Big Daddy cool IRON

Hougabounga ! Coool Meeting, hyper sympa, qqs gens bruyant... Mais C po grave...

Rald

Et bien il semblerait que cette seconde édition du CROCO soit encore une fois une réussite complète : participants nombreux et super cools (mais aussi a l'humour de plus en plus crade...), organisation sans failles, ambiance du tonnerre, salle spacieuse...

Un grand bravo a Candy, Shap, Winny et Fafa !

Targhan

C'est mon premier meeting et je dois dire que c'est vraiment génial, j'ai commencé à apprendre l'asm grâce à IRON et tous les gentils CPCiens qui m'ont aidé et je les remercie. At au prochain meeting.

Plissken

Hé bien comme à l'habitude des Croco Chanel une bonne ambiance, une cuisine d'exception... Et des moments mémorables... Si seulement ça serait tous les week ends!!!

Blood

Coool, super meeting bien poilant, super organisation bravo. Seul regret les horaires des repas quelques peu perturbé sans doute le décalage horaire. Félicitations à la seule présence féminine, Winnie ce fut un service digne d'un grand chef !!!

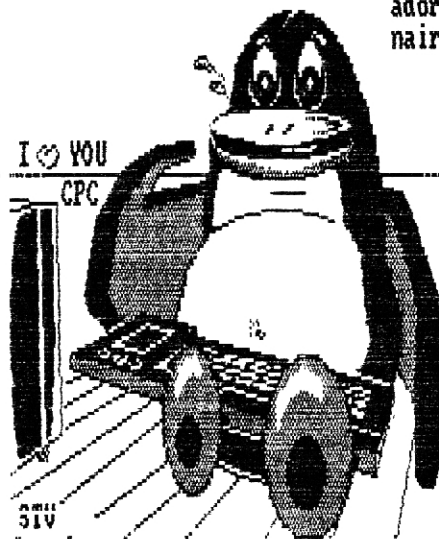
RAM7

Venu retenter l'expérience, j'ai adoré l'ambiance (même si d'ordinaire ce sont les idoles qu'on adore). J'ai particulièrement apprécié le contraste entre les moments délires et les discussions plus techniques. Il apparait que les CPCistes sont des personnes pétries d'humour et d'humanité.

Vivement le prochain !  
Yves.

Encore une fois cela a été un très bon meeting, on a bien ri, vivement l'an prochain !

Winny



"I love You CPC" STU  
Vivement la prochaine orgie...

J'avais bien besoin de me changer les idées. Mission réussie grâce à un super meeting avec une ambiance du tonnerre ! Et pi on a bien mangé ! Fameux hé...

Ker

Hello! Des moments mémorables comme dans chaque meeting, mais cette fois-ci, on a dépassé la dose prescrite en délires...

Eliot!

Oooooohhhh !!!! et bien ce fut un bien agréable meeting et je ne dis pas cela parce que je fais parti des organisateurs... FDC rullezzzzz, la lune est beaucoup plus utile que le soleil car elle éclaire la nuit.

ShAp/OvL!

Une ambiance pas mal, même très bien. La bouffe est excellente.

Romain

Un bon bain de CPC Scène avant 2 ans de folie en Bulgarie... avec mon CPC bien sûr ! En bref, un très bon Croco Deal...

Tony

RAMDE  
JRAMB  
GOSERABEUS  
RAMAJERRY  
RAMONE  
ERISIRAM  
HYDRAM  
RAM7  
RAMLEID  
MADRAM

RAMBIBI  
SHAMPERRAM  
RAMDOU  
STUJERRAM  
RAMBIB  
RAMINNY  
HYPERBIB  
TRERRAM  
RAMBOUR  
RAMBOUR  
RAMBOUR

J'ai lu avec attention un article (je ne nommerai pas le journal) sur les émulateurs pour anciennes machines genre Amstrad CPC. Il se trouve que je possède deux Amstrad, un CPC464 + lecteur de disquettes et un CPC6128, de nombreux logiciels, ainsi qu'une collection de magazines de l'époque. Ne sachant que faire de ce matériel qui m'encombre maintenant, je souhaiterais l'échanger contre un logiciel pour PC (faire proposition). Mon email : [janay@oreka.fr](mailto:janay@oreka.fr)

J'ai retrouvé un Amstrad CPC6128 et je voudrais l'utiliser pour mes enfants. Malheureusement, je n'ai plus de disquettes CP/M pour lancer l'utilisation. Quelqu'un pourrait-il m'en envoyer une ? Merci de me contacter sur [freginator@aol.com](mailto:freginator@aol.com)

Bonjour, je cherche un manuel de l'utilisateur pour un Amstrad 6128 plus. Merci d'avance. [A.Lafdic@Brinks.fr](mailto:A.Lafdic@Brinks.fr)

Je voudrais savoir s'il y a un site qui aurait la liste des commandes, ou si quelqu'un pourrait m'envoyer la liste des commandes du Basic du CPC6128. [letof@cybernaute.fr](mailto:letof@cybernaute.fr)

J'ai racheté un Amstrad CPC464 et je recherche désespérément les logiciels Manic Miner et Fruity Franck, merci de me contacter sur [patrickgilbert@hotmail.com](mailto:patrickgilbert@hotmail.com)

Achète CPC6128 couleur TBE sur Saint-Etienne et sa région (je ne peux pas me déplacer). [berengue.michael@wanadoo.fr](mailto:berengue.michael@wanadoo.fr)

A vendre CPC464 et CPC464+ avec des jeux, le tout en parfait état de marche. Les deux CPC sont avec des écrans couleurs, plus éventuellement un écran monochrome. Faire offre. Contacter-moi : [obsolom@yahoo.fr](mailto:obsolom@yahoo.fr)

Quinquagénaire nostalgique et pas du tout bidouille, je souhaite rejouer avec mes petits enfants à Sorcery, Boulder Dash, Bombjack, Bruce Lee, Fruity Franck et une course de moto dont j'ai oublié le nom (il y a un tableau avec les Monts Rushmore e.a.). Comme on m'a volé mon Amstrad et les 20 disquettes, je suis acheteur d'un CPC6128 avec joystick avec ou sans les jeux précités. Etudie toute offre. Merci d'avance. [Yhis6@aol.com](mailto:Yhis6@aol.com)

Je cherche le "Livre du lecteur de disquettes Amstrad CPC" (collection Micro Application). Contactez-moi : [enzoe@galema.com](mailto:enzoe@galema.com)

Vends CPC464 + écran couleur + imprimante avec ruban d'encre + joystick + jeux. Ensemble ou séparément. Prix à débattre. Contactez-moi rapidement à : [arnaud.turier@free.fr](mailto:arnaud.turier@free.fr)



# LA RUBRIQUE X

Oh bien, voici une nouvelle Rubrique X qui commence ! J'en profite pour vous rappeler qu'un fanzine vit essentiellement grâce à ses lecteurs... donc, écrivez-nous abondamment ! Et ne vous contentez pas d'envoyer vos proses à La Rédac', faites plaisir à tout le monde !

Comme cela est désormais la tradition, voici pour commencer l'adresse du lieu où tout a commencé. L'antre d'où sont sortis chacun des numéros de Quasar CPC depuis les premiers balbutiements... La Rédac' !

La Rédac' — RIMAURO Philippe & Gilles  
8, chemin des Maillos  
09200 SAINT-GIRONS  
quasarcpc@free.fr  
www.chez.com/futurs

Vous voulez commander des anciens numéros ? Vous abonner ? Nous faire part de vos critiques ? C'est à cette adresse qu'il vous faut écrire !

Ensuite, si vous désirez plus particulièrement écrire à un des rédacteurs à propos de ses rubriques, pour le conseiller, le corriger, l'insulter, voici toutes les adresses.

Offset — RIMAURO Philippe  
1, avenue Saint Donatien  
06600 ANTIBES  
rimauro@ifrance.com

Embêtez prestement tonton Offset pour tout ce qui concerne la programmation de votre CPC ou CPC+.

Zik — RIMAURO Gilles  
6, allée des  
Sciences Appliquées  
Résidence ? - Appt 892  
31400 TOULOUSE  
grimauro@etud.insa-tlse.fr

Zik attend avec impatience vos réactions à propos de ses articles afin de savoir dans quelle voie s'engager dans les prochains numéros.

CNGSoft — GONZALES César Nicolás  
cngsoft@lettera.net  
members.es.tripod.de/cng

CNGSoft est lui aussi dans l'attente de vos réactions pour cibler ses prochaines interventions.

Hicks — THOMASSETTE David  
26, rue des Maisonnettes  
54300 LUNEVILLE  
another-world@cpcscene.com

Hicks espère avoir de nombreux échos à propos de son gros travail sur le dossier consacré à la programmation des jeux.

Tony — RENEAUT Antoine  
KRESTIO PASTOUHOV, 22  
4000 PLOVDIV  
BULGARIE  
areneaut@bigfoot.com  
www.chez.com/tonylife

Tony, notre expatrié, attend vos cartes postales !





# Histoires Perpendiculaires

## - Des Trous -

### Des trous.

C'est en sortant de l'aéroport que j'ai pu les voir pour la première fois. Enfin, je dis "les voir", mais je devrais plutôt dire "les sentir" car dès le début ils se sont montrés agressifs.

### Des trous.

Ils sont là, simplement là, à chaque pas que je fais, à chaque regard que je pose, à chaque morceau de gruyère que je mange (c'est pas vrai j'en mange pas).

### Des trous.

Sur les murs, sur les routes. Dans les nuages, dans les grillages. En voiture, ils déroutent. Dans les bagages, j'en ai la rage.

### Des trous.

Je vous dis qu'il y en a partout. Y'a rien à faire, leur présence m'est une pitance.

On connaissait ceux qui sont là-haut : dans l'ozone ou dans l'espace, tout noirs.

On connaissait ceux que chaque jour on cotoie, qui déballetent leur science pour que dalle, d'balle.

On connaissait ceux



dans lesquels les plus grandes rencontres de l'histoire eurent lieu, ces rencontres qui ne cesseront de nous étonner à tous, que d'émois : Bassoues, Catenoy.

Mais la rude et pourtant si précieuse expérience de cette vie m'a conduit à l'apogée du mystère, là où les trous surgissent comme des éclairs, apparaissent et dis-

paraissent comme l'ondée du matin qu'on agresse.

Oui, mes amis, c'est ici, en Bulgarie, là où petit à petit je fais mon trou, qu'ils se manifestent le plus souvent.

D'où viennent-ils, que veulent-ils, pourquoi sont-ils si méchants... que de questions sans réponse à celui qui chaque jour dans leur piège s'enfoncé.

Note de l'auteur :

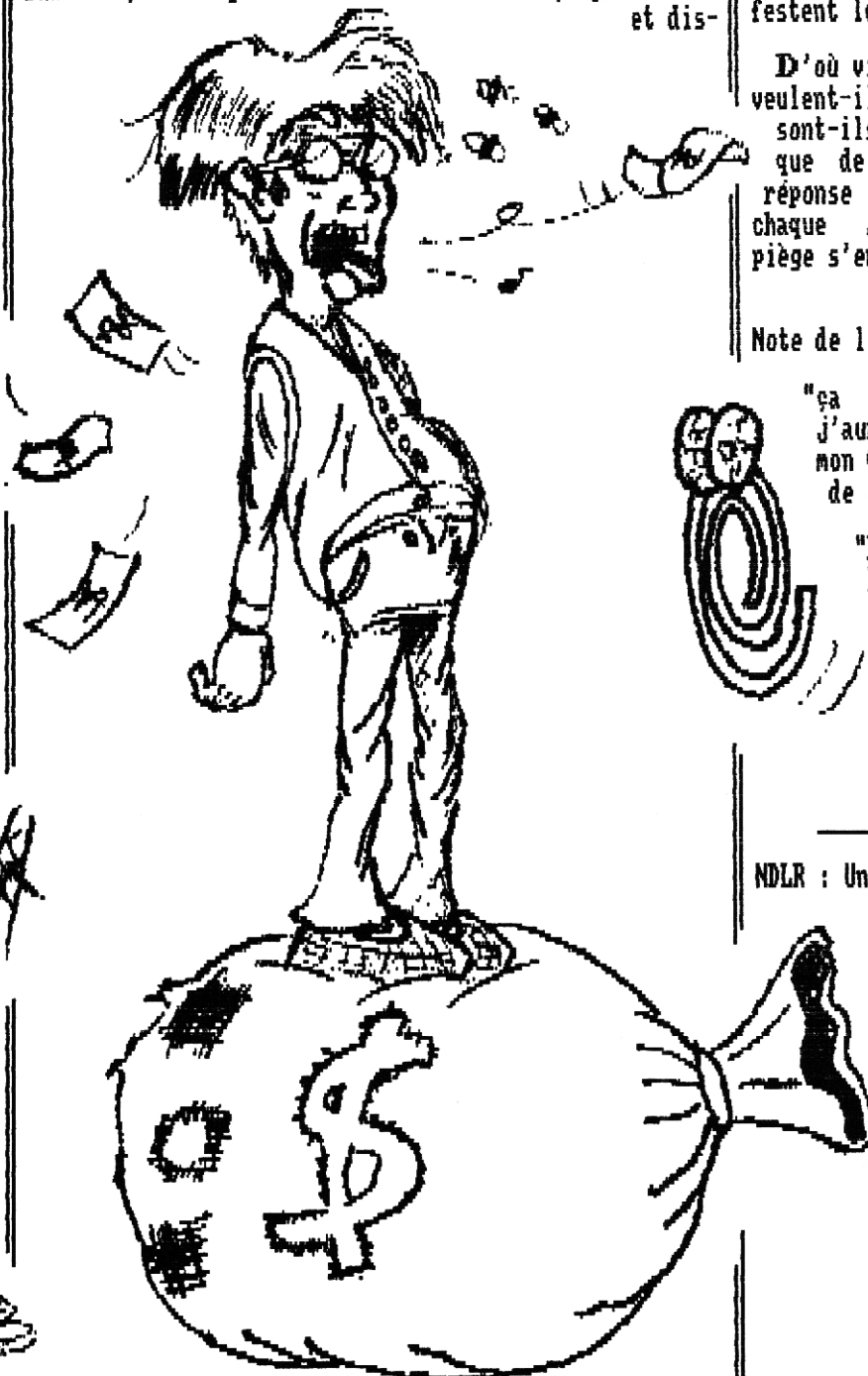
"ça ira mieux quand j'aurai une alim pour mon CPC qui n'a pas de courant."

"Par ailleurs, la Bulgarie est un fort beau pays malgré trou."

Tony

NDLR : Un grand merci à Tony qui a fait l'effort de participer à ce numéro de Quasar CPC malgré son exil...

Trou !



## Le déplombage

Le petit blond à lunettes lit le titre et dit : "Keskesesah ? Quasar CPC favorise les actions illégales maintenant ?". Non, non, pas de panique, ce n'est pas l'objectif de cet article. Je pense en fait à tout autre chose : comment conserver pour toujours ces jeux sauvés sur cassette avec des protections qui ne servent qu'à provoquer des erreurs et nous obligent à rembobiner encore et encore jusqu'à détruire la bande magnétique ? C'est un peu comme l'histoire des civilisations inconnues : leur condamnation vient de l'impossibilité de comprendre leurs textes, alors que les civilisations plus tournées vers le futur faisaient l'effort d'être plus explicites. Ainsi, avec cet article vous serez capables de sauver pour la postérité les programmes et jeux du CPC.

## Préliminaires

D'abord, c'est quoi une protection ? Il s'agit de l'utilisation de techniques de sauvegarde et récupération incompatibles avec les méthodes standard du système pour rendre impossible la manipulation frauduleuse. Sur CPC, cela signifie enregistrer des cassettes avec des signaux illisibles sous Basic, des disquettes refusant d'être lues par l'Amsdos, etc...

Mais les protections ne sont pas indestructibles : après tout, il faut toujours un programme de chargement sans protection pour pouvoir être exécuté depuis le système ; et ce programme contient les procédés capables de charger les programmes protégés !

C'est alors que le déplombage (un mot assez idiot, mais l'anglicisme "cracking" est encore pire) est essentiellement la manipulation du chargeur pour pouvoir charger le programme protégé et le sauvegarder sans protection pour pouvoir ensuite l'utiliser tranquillement. Voyons quelques exemples par ordre de difficulté.

## Fichiers standards

Dans les premiers jours du CPC, Ansoft publiait les programmes sans protection ; alors, pas de problème, un copieur de fichiers (Transmat, Discology, ...) et voilà

c'était fait. Quelques fois, il y avait des modifications à faire, mais pas grand chose : par exemple, un LOAD "fichier" avec un nom valide sur cassette mais pas sur disquette (avec des espaces, des points, etc.). Une particularité est l'emploi d'un fichier binaire chargeur : sur disquettes il y a un problème car le système met aux indirections standard toutes les adresses système signifiant que l'Amsdos est inhibé et utilise les routines cassette. Mais il n'était pas très difficile d'écrire un petit chargeur Basic chargeant le binaire (chargeur, chargeant... ha ha !) via les valeurs du header binaire (tous les copieurs de fichiers vous informent de ces valeurs) :

```
10 MEMORY commence-1:LOAD"chargeur.bin",commence
20 CALL execution
```

## Blocs non standards principaux

Tiens, la bordure de l'écran a de belles couleurs pendant le chargement du jeu, la disquette bouge assez rapidement, c'est rigolo ! Oui, mais un peu plus difficile car ça signifie que notre prog est protégé avec une sauvegarde non standard. Que faire ? Un copieur de cassettes ne comprendra pas le bloc et un copieur de disquettes peut faire des faces en essayant de lire les secteurs du bloc.

Ce que nous devons faire maintenant est un peu plus difficile. Connaissez-vous les instructions du Z80 ? Je suppose que oui, car vous êtes de fidèles lecteurs et pas mal de numéros de Quasar CPC ont dédié la rubrique Coding au Z80... :) Sans blague, si vous ne connaissez pas le Z80, n'essayez pas de suivre car ce n'est pas facile à expliquer sans un peu de technique.

Démarrons un debugger pour examiner le programme chargeur. A moins qu'il ne soit encrypté (nous verrons cela plus tard) il est facile d'y trouver des lignes similaires :

```
LD IX,adresse (c'est sûr !)
LD DE,longueur (c'est sûr !)
LD A,synchronisation (très probable)
SCF (très probable)
CALL routine (ou JP routine, ça depend, mais sûr !)
```

Les registres peuvent être différents, mais la structure est toujours la même : c'est un appel à la routine

qui lit le contenu du bloc non standard à l'adresse spécifiée.

Il y a deux façons de sauvegarder ces blocs. La première consiste à introduire, après le dernier "CALL routine", un peu de code pour sauvegarder tout ce qui a été chargé ; mais ce n'est pas toujours une bonne idée car le chargeur peut charger un bloc sur un autre (par exemple : un jeu chargeant l'écran de présentation et plus tard l'écran des PANNES ET MARQUEURS). La deuxième solution est de faire une sauvegarde pour chaque bloc, ce qui a l'avantage de nous permettre d'employer des adresses différentes (beaucoup de jeux chargent un bloc sur les appels système !).

Après avoir sauvegardé tous les blocs, il n'y a plus qu'à écrire un chargeur Basic chargeant les blocs devenus fichiers aux adresses correctes, et voilà ! Une légère difficulté est -à nouveau- la gestion des blocs effaçant les appels du système ; mais la solution est élémentaire : charger les blocs à un endroit différent et, après le chargement du dernier bloc, les déplacer aux bonnes adresses avec une routine binaire. Pour ajouter le code binaire dans le chargeur Basic, des lignes de DATA et un peu de POKE et CALL sont suffisants. Voici un exemple :

```
10 MEMORY adresse minimale-1
20 LOAD "bloc.1",adresse.1:LOAD "bloc.2",adresse.2, etc.
30 FOR h=&BF00 TO &BF0E
40 READ h#:POKE h,VAL("&"#h)
50 NEXT
60 CALL &BF00
70 DATA F3,21,00,C0,11,00,A7,01,00,19,ED,B0,C3,00,01
```

Où le code binaire est :

```
DI ; pas d'interruptions!
LD HL,&C000 ; origine
LD DE,&A700 ; destination
LD BC,&1900 ; longueur
LDIR ; copier
JP &0100 ; sauter
```

Si le chargeur original fait plus de choses, tels que des appels aux blocs chargés, reproduisez-les dans le chargeur Basic.

## Blocs non standards secondaires

Nous avons déplombé un beau jeu et nous voulons jouer ; tout va bien... Non ! Le jeu a besoin de charger les niveaux ! Il semble que notre déplombage ne soit pas complet ; le programme utilise plus de blocs, non seulement ceux contenant le programme lui-même, mais aussi ceux contenant les données.

Récupérez le debugger car il faut chercher l'endroit où le programme demande les blocs secondaires. Beaucoup de compagnies de logiciels utilisent toujours la même

routine ; il est alors assez facile de la trouver dans le programme car c'est la même que celle du chargeur. C'est encore plus facile si le programme écrit des textes du type "Chargement bloc XX", "Erreur de chargement" etc. car il n'y a plus qu'à les trouver et à localiser la partie du programme les utilisant. En modifiant un peu le code, on peut alors sauvegarder les blocs secondaires de la même façon que les primaires.

Mais comment faire maintenant pour que le programme principal utilise les blocs secondaires ainsi sauvegardés ? C'est l'heure de patcher sérieusement le programme. Si le programme n'écrase jamais les appels système (c'est à dire les régions mémoire &0000-&003F et &A700-&BFFF (pour CPC sans ROM disque : &0000-&003F et &AC00-&BFFF)), ceux-ci peuvent être utilisés pour patcher la partie du programme qui charge les blocs secondaires. Mais dans le cas où la mémoire système est détruite par le programme (le code, les données ou les buffers), l'emploi des extensions de mémoire est obligatoire.

Evidemment, le programme ne doit pas utiliser les extensions mémoire ou, il doit au moins y laisser assez de place pour sauvegarder les appels système et la mémoire manipulée par le programme. Pourquoi ? Parce que l'on doit récupérer l'état de la machine d'avant le chargement du bloc. Voici un exemple très simple :

Programme chargeur :

```
10 MEMORY minimum-1:LOAD "programme.1",adresse.1, etc.
20 FOR h=adresse.du.patch TO adresse.du.patch+longueur
   du.patch
30 READ h#:POKE h,VAL("&"#h)
40 NEXT
50 DATA donnees.du.patch
60 FOR h=adresse.des.preparations TO adresse.des.preparations+longueur
   des.preparations
70 READ h#:POKE h,VAL("&"#h)
80 NEXT
90 DATA donnees.des.preparations
100 CALL adresse.des.preparations
```

Préparations :

```
...préparations initiales...
DI ; pas d'interruptions !
LD BC,&7FC4 ; bloc 4 de mémoire 128k
OUT (C),C ; sélection du bloc
LD DE,&4000 ; adresse de sauvegarde (1)
LD HL,&0000 ; origine
LD BC,&40 ; longueur
LDIR ; sauvegarder
LD HL,&A700 ; origine
LD BC,&1900 ; longueur
LDIR ; sauvegarder
LD BC,&7FC0 ; état normal de la mémoire
OUT (C),C
...préparations finales...
JP execution.du.programme
```

## Patch :

```
...préparations initiales...
DI ; pas d'interruptions !
LD BC, &7FC4 ; bloc 4 de mémoire 128k
OUT (C), C ; sélection du bloc
LD DE, &5940 ; adresse de sauvegarde (2)
LD HL, &0000 ; origine
LD BC, &40 ; longueur
LDIR ; sauvegarder
LD HL, &A700 ; origine
LD BC, &1900 ; longueur
LDIR ; sauvegarder
LD (&7280), SP ; sauvegarde de la pile
LD HL, &4000 ; adresse de sauvegarde (1)
LD DE, &0000 ; origine
LD BC, &40 ; longueur
LDIR ; récupération
LD DE, &A700 ; origine
LD BC, &1900 ; longueur
LDIR ; récupération
LD BC, &7FC0 ; état normal de la mémoire
OUT (C), C ; sélection du bloc
LD SP, &C000 ; une bonne valeur pour le système
... chargement du bloc...
DI ; pas d'interruptions !
LD BC, &7FC4 ; bloc 4 de mémoire 128k
OUT (C), C ; sélection du bloc
LD HL, &5940 ; adresse de sauvegarde (2)
LD DE, &0000 ; origine
LD BC, &40 ; longueur
LDIR ; récupération
LD DE, &A700 ; origine
LD BC, &1900 ; longueur
LDIR ; récupération
LD SP, (&7280) ; récupération de la pile
LD BC, &7FC0 ; état normal de la mémoire
OUT (C), C ; sélection du bloc
...préparations finales...
RET/JP adresse ; dépendant du chargeur original
```

## Recommandations finales

Les chargeurs encryptés peuvent être très compliqués (Speedlock, Nemesis...) mais la plupart se contentent de chaînes auto-modifiantes de XORs. Un peu de traçage avec le debugger serait suffisant.

Pour utiliser la compression proposée dans mon article précédent, n'oubliez pas que le CPC vous exige un buffer de 2ko pour la lecture des données ! Les programmes de compression comme Cheese ou Imploder peuvent être dangereux car leur optimisation est assez agressive et votre code chargeur peut souffrir de leurs effets. Par contre, ils sont magnifiques pour des programmes d'un seul fichier.

L'utilisation de l'espace &BE80-&BF7F pour votre code est traditionnelle, mais faites attention s'il faut écrire dessus ; alors, essayez &0040, une bonne alternative. Le RST 6 peut-être très utile pour des

patches d'espace limite : pomez en &0030 quelque chose du type LD BC, &7FC4:OUT (C), C:JP adresse.d.extension.du .patch, c'est très intéressant !

Essayez d'inclure votre pseudo et la date du dépliement dans le chargeur ; si possible faites une jolie intro. Mais ne modifiez pas les programmes ; s'il faut les patcher, faites le dans le chargeur. N'oubliez pas que notre but est la conservation des programmes. Sur-tout, ne changez pas les textes des programmes : ce serait idiot !

CNGSOFT



Après la mise au point sur les interruptions du numéro 18 j'ai décidé, après mûre réflexion, de vous faire un petit dossier sur ce que nous appelons communément de Gate Array. C'est bien, je vois que le petit blond à lunettes a de la mémoire... j'avais effectivement déjà fait un tel dossier dans notre numéro 6... il y a sept ans déjà ! Mais bon, le dossier que je vous propose aujourd'hui est bien plus complet et va nous permettre de corriger quelques petites erreurs, par omission certes, qui s'étaient glissées dans cet ancien article.

## Le Gate Array

Derrière cette appellation se cachent en fait trois composants essentiels du CPC, tous sur le port &7Fxx :

- Le Gate Array proprement dit,
- Le PAL 40030 de gestion des banks,
- L'Asic sur les CPC plus.

Le premier est le circuit qui est réellement au coeur du CPC : il distribue les horloges, génère la vidéo et les interruptions, gère les accès aux ROMs... bref, sans lui pas la peine d'espérer faire fonctionner un CPC. Pratiquement, au niveau logiciel, on va s'adresser à la bête lorsqu'on voudra changer les couleurs et le mode vidéo, retarder les interruptions ou encore modifier l'état des commutations de ROMs. Jetez un coup d'oeil sur le synoptique du port &7Fxx trois pages plus loin... Eh bien notre Gate Array s'occupe des registres PENR, INKR et RMR.

Vient ensuite notre PAL 40030. Physiquement situé juste à côté de notre Gate Array sur la carte mère des CPC6128, celui-ci a pour rôle de gérer nos banks d'extension RAM. C'est lui qui entre en jeu lors de la programmation du registre RMR sur notre port &7Fxx.

Enfin, sur les CPC plus, nous avons l'Asic. Celui-ci remplace à lui seul les deux chipsets cités ci-dessus en ajoutant par ailleurs le registre RMR2, spécifique à l'adressage des registres de l'Asic (page I/O Asic) et aux ROMs du port cartouche.

Cette petite mise au point ayant été faite, nous allons voir de plus près comment réagit ce port... Comme vous le constaterez sur le schéma, l'octet que nous envoyons sur ce port peut être découpé en deux : les trois bits de poids fort (5, 6 et 7) sélectionnent le registre à programmer (PENR, INKR, RMR, ...) certaines valeurs sont en effet illégales (d'où seulement 5 registres pour 8 combinaisons de bits possibles), et les cinq bits de poids faible (0 à 4) représentent les paramètres : la valeur associée au registre choisi. Enfin, faites bien attention au fait que ce port fonctionne en écriture seule, impossible d'y lire la moindre information.

## Les registres de couleur

Si vous avez un soupçon d'intuition, vous aurez sans doute compris que ceux-ci sont au nombre de deux : PENR (PENcil Register) et INKR (INK Register). Je ne vais pas m'attarder sur ces deux registres qui sont certainement les plus connus du CPC ! C'est en effet grâce à ces registres qu'il nous est possible de faire des rasters... Pour ce qui ne savent pas comment ça marche, je les renvoie au tout premier numéro de Quasar CPC !

En résumé, le registre PENR permet de sélectionner le "crayon" à modifier alors que le registre INKR définit la nouvelle "encre" à affecter au "crayon" courant. Je pense qu'il n'y a pas plus simple ! En outre, il est important de noter que le numéro d'encre (ou de couleur si vous préférez) n'est pas celui communément utilisé sous Basic (via l'instruction INK), il s'agit en effet des numéros de couleurs dites hardware, à titre de rappel je vous ai mis un tableau de correspondance entre les couleurs software (système) et les couleurs hardware quatre pages plus loin.

## Le registre RMR

Nous attaquons les choses sérieuses. Ce registre, RMR (Rom Memory Register ?), permet de définir de façon cruciale la configuration mémoire et vidéo du CPC. Nous allons donc voir une à une les fonctions qui se cachent dans ses cinq petits bits.



Les deux premiers bits, qui répondent aux deux noms de bits 0 et 1, permettent de choisir le mode vidéo. Ah, je vois que même le petit blond à lunettes arrive à suivre... ah ben non, il comprend pas : pourquoi diable y'a-t-il 4 modes possibles (0, 1, 2 ou 3) alors qu'il est bien connu que sur CPC nous n'avons que trois mode graphiques ? hum ? Eh bien la réponse est simple : nous avons effectivement quatre mode vidéo et pas seulement trois ! Ceci étant, le mode supplémentaire (le 3), n'est pas utilisé car il nous offre la résolution du mode 0 avec le nombre de couleurs du mode 1 ! Autant dire qu'on peut facilement s'en passer...

Le bit qui suit (oui, c'est bien le bit 2, un bon point pour le petit blond à lunettes qui a su rester concentré) nous permet de commander la connexion de la ROM inférieure. Mais qu'est-ce donc que la ROM inférieure ? Sur CPC ancienne génération, rien de plus simple, il s'agit de la ROM Firmware, celle qui contient le système d'exploitation du CPC, attention, ce n'est pas la ROM Basic ! Cette ROM se connecte sur la page mémoire &0000-&3FFF. Sur CPC+, c'est un peu plus complexe puisque l'adresse et le contenu de la ROM inférieure sont déterminés via le registre RMR2 décrit un peu plus loin.

Vient ensuite le bit 3. Celui-ci nous permet de commuter la ROM supérieure. Le petit blond à lunettes est complètement largué... J'explique : cette fameuse ROM se connecte en &C000-&FFFF et peut être n'importe quelle ROM d'extension de votre CPC. Sur un CPC6128 de base nous n'avons que deux ROM d'extension : la ROM 0 (qui contient le Basic) et la ROM 7 (qui contient l'Amsdos) mais il est possible d'en ajouter presque à volonté (256 ROMs au maximum) grâce à des extensions telles que le Romboard, la Romcard ou la Ramcard. Le choix du numéro de la ROM commutée se fait grâce à un registre auxiliaire situé sur le port &DFxx. Ce port est présenté deux pages plus loin sur un petit schéma. Vous pouvez constater que s'il est très simple sur les CPC ancienne génération (on y envoie le numéro de la ROM que l'on désire commuter), il dispose de deux modes de fonctionnement sur CPC+ : lorsque le bit 7 vaut 0 (valeurs 0 à 127), il se comporte comme sur les vieux CPC et représente le numéro de ROM à commuter. En revanche, quand ce bit 7 est mis à 1, les bits 0 à 4 permettent alors de choisir le numéro de ROM de la cartouche à commuter ! Eh oui, sur CPC+, outre les ROMs d'extension classiques, nous avons des cartouches qui ne sont ni plus ni moins que d'autres ROMs d'extension. Afin d'éviter toute confusion, les ROMs d'extension classique (comme sur CPC ancienne génération) s'appellent les ROMs logiques ; et les ROMs d'extension contenues dans les cartouches des CPC+, les ROMs physiques.

Pour les curieux j'ai mis le numéro des ROMs physique de la cartouche de Burnin'Rubber des CPC+ dans le

schéma. Nooon ! J'y crois pas, le petit blond à lunettes à une forme d'enfer aujourd'hui ! Il a remarqué une petite subtilité intéressante... Sur CPC+, si une ROM ne peut être accessible que par un numéro physique ou un numéro logique, il existe des ROMs qui sont accessible via les deux normes ! Eh oui, la ROM Basic par exemple, porte le numéro logique 0 et le numéro physique 1 ; mais c'est également le cas de la ROM Amsdos qui a le numéro physique 3 et logique 7. Ceci dit, si le numéro physique est une constante (d'où son nom), la ROM correspondant à un numéro logique peut-être changée. En effet, si, sur CPC+, vous connectez une Ramcard dans laquelle vous programmez la ROM Parados en numéro 7. Eh bien la ROM logique numéro 7 sera désormais Parados mais la ROM physique numéro 2 (qui était aussi la ROM logique numéro 7) reste l'Amsdos !

Et pour en finir avec ce registre, nous avons le petit bit 4. Si vous le mettez à zéro rien ne se passe. En revanche, lorsque vous le mettez à 1, vous faites une remise à zéro du diviseur d'interruption. Mais qu'est-ce donc que cette bestiole ? Eh bien, comme vous le savez certainement, le Gate Array génère les interruptions dans le CPC : c'est lui qui, tous les 300èmes de secondes (ou toutes les 52 lignes si vous préférez) demande à notre Z80 d'entrer en interruption (pour en savoir plus allez de ce pas lire l'article du numéro précédent). Lorsque vous mettez le diviseur d'interruption à zéro (ce qui a lieu automatiquement à chaque VBL et toutes les 52 HBL) vous allez retarder la prochaine interruption en provenance du Gate Array qui aura lieu 52 ligne plus bas (à partir de votre remise à zéro).

## Le registre RMR2

Ce registre est disponible uniquement sur CPC+. Il a deux objectifs : connecter/déconnecter la page I/O des registres de l'Asic et configurer la ROM inférieure (dont la connexion est pilotée via le bit 2 du registre RMR). Les trois bits inférieurs nous permettent de choisir le numéro de la ROM physique (parmi les huit premières) qui sera commutée en tant que ROM inférieure. Les deux bits suivants sélectionnent la configuration ROM/Asic désirée. On peut choisir de commuter la ROM inférieure en &0000-&3FFF, &4000-&7FFF ou &8000-&C000. Une combinaison permet également de commuter la page I/O Asic en &4000-&7FFF, dans ce cas précis la ROM basse se trouve alors en &0000-&3FFF.

Par défaut le registre RMR2 vaut 0 ce qui signifie que la page I/O Asic est déconnectée, la ROM inférieure commutable en &0000-&3FFF et qu'il s'agit de la ROM physique numéro 0. Nous sommes dans ce cas entièrement compatible avec l'adressage de la ROM inférieure des CPC ancienne génération puisqu'il se trouve que la ROM physique 0 de la cartouche système des CPC+ est le Firmware (pfieu, on a eu chaud !).

## Le petit dernier : Le registre MMR

Ce registre (Memory Management Register ?) a été largement décrit à plusieurs reprises dans nos pages. Il permet de choisir la page de 64ko dans la laquelle sélectionner la bank de 16k à commuter en &4000-&7FFF. Il offre également la possibilité d'effectuer des commutation de RAM spéciales qui sont décrites sur le schéma de la page suivante. Je vous rappelle juste que sur CPC464 il n'y a pas de RAM d'extension et que sur CPC6128 il y a une page de 64ko (la numéro 1) d'extension. Ce registre permet en théorie un adressage maximum de 512ko... On peut toutefois passer outre grâce à des registres MMR auxiliaire comme cela est le cas dans la fameuse MemCard 2Mo de RAM7 ; cette carte en plus du registre MMR qui permet l'accès aux 512 premiers kilooctets, offre 3 autres registres similaires ; en fait nous avons un registre MMR par page de 512ko.

### Quelques détails

Je pense que nous avons à peu près fait le tour des fonctions accessibles via le port &7Fxx. Je vous propose maintenant de mettre en avant quelques variantes dans le fonctionnement du Gate Array émulé par l'Asic des CPC+.

Tout d'abord, en ce qui concerne la mise à jour des couleurs par rapport aux données vidéo, vous constaterez, si vous vous essayez aux split rasters, que sur CPC+ ceux-ci sont décalé d'une microseconde (un NOP Z80) vers la droite. Donc si vous avez à mettre en place des splits rasters avec un positionnement bien précis par rapport à ce qui est affiché à l'écran pensez à en tenir compte !

Ensuite, concernant la programmation des registres PENR et INKR sur CPC+, étant donné que celui-ci dispose d'une palette de 4096 couleurs (RVB 12 bits) au lieu de 27 couleurs, vous devez bien comprendre que l'Asic fait en fait une conversion du codage de couleurs "ancienne génération" vers son propre code. En fait, la palette des 27 couleurs des CPC classiques est déduite d'un codage RVB trois états, chaque composante RVB pouvant être à zero, à demi-intensité ou à pleine intensité alors que sur CPC+ ce codage est du 4 bits (valeurs de 0 à 15) pour chacune de ces mêmes composantes. Ah, je

vois que le petit blond à lunettes a compris le problème ! Décidément il devient bon ! Lors de la conversion, la pleine intensité c'est évidemment 15... Et la demi-intensité ? Hum,  $15/2=7.5$ ... On met quoi alors ? 7 ou bien 8 ? Eh bien l'Asic met 6 ! Mais pourquoi 6 ? L'astuce vient du fait que la "demi-intensité" du CPC n'est en fait pas demi du tout... Le niveau haut ne vaut pas le double du demi-niveau d'intensité, c'est pourquoi la palette du CPC contient des couleurs très proches alors que d'autres n'ont pas de voisins proches : l'échelle n'est pas linéaire. Ceci dit, malgré cette conversion à 6 plutôt que 7 voire 8, les couleurs "ancienne génération" apparaissent toujours plus vives sur un CPC+ que sur un CPC... mais bon, si vous essayez de convertir à la main à 5 plutôt que 6 vous constaterez que vous êtes trop pâle, la bonne valeur serait entre les deux !

### Quelques informations sur Les ports

Je vous propose un petit jeu amusant. Depuis notre bon vieux Basic, tapez la ligne suivante :

```
OUT &7F00,&10:OUT &7F00,&40
```

Vous avez logiquement un flash gris (couleur 0) dans le border (PENR=&10)... Je vous rappelle en effet que le système d'exploitation remet les couleurs à jour régulièrement, donc votre modification ne dure pas. Mais bon, là n'est pas le problème. Tapez maintenant :

```
OUT 0,&C2
```

Paf, votre CPC a planté ! c'est normal, on a activé le mode spécial du registre MMR qui consiste à

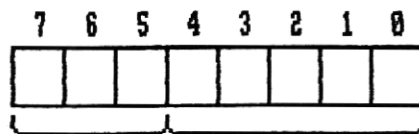
échanger les 64ko de base avec les premiers 64ko d'extension : crash inévitable... Pourtant... Comment se fait-il que le registre MMR ait été programmé puisqu'on pas accédé au bon port ? De plus, tapez le premier exemple mais sur le port 0 au lieu du port &7Fxx ; ça ne marche pas, aucun changement de couleurs au rendez-vous ! En fait, tout ceci est normal, rappelez-vous, le registre MMR est pris en compte par le PAL et non le Gate Array ! Or il se trouve que le Gate Array réagit à tous les ports pour lesquels le bit 15 est à zéro et le bit 14 à un alors que le PAL se contente du bit 15 à 0... pensez-y lorsque vous optimisez vos accès aux ports, notamment pour programmer le CRTIC via les ports xC et xD plutôt que BC et BD... Le PAL vous guette ! Zut ! La place me manque pour la dernière remarque... Prenez un CPC et un CPC+, et comparez : `OUT &7F00,&10:a=INP(&7F00)`... Offset



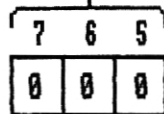
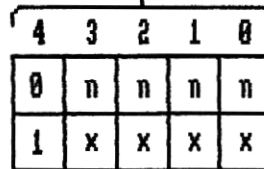
PORT &7Fxx : Gate Array, RAM, Asic



Bits de contrôle



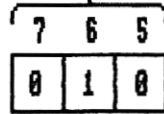
Bits de commande



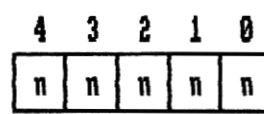
**PEMR** : Sélection du stylo courant

Sélection du stylo mmm

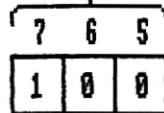
Sélection du border



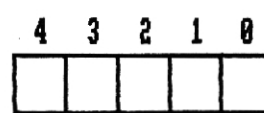
**INKR** : Sélection de l'encre du stylo courant



Sélection de la couleur  
Dans la palette des 27 couleurs même sur CPC plus ! (la palette des 4096 couleurs est accessible uniquement via la page I/O Asic)



**RMR** : - Interruptions  
- Modes écran  
- Connexion des ROMs

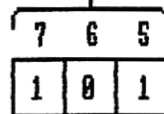


Diviseur d'interruption  
0 : Inchangé  
1 : Remis à zéro

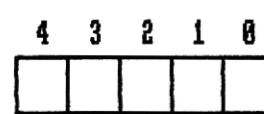
Connexion ROM supérieure  
0 : Oui (cf. port &DFxx)  
1 : Non

Connexion ROM inférieure  
0 : Oui (cf. MODE 2+)  
1 : Non

Mode écran  
0 à 3



**RMR2** : - Connexion de la page I/O Asic  
- Choix de la configuration pour la ROM inférieure

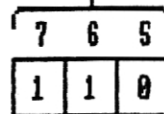


Les modes écran :  
MODE 0 : 160x200 16 couleurs  
MODE 1 : 320x200 4 couleurs  
MODE 2 : 640x200 2 couleurs  
MODE 3 : 160x200 4 couleurs

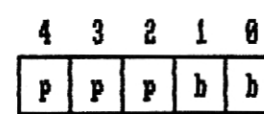
Adresse ROM inférieure / Page I/O Asic  
0 0 : &0000-&3FFF / Non connectée  
0 1 : &4000-&7FFF / Non connectée  
1 0 : &8000-&BFFF / Non connectée  
1 1 : &0000-&3FFF / &4000-&7FFF

Numéro physique de la ROM inférieure  
0 à 7

Ce mode n'existe que sur CPC plus, sur les CPC ancienne génération la seule ROM logique existante est le firmware et elle ne peut se connecter qu'en &0000-&3FFF. Sur CPC plus les ROMs logiques sont celles de la cartouche.

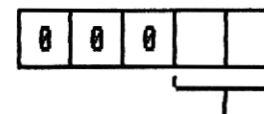


**MNR** : - Connexion des banks  
- Opérations spéciales sur la RAM centrale



Bank de 16ko bb de la page de 64ko ppp connectée en &4000-&7FFF (ppp ≠ 000)

0 0 : RAM centrale linéaire bank déconnectée  
0 1 : Bank 3 de la première page de 64ko en &C000-&FFFF  
1 0 : Première page de 64ko en &0000-&FFFF  
1 1 : Bank 3 de la première page de 64ko en &C000-&FFFF et RAM centrale &C000-&FFFF en &4000-&7FFF



Opérations sur la RAM

L'adressage mémoire maximum via ce registre est de 512ko. Toutefois, des cartes d'extension telles la MemCard permettent de dépasser cette limite en ajoutant des registres MNR supplémentaires par page de 512ko.



Port &DFxx : Choix de la ROM supérieure

Sur CPC ancienne génération

7 6 5 4 3 2 1 0

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|

Choix du numéro de ROM (0-255)  
(ROM logiques seulement)

Sur CPC plus

7 6 5 4 3 2 1 0

|   |   |   |  |  |  |  |  |
|---|---|---|--|--|--|--|--|
| 0 |   |   |  |  |  |  |  |
| 1 | x | x |  |  |  |  |  |

Choix d'une ROM logique (0-127)

Choix d'une ROM physique (0-31)

Correspondance numéros de couleurs Software et Hardware

| Couleurs    | Software | Hardware | Couleurs         | Software | Hardware |
|-------------|----------|----------|------------------|----------|----------|
| Noir        | 0        | 20       | Bleu Pastel      | 14       | 31       |
| Bleu        | 1        | 4        | Orange           | 15       | 14       |
| Bleu Vif    | 2        | 21       | Rose             | 16       | 7        |
| Rouge       | 3        | 28       | Magenta Pastel   | 17       | 15       |
| Magenta     | 4        | 24       | Vert Vif         | 18       | 18       |
| Mauve       | 5        | 29       | Vert Marin       | 19       | 2        |
| Rouge Vif   | 6        | 12       | Turquoise Vif    | 20       | 19       |
| Pourpre     | 7        | 5        | Vert Citron      | 21       | 26       |
| Magenta Vif | 8        | 13       | Vert Pastel      | 22       | 25       |
| Vert        | 9        | 22       | Turquoise Pastel | 23       | 27       |
| Turquoise   | 10       | 6        | Jaune Vif        | 24       | 10       |
| Bleu Ciel   | 11       | 23       | Jaune Pastel     | 25       | 3        |
| Jaune       | 12       | 30       | Blanc Brillant   | 26       | 11       |
| Blanc       | 13       | 0        |                  |          |          |

ROM logiques sur un CPC de base (qu'il soit "plus" ou non)

ROM 0 : Basic

ROM 7 : Amsdos (sauf CPC464)

Des ROM logiques peuvent être ajoutées au système via un Romboard, une Roacard, un Hacker, etc... Elles se connectent uniquement en ROM supérieure.

ROM physiques sur un CPC plus avec la cartouche de Burnin'Rubber verte :

ROM 0 : Firmware

ROM 1 : Basic

ROM 2 : Vierge (inutilisée)

ROM 3 : Amsdos

ROM 4 : Burnin'Rubber ROM 0

ROM 5 : Burnin'Rubber ROM 1

ROM 6 : Burnin'Rubber ROM 2

ROM 7 : Burnin'Rubber ROM 3

Les ROMs physiques sont celles de la cartouche et se connectent en ROM inférieure ou supérieure.



Dans tout programme conséquent on est amené à effectuer des calculs. Selon le cas les routines de calcul sont plus ou moins facile à mettre au point et leur temps d'exécution est critique ou non. Lorsque l'on peut se permettre de "sacrifier" un peu (ou beaucoup) d'espace mémoire on a alors souvent recours à l'utilisation de tables de valeurs. C'est le thème de la rubrique coding de ce numéro de Quasar.

## Quoi ? Comment ?

Evidemment, à partir du moment où l'on décide d'utiliser un tableau, plusieurs questions se posent. Il faut choisir le format des valeurs qu'il contiendra, la place mémoire qu'on peut lui allouer, il faut également choisir la manière la plus judicieuse pour agencer ses données.

Je vous propose maintenant de voir cela plus concrètement sur des exemples.

## Premier exemple

Imaginez que vous voulez récrire le PLOT du Basic pour l'adapter à votre application, c'est à dire que vous souhaitez afficher un pixel à l'écran à partir de ses coordonnées. Vous devez donc vous débrouiller pour obtenir l'adresse à laquelle vous devrez écrire l'octet adéquat (comme Sheila). Je vais m'intéresser ici seulement au calcul de l'adresse de l'octet de début de ligne (à gauche de l'écran).

Vous avez à gauche le repère associé à l'écran. On souhaite obtenir l'adresse de l'octet de coordonnées (0,y). Du fait de la structure de la mémoire vidéo du CPC (vue dans d'anciens numéros de Quasar CPC), le calcul est le suivant :

$$\begin{aligned} \text{adr} &= \text{largeur} * y / 8 + (y \text{ and } 7) * 8000 \\ \text{adr} &= 80 * y / 8 + (y \text{ and } 7) * 8000 \\ &= 10 * y + (y \text{ and } 7) * 8 * 256 \\ &= (8+2) * y + (y \text{ and } 7) * 8 * 256 \end{aligned}$$

"largeur" est la largeur en octets (axe x) de l'écran.

Je prends pour largeur la valeur standard qui est 80. Si l'on écrit ce calcul en assembleur, voilà ce que ça peut donner :

C'est pas si mal, le calcul prend 28ps. Cela dit, il faut encore ajuster le résultat si l'écran n'est pas situé à l'adresse 8000.

Mais on se doute bien que l'on peut faire plus rapide grâce à une table précalculée.

Et en effet, si on stocke les adresses de début de ligne dans l'ordre, de la ligne 0 à 255 maximum, il est facile d'aller les récupérer en fonction d'un paramètre.

On a quand même recours ici à une astuce souvent très avantageuse pour les tables, il s'agit de la placer à une adresse mémoire de la forme &XX00 (c'est-à-dire dont l'octet de poids faible est nul). Ceci évite des opérations coûteuses en temps machine.

```

; l = Coordonnée y
ld a,l
ld h,0
add hl,hl
ld d,h
ld e,l
add hl,hl
add hl,hl
add hl,de ; hl = y * 10
and 7
add a,a
add a,a
add a,a ; a = (y and 7)*8
ld e,0
ld d,a ; de = a*256
add hl,de ; hl = résultat
    
```

Calcul de l'adresse sans table.



```

; l = Coordonnée y
Calcul xor a ; a = 0 ; Carry = 0
rl l ; l = l*2
adc a,table/256
ld h,a
ld a,(hl) ; hl = adresse paire
inc l ; donc "inc l" suffit
ld h,(hl)
ld l,a ; hl = résultat
    
```

```

Table .org &xx00
dw &C000
dw &C800
...
    
```

Cette routine nécessite une table de 512 octets mais on obtient le résultat cette fois en seulement 12ps. Le temps de calcul a donc été sensiblement diminué. Je précise un détail sur le programme : table/256 est calculé lors de l'assemblage et a comme valeur le poids fort de l'adresse à laquelle est stockée la table.

Là où du temps se perd dans la routine précédente, c'est lors de la multiplication par 2 de l qui permet

de déduire l'adresse de la table où se trouve la valeur qui nous intéresse. Cette multiplication est due à la taille des éléments de notre table qui sont des valeurs 16 bits et qui occupent donc deux octets. Pour s'en débarrasser il faut donc découper la table de word en deux d'octets.

C'est alors que j'introduis la dernière routine de ce premier exemple.

```

; l = Coordonnée y
Calcul ld h,table/256
      ld a,(hl)
      inc h
      ld h,(hl)
      ld l,a ; hl = résultat

Table Org &4500 ; adresse en &xx00
      db &00 ; poids faible
      db &00

      Org &4600 ; = &4500 + &100
      db &C0 ; poids fort
      db &C8
    ...

```

Les deux tables sont toujours placées à des adresses de poids faible nul et elles sont distantes de 256 octets (&100). La première table contient les poids faibles des éléments de la table et la deuxième les poids forts. Ainsi un simple "ld" permet de trouver l'adresse de poids faible et un simple "inc h" passe aux poids forts !

Le résultat ne prend plus que 8ys pour le même résultat, comme quoi, ça vaut le coup de chercher un peu !

## Deuxième exemple : trigonométrie

Oui, vous avez bien lu. Dans des numéros passés de Quasar on vous a parlé de 3D avec des jolies formules comportant parfois des sinus et cosinus. Eh bien si vous comptez faire de la "3D temps réel", comme on dit, vous avez plus besoin de vitesse que de précision, l'utilisation de tables est toute indiquée.

Avant de foncer tête baissée, intéressons-nous aux spécificités des fonctions sinus et cosinus. Déjà, elles peuvent prendre des valeurs positives ou négatives réelles, ce qui nous déplaît plutôt. Ensuite, elles sont périodiques et le cosinus peut être déduit du sinus simplement. Pour finir, les valeurs prises sont réelles mais bornées, on sait en effet que les fonctions sinus et cosinus ne retournent que des valeurs comprises entre -1 et +1, ensemble symétrique par rapport à 0.

La suite dépend de la précision nécessaire à vos calculs. La table sera la valeur du sinus sur toute une période pour chaque valeur d'angle. Certes on pourrait reconstituer le sinus complet à partir d'un quart ou d'une demi-période mais cela demanderait des opérations

supplémentaires.

Je propose de stocker chaque valeur de sinus sur seulement 1 octet, en binaire signé (aussi appelé complément à 2). Donc on place finalement dans la table la valeur de  $127 * \sin(\text{angle})$ , tant pis pour la valeur -128, restons sur un intervalle symétrique. Si vous souhaitez une précision plus grande vous pouvez allouer 2 octets, vous vous retrouvez alors dans le cas du premier exemple.

La valeur de  $\sin(\text{angle})$  est réelle mais l'angle lui aussi est réel. Dans beaucoup de cas une précision au degré près est amplement suffisante. Mais si l'on pense au programme il est nettement plus intéressant de choisir une puissance de 2 au lieu de 360, et je pense en particulier à la valeur 256.

Le nombre 256 est miraculeux dans ce cas car il permet de traduire sans effort le caractère périodique des fonctions périodiques telles que le sinus.

Démonstration :

- En radians :  $\sin(a) = \sin(a \pm 2\pi)$
- Avec notre choix d'angle :  $\sin(a) = \sin(a \pm 256)$   
comme  $280 - 256 = 24$  :  $\sin(280) = \sin(24)$

Quand on va ajouter, soustraire, multiplier... bref, faire des calculs en 8 bits sur les angles, le modulo 256 va se faire naturellement, on n'aura plus qu'à aller chercher la valeur du sinus parmi nos 256 octets sans se poser plus de questions.

Au niveau assembleur on prendra soin de placer la table sur une adresse de poids faible nul et le code résultant est alors ridiculement simple :

```

; l = angle
Calcul ld h,table/256
      ld a,(hl) ; a = résultat

Table Org &xx00
      ds 256 ; table de sinus

```

Pour ce qui est du cosinus, comme on l'a déjà dit, on peut le déduire de la table de sinus :

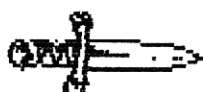
- En radians :  $\cos(a) = \sin(a + \frac{\pi}{2})$
- Avec notre choix d'angle :  $\sin(a) = \sin(a + 64)$

Il suffit donc d'ajouter 64 à l'angle et d'appeler la même routine que précédemment pour obtenir le cosinus.

## Conclusion

Un dernier mot avant de terminer cet article. Il vous faut à moment donner générer les tables, le Basic s'y prête bien ; vous pouvez calculer la table à une adresse convenue ou générer un morceau de source assembleur (en ASCII pour Maxam) à inclure dans le source principal. Vous pouvez aussi générer la table en asm en utilisant les vecteurs mathématiques du système ou non !

Zik



# DOSSIER



Après avoir fait un bref rappel sur l'affichage dans le dernier numéro, intéressons-nous aujourd'hui à la structure de nos niveaux et à plusieurs méthodes qui vont nous permettre de tester les collisions des sprites avec le décor puis avec d'autres sprites qui se déplacent.

Mais avant toute chose, petite correction d'une coquille parue dans le source de masquage au pixel du dernier numéro : il faut incrémenter l'adresse écran après avoir envoyé l'octet masqué et non après avoir récupéré de décor ! Ils n'ont vraiment aucune conscience professionnelle chez Quasar CPC et ne relisent même pas les articles qu'ils reçoivent ! (NDLR : en fait nous préférons laisser des erreurs dans les articles des intervenants extérieurs afin que les lecteurs notent bien la supériorité des rédacteurs titulaires...).

## STRUCTURER POUR MIEUX RÉGNER

Afin de simplifier le travail du graphiste et de limiter la mémoire occupée par le niveau, nous découperons les graphismes des éléments de décor en rectangles identiques que nous rassemblerons ensuite pour créer nos décors. De cette façon, à partir d'un minimum de graphismes, nous pourrons créer un nombre presque infini de décors ce qui est quand même bien plus efficace que dessiner chaque décor comme un écran à part entière. Bien, nous aurons besoin de deux choses pour pouvoir afficher nos décors :

- un tableau (à deux dimensions) pour chaque écran, et où chaque valeur (de 8 bits chacune, cela devrait suffire) représentera le numéro du rectangle à afficher, d'une taille de :

$$\text{lon\_ecr} = (\text{larg\_e}/\text{larg\_r}) \times (\text{haut\_e}/\text{haut\_r}) \text{ octets}$$

(larg\_e=largeur de l'écran, larg\_r=largeur des rectangles, haut\_e=hauteur de l'écran, haut\_r=hauteur des rectangles. Chaque couple (larg\_e, larg\_r) et (haut\_e, haut\_r) doit être exprimé dans la même unité (word, octet, ligne...)),

- les rectangles placés en mémoire les uns à la suite des autres.

La correspondance entre les valeurs figurant dans le tableau et l'adresse du rectangle à afficher se

calculera simplement :

$$\text{adr\_r} = \text{adr\_dr} + (\text{valeur} \times \text{larg\_r} \times \text{haut\_r})$$

(adr\_dr=adresse de début des rectangles, larg\_r et haut\_r doivent cette fois-ci être exprimés en octets).

Selon la catégorie de jeu que vous souhaitez réaliser, votre niveau se présentera sous la forme d'un seul tableau ou bien d'un tableau par écran (ou par zone), qui sera lui-même rempli de valeurs permettant d'afficher le décor ligne) ligne, colonne par colonne, etc...

## Collisions et coordonnées

Maintenant que nous savons créer nos niveaux, testons les collisions entre les sprites qui peuvent se déplacer. On dit qu'il y a collision lorsque deux sprites, définis par les paramètres (x1, y1, larg1, haut1) pour le premier et (x2, y2, larg2, haut2) pour le second, se chevauchent. Pour cela, ces inégalités doivent toujours être vérifiées :

$$\begin{aligned} x1 + \text{larg1} &> x2 \\ x2 + \text{larg2} &> x1 \\ y1 + \text{haut1} &> y2 \\ y2 + \text{haut2} &> y1 \end{aligned}$$



Cette méthode est toutefois assez rudimentaire et ne permet d'avoir un rendu parfait que si les sprites sont rectangulaires : ce n'est pas parce que les matrices rectangulaires qui définissent les sprites se chevauchent que les sprites se chevaucheront eux aussi !

Dans l'absolu, il faudrait définir un couple (xi, largi) pour chaque ligne i de nos sprites et un couple (yj, hautj) pour chaque colonne de pixels j. Pour obtenir un résultat parfait, il suffirait ensuite d'appliquer les 4 tests précédents pour chaque ligne et chaque colonne de pixels. Mais comme cela serait bien trop long, nous allons ruser un peu en effectuant la moyenne de tous les xi, yj, largi, hautj. Grâce à cette approximation, nous gagnons énormément de temps machine et le résultat sera tout à fait satisfaisant. On pourra même améliorer cette technique si plusieurs zones xi, yj, largi, hautj se distinguent sur les sprites (fusil du personnage qui dépasse, coup de pied dans un jeu de

combat...). Dans ce cas, il faudra faire la moyenne des  $x_i, y_j, larg_i, haut_j$  pour chaque zone et appliquer les 4 sempiternels tests donnés dans le paragraphe précédent pour chaque zone.

## Dans Le décor

Pour tester les collisions avec le décor, une première technique consisterait à définir avec des segments l'aire non accessible, et d'appliquer des tests similaires (dans le principe) à ceux déjà évoqués plus tôt. Mais cela monopoliserait encore de la mémoire supplémentaire (4 octets par segment !) et nous obligerait à tester à tout moment s'il y a collision avec chaque segment, ce qui ne nous assurerait pas d'avoir un temps machine constant selon qu'il y ait 2 ou 20 segments à tester. Oublions cette mauvaise méthode.

Seconde technique : des tests de couleurs. En réservant une ou plusieurs encres au décor inaccessible, la simple lecture d'un ou plusieurs (ce nombre sera logiquement inversement proportionnel au nombre d'encres réservées !) octets composant le décor nous dira immédiatement si oui ou non il y a collision. Comme toutes les techniques utilisant stratégiquement les encres elle n'est correctement exploitable qu'en mode 0.

Attardons-nous plus en détail sur la troisième méthode qui est à mon avis la plus performante tant au niveau mémoire que temps machine. Cette fois, nous raisonnerons sur le tableau qui nous a permis de créer le niveau. En stockant en mémoire tous les rectangles accessibles, puis à partir du rang  $n$ , tous les rectangles inaccessibles, nous saurons que si le personnage chevauche un rectangle possédant un numéro supérieur ou égal à  $n$ , alors il y aura collision. La conversion des coordonnées  $(x,y)$  de notre personnage en adresse du tableau où se trouve la valeur à lire se fait de cette façon :

$$adr\_tab = (y / haut\_c) \times (larg\_e / larg\_c) + (x / larg\_c)$$

Si le déplacement s'effectue horizontalement, il faudra tester  $(haut\_p / haut\_c)$  valeurs des rectangles en colonne dans le tableau. Si le déplacement est ver-

tical, il faudra tester  $(larg\_p / larg\_c)$  valeurs en ligne (le couple  $(larg\_p, haut\_p)$  représente la largeur du personnage et la hauteur).

Quoi qu'il en soit, au moins d'être dans le cas avantageux où  $haut\_p$  modulo  $haut\_c$  est égal à 1, le nombre de valeurs à tester en colonne pour un déplacement horizontal pourra varier puisque la dernière ligne du personnage pourra dépasser sur un nouveau rectangle alors que la première ligne restera sur le même (faites un petit dessin pour mieux comprendre). Plutôt que de longues explications, voici une formule qui calcule si on doit tester une valeur supplémentaire :

$$\text{si } y \text{ modulo } haut\_c > haut\_c - (haut\_p \text{ modulo } haut\_c) \\ \text{alors } nbr\_r = nbr\_r + 1$$



On procède exactement de la même manière pour tester les collisions d'un déplacement vertical (si  $larg\_p$  modulo  $larg\_c$  est différent de 1 !):

$$\text{Si } x \text{ modulo } larg\_c > larg\_c - (larg\_p \text{ modulo } larg\_c) \\ \text{Alors } nbr\_r = nbr\_r + 1$$

Voilà, vous pourrez maintenant améliorer cette méthode en associant à chaque rectangle un tableau où chaque bit permettrait de savoir si oui ou non le pixel est accessible, ce qui aurait pour conséquence d'affiner la méthode, ne donnant ainsi plus l'impression de travailler avec de gros blocs.

Hicks

NDLR : profitons de cet espace libre pour vous rappeler quelque chose d'essentiel : nous avons besoin de vous ! Oui, un fanzine, s'il vit par ses rédacteurs, vit avant tout pour ses lecteurs ! Toutefois, force est de constater que vos réactions à nos différents articles sont de moins en moins nombreuses... aussi, il nous est difficile d'apprécier réellement ce que vous attendez de nous. Alors, écrivez-nous ! Conseillez-nous, critiquez-nous, insultez-nous ! Mais exprimez-vous ! Depuis quelques temps, outre certaines rubriques, nous avons réellement l'impression d'avancer à taton... alors, plutôt que de nous laisser choisir pour vous, n'hésitez pas à nous faire part de vos envies !

Devant de flot incessant de demandes concernant l'installation d'un lecteur 3"1/2 en lieu et place du lecteur 3", je me suis donc décidé à vous écrire un petit article sur la question. Nous avons déjà évoqué l'installation d'un lecteur 3"1/2 en tant que deuxième lecteur dans notre numéro 14, aussi, cette fois-ci, je me limiterai à ce qu'il est nécessaire de savoir pour une installation en interne. Mais avant tout, bien que j'en ai déjà parlé à maintes reprises tant dans ces pages que lors des meetings, voici un petit discours sur la façon de gérer votre lecteur 3"1/2 qu'il soit en interne ou en externe.

## Le CPC sait gérer Les Lecteurs 3"1/2

Force est de constater que vous êtes une majorité à limiter l'utilisation des lecteurs 3"1/2 en "compatibilité 3 pouces". C'est-à-dire que vous utilisez un interrupteur pour changer de face manuellement et formatez vos disquettes au lamentable format Amsdos... Vous vous retrouvez ainsi avec des disquettes de 2 x 180ko alors qu'un lecteur 3"1/2 permet la gestion linéaire de disquettes de 800ko !

Ça y est, le petit blond commence à la ramener. Oui il existe une bidouille permettant de monter à 256ko par face avec Copyluck, mais ça ne marche ensuite qu'en lecture et on est bien loin des 800ko linéaires ! Oui, l'Amsdos est incapable de gérer correctement les disquette 3"1/2... eh bien il serait temps d'en changer ! Il existe de nombreuses ROM alternatives dont la plus performante est sans nul doute The Rodos System, mais elles ont toutes deux défauts majeurs : elles ne sont pas compatible Amsdos et elles nécessitent l'achat d'une Ramcard ou d'un Romboard.

Bon, la solution "idéale" existe, c'est Parados. Il s'agit d'une ROM 100% compatible Amsdos et a l'avantage de savoir gérer correctement les lecteurs 3"1/2 de manière complètement transparente ! Une fois cette ROM installé, fini le switch de face ! Cette ROM existe depuis 1993 et pourtant quasiment personne ne l'utilise. A la redac', tous nos CPCs en sont équipés depuis 3 ans et nous n'avons jamais eu le moindre problème !!! Alors pourquoi ne bénéficiez-vous pas vous aussi des avantages de cette ROM ? J'ai déjà réussi à convaincre Shap il y a peu... pourquoi pas vous ?

## Les fonctionnalités

Comme je l'ai déjà dit, Parados se substitue complètement à l'Amsdos tant en terme de RSX, que de vecteurs, ou même d'appels directs en ROM. Ça signifie que même les programmes mal codés qui font des appels directs en ROM au lieu de passer par les vecteurs systèmes seront à même de tourner sur les nouveaux formats ! D'ailleurs, parlons-en des nouveaux formats... Il y en a une flopée. La plupart sont là simplement pour la compatibilité avec d'autres ROM disques et les formats qui nous intéressent réellement sont les formats 800ko qui sont déclinés en deux versions : 128 entrées ou 256 entrée. Le premier a des tables d'allocation de 2k et le second de 4k, à vous de choisir ! De toute façon, à partir du moment où votre disquette est formatée tout est transparent (de la même manière que la gestion des format SYSTEM, DATA et IBM sous Amsdos).

Eh, le petit blond à lunettes pose une question sensée : comment formater ces disquettes ? Eh bien, lorsque j'ai dit que Parados était 100% compatible au niveau des RSX, j'ai un peu menti... la RSX ÜDRIVE fonctionne un peu différemment puisque lancée sans paramètres elle ne renvoie plus un message d'erreur mais vous fait entrer dans un utilitaire disque très pratique. Il vous permettra de formater, copier, renommer... Bref, de gérer vos disquettes (à la trappe Disc'O'Magic ou Crime !).

## Comment l'installer

Ça y est, vous êtes convaincu ? Passons à l'installation. Dès lors, nous avons deux possibilités. En ROM externe numéro 6 ou en ROM interne numéro 7 (à la place de l'Amsdos). Je vous conseille la seconde car c'est la seule qui soit 100% compatible en cas de programme sautant en ROM. Eh oui, si vous avez installé un tel programme sur une disquette 800k et qu'il saute en ROM 7 numéro il va taper dans l'Amsdos et non dans Parados donc : Read Fail assuré !

Je ne vais donc vous parler ici que de l'installation en ROM 7 à la place de l'Amsdos. Nous avons trois cas de figure selon que vous possédez un CPC664, 6128, ou 6128plus. Si vous avez un CPC664 et pas de Ramcard, il vous suffit de vous faire programmer une EPROM de 16k

avec le programme de Parados et la mettre à la place de la ROM CP/M sur la carte mère. En effet, sur CPC664, la ROM CP/M est sur support, elle est donc facilement interchangeable. Si vous avez une Ramcard et que vous n'avez pas la possibilité de vous faire programmer une EPROM alors retirez simplement la ROM Amsdos de la carte mère et programmer Parados à l'emplacement de la ROM 7 sur votre Ramcard ! Simple non ?

Si vous avez un CPC6128, ça se complique un peu. En effet, la ROM Amsdos est soudée sur la carte mère... Si vous en avez la possibilité le mieux est de la dessouder et de placer un support ; vous vous retrouvez alors dans le même cas de figure que les possesseurs de 664. Une autre solution consiste simplement à invalider la ROM interne pour pouvoir programmer une ROM 7 sur votre Ramcard : jetez un coup d'oeil dans le numéro 16 de Quasar CPC, tout y est expliqué en détail. Si aucune de ces deux solutions ne vous convient alors vous ne pourrez utiliser Parados qu'en ROM numéro 6 ce qui est nettement moins intéressant.

Si vous possédez un 6128plus... Inutile de chercher la ROM Amsdos sur la carte mère : il n'y en a pas ! La ROM se trouve en effet dans la cartouche ; il n'est donc pas possible de la substituer physiquement comme sur CPC664 ou CPC6128. Heureusement, elle est très facilement invalidable : prenez votre Ramcard, programmez y Parados en ROM numéro 7, et hop, votre 6128plus va automatiquement invalider la ROM Amsdos de la cartouche pour prendre la vôtre : c'est beau le progrès !

Pour en finir avec Parados, je dois vous signaler qu'il en existe deux versions : la 1.0 (de 1993) et la 1.1 (de 1997). Vous conseille la première car la seconde n'est plus compatible avec les disquettes DATA gérées à l'aide du switch de face manuel. De plus, la version 1.1 ne corrige qu'un bug d'affichage mineur dans le gestionnaire de fichiers intégré.

Maintenant que vous avez installé Parados, nous allons lui donner de quoi faire en remplaçant le lecteur 3" interne par un lecteur 3"1/2.

## Un Lecteur 3"1/2 en interne

Il existe une méthode sale qui consiste à repiquer le connecteur externe pour ramener une nappe en interne puis à forcer le signal Drive 1 Select : elle n'est pas digne des perfectionnistes que nous sommes ! Ce que je vais vous expliquer ci-après, c'est comment modifier la nappe 3" et le connecteur d'alimentation pour avoir un véritable lecteur 3"1/2 interne. Ainsi, votre connecteur pour un lecteur externe restera disponible pour un second lecteur 3"1/2 (ou 3") : voir Quasar CPC numéro 14...

## La fiche d'alimentation

À première vue, le connecteur d'alimentation de notre bon vieux lecteur 3" est identique à celui des lecteurs 3"1/2... mais, comme le dit l'adage, l'habit ne fait pas le moine ! Eh oui, même si le type de connecteur est identique, son brochage est différent.

Sur CPC ancienne génération, nous avons 4 fils qui arrivent : un orange (le 5V), un rouge (le 12V) et deux noirs (les masses). Eh bien ceux-ci sont tout bonnement inversés par rapport à la connexion sur 3"1/2 ; de ce fait, il vous faut échanger les fils rouge et orange sur le connecteur. Pour ce faire, rien de plus simple ; sortez les "picots" du boîtier en plastique blanc (il suffit de soulever la languette sur le dessus du connecteur puis tirer sur le fil) puis ré-engagez-les au bon endroit (dans tous les cas faites attention de ne pas casser les languettes, sinon les picots ne tiendraient plus). Voilà, désormais, un code de couleurs pas très orthodoxe votre connecteur apporte maintenant les 5V et le 12V où il faut.

Sur 6128plus, la situation est identique sauf que nous avons uniquement un fil rouge (le 5V) et les deux masses (les fils noirs). Ceci étant, notre 5V n'arrive toujours pas au bon endroit et il faut le déplacer à l'autre bout du connecteur. Ah, le petit blond à lunettes est sorti de sa torpeur : eh, oui, il n'y a pas de 12V sur les CPC+, il faut donc vous assurer que votre lecteur 3"1/2 n'en a pas besoin (tous les lecteurs récents n'utilisent que le 5V).

## Le connecteur principal

Les choses seraient trop simples si le standard 3" étaient effectivement un standard... Notre connecteur n'a pas grand chose à voir avec ce dont un lecteur 3"1/2 a besoin ; de plus, celui-ci a été monté différemment sur CPC et sur CPC+. Enfin, dans notre malheur, l'ordre des signaux est le même à une exception près. Il nous suffit donc d'ajouter le connecteur HE10 à 34 broches (le standard de connectique sur 3"1/2) sur notre nappe, quelques centimètres avant le connecteur HE10 à 26 broches (le standard de connectique sur 3"). Bien sûr, si vous le souhaitez, vous pouvez également franchement retirer le connecteur d'origine et mettre le nouveau à la place ; ceci vous permettra d'avoir une plus grande longueur de nappe pour arriver jusque derrière le lecteur 3"1/2 (notamment sur CPC+ où Amstrad a été particulièrement radin). Ceci ayant été précisé, nous allons enchaîner par des petits schémas, car il est bien connu qu'un bon dessin vaut mieux que des tonnes de texte ! Mais avant tout, il va vous falloir tourner la page...

## Les derniers trucs

Vous êtes maintenant en mesure de remplacer votre bon vieux 3" par un 3"1/2 ! Ça y est, le petit blond à lunettes est encore en train de critiquer... Non, je ne parlerai pas du problème du signal "Ready" (broche 1 du lecteur de disquettes 3"1/2) ni de l'interrupteur de changement de face !

Concernant le signal "Ready", je vous conseille de vous procurer un lecteur 3"1/2 capable de le générer. En fait, 99% des lecteurs actuellement en vente chez les épiciers génèrent un signal "Disc Change" en broche 1 ; or il s'agit de la norme PC, qui diffère de la norme qui avait été choisie par toutes les autres marques (Amstrad, Atari, Amiga)... Heureusement, certains lecteurs (notamment chez Sony et TEAC) sont paramétrables ; vous pourrez ainsi les configurer pour qu'ils soient compatibles avec le CPC. Si vous ne pouvez pas vous procurer de tels lecteurs il ne vous reste plus qu'à lire Quasar CPC 14 pour vous bricoler votre propre signal "Ready".

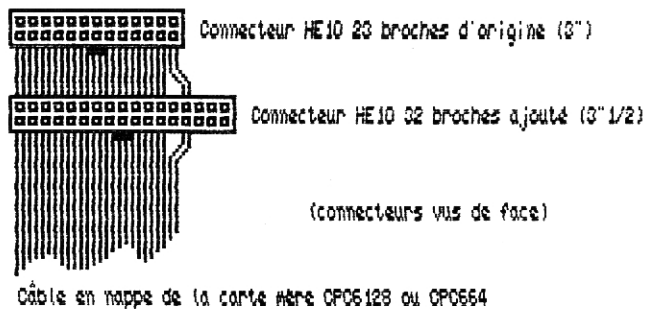
Pour finir, en ce qui concerne l'interrupteur de changement de face, je vous répète qu'il est totalement inutile à partir du moment où vous utilisez Parados. Néanmoins, si vous désirez l'installer, il vous faut savoir qu'il sera obligatoirement commun aux deux lecteurs puisque tous les signaux sont partagés... Pour plus de détails, je vous renvoie là aussi à notre numéro 14.

## Vous savez tout

Voilà, je crois que tout a à présent été dit sur le montage des lecteurs 3"1/2 sur CPC. Si vous avez des questions ou des suggestions, n'hésitez pas à nous contacter. Je vous précise également, suite à de nombreuses questions de votre part, que ce type de montage est aussi réalisable dans les lecteurs FDI Amstrad qui étaient livrés avec l'extension DD1 pour les CPC464. De plus, sachez que la ROM Amdos est sur support dans ces cartes, il est donc très simple de la remplacer par une EPROM Parados.

## Commande groupée de Lecteurs 3"1/2

Pour en finir, si vous êtes intéressés, vous avons le projet de faire une commande groupée de lecteurs 3"1/2 de marque TEAC qui ont l'avantage d'être compatibles CPC grâce à une configuration du signal "Ready". Ces lecteurs sont sensiblement plus cher que les lecteurs PC standards (~150F), mais si vous en voulez un contactez-nous au plus vite !



Câble en nappe de la carte mère CPC6128 ou CPC654

Le premier schéma vous montre comment ajouter le connecteur nécessaire au branchement du lecteur 3"1/2. Comme vous pouvez le voir, celui-ci est en parallèle avec le connecteur d'origine à l'exception des deux fils les plus à droite qui sont décalés de deux broches vers la droite. Enfin, le renflement noir sous les connecteurs représente le détrompeur (à positionner correctement pour pouvoir se connecter !).

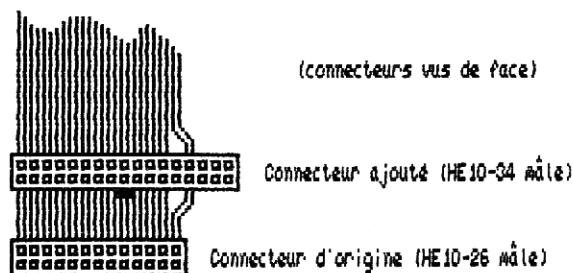
Oh, un dernier petit détail, voici les références exactes du connecteur qu'il vous faut acheter : connecteur HE10 34 broches mâle à sertir sur câble en nappe. Pour le sertissage, nul besoin d'une pince à sertir, un étaux fera parfaitement l'affaire ! Mais faites bien attention au positionnement des fils de la nappe sur les contacts en V, surtout pour les deux fils décalés.

## La petite différence sur 6128plus

Sur les 6128plus, le connecteur a été monté à l'envers sur la nappe (allez savoir pourquoi) qui, de plus, est extrêmement courte ! Aussi, je vous conseille de la remplacer par une nappe un peu plus longue pour ne pas être embêté par la suite. Lorsque vous dessertirez la nappe côté carte mère soyez soigneux pour ne pas abîmer les contacts en V et le capot du connecteur.

Et puis comme je suis bon prince et que je veux vous éviter toute gymnastique de l'esprit, voici le schéma vous montrant comment ajouter votre connecteur HE10-34 sur la nappe 3" des CPC+ :

Nappe allant vers la carte mère du 6128plus





# HELPS

## La Abadía del Crimen 3/3

Nous sommes arrivés au dernier épisode des aventures de Guillaume et Adson ; nous approchons de la solution du mystère de l'abbaye... mais faites attention, car c'est beaucoup plus facile à lire qu'à faire !

### Cinquième jour

#### ~prime

Tous les jours commencent avec la messe de Prime, mais pendant que Guillaume arrive à l'église, Severin s'approche et parle :

Severin : "ESCUCHAD HERMANO, HE ENCONTRADO EN EXTRAÑO LIBRO EN MI CELDA." - Ecoutez frère, j'ai trouvé un livre étrange dans ma chambre.

Un livre étrange ? Pas de doute, c'est le livre qui était sur la table de Venance, volé par Berenger la deuxième nuit et disparu après la mort de l'apprenti bibliothécaire. Mais ce n'est pas le moment de parler car la messe doit commencer.

#### ~tierce

L'abbé demande l'attention de Guillaume :

Abbé : "BERNARDO ABANDONARA HOY LA ABADIA." - Bernard quitte l'abbaye aujourd'hui.

Mais pendant que l'abbé parle, un moine marche vers une place inhabituelle : Malachias, le bibliothécaire, est en train de suivre Severin ! Il semble qu'il ait lui aussi entendu les nouvelles sur l'étrange livre trouvé dans l'hôpital.

Quand l'abbé a fini de parler, Guillaume se rend à l'hôpital. Mais quelque chose va mal car le bibliothécaire sort de l'hôpital sans dire un mot lorsque Guillaume et Adson y arrivent. Il s'en va à la bibliothèque. La porte de l'hôpital est fermée.

#### ~sexe

Heure du repas. Bernard vient de quitter l'abbaye, dès lors, quatre moines seulement occupent la table du réfectoire : l'abbé, Guillaume, Adson... et Severin ? Severin ne vient pas et l'abbé décide de commencer le repas sans lui.

#### ~neufvaine

L'absence de Severin est décidément trop étrange :

Abbé : "VENID, FRAY GUILLERMO, DEBEMOS ENCONTRAR A SEVERINO." - Venez, frère Guillaume, nous devons trouver Severin.

L'abbé, Guillaume et Adson marchent vers l'hôpital.

L'abbé frappe alors violemment à la porte de l'hôpital. Et ensuite, le silence.

Abbé : "DIOS SANTO... HAN ASESINADO A SEVERINO Y LE HAN ENCERRADO." - Dieu saint... Ils ont assassiné Severin et l'ont enfermé !

#### ~vêpres

Les cloches de la messe des vêpres sonnent et les trois moines marchent vers l'église. Une fois de plus, le bibliothécaire ferme la moitié Nord de l'abbaye. Mais en arrivant à l'église il parle :

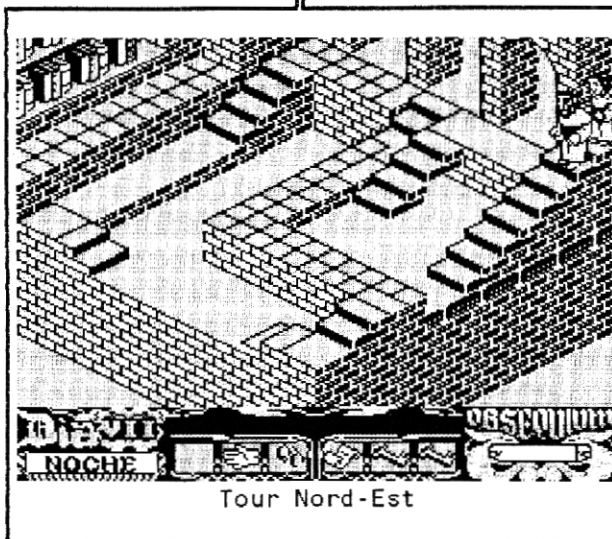
Malachias : "ERA VERDAD, TENIA EL PODER DE MIL ESCORPIONES". C'est vrai, il avait le pouvoir de mille scorpions. Malachias quitte alors notre monde.

Abbé : "MALAQUIAS HA MUERTO" - Malachias est mort.

#### ~nocturne

Après un jour si terrible, avec deux morts, dormir serait la meilleure option, mais nous sommes trop proches de la solution et il n'y a pas de temps à perdre : c'est le moment de visiter la bibliothèque. C'est comme ça que Guillaume et Adson traversent le passage secret, la porte que Malachias gardait tous les jours sans repos, et montent l'escalier de la bibliothèque.

La bibliothèque est doublement dangereuse : c'est un labyrinthe et il n'y a pas de lumière ! Heureusement,



Tour Nord-Est

pour le premier danger, regardez la carte ci-contre ; et pour le second, la lampe à huile prise par Adson le troisième jour illuminera un peu le chemin.

Pour que personne ne se trompe, j'ai seulement dessiné les deux chemins utiles (les autres chemins sont faux), dont le supérieur doit être pris dès maintenant pour arriver à la tour nord-est. Et voilà, ce qui se trouve ici : les lunettes de Guillaume !

La lampe à une durée limitée, alors revenez vite au scriptorium. Sur la table de Malachias, il y a maintenant une clef que Guillaume doit prendre. On verra plus tard la porte qu'elle ouvre. Il est très tard, Guillaume et Adson reviennent au dortoir.

## Sixième jour

### ~tièrce et neubaine

Abbé : "MAÑANA, ABANDONAREIS LA ABADIA". - Vous quitterez l'abbaye demain.

Il reste peu de temps pour achever l'aventure. Guillaume se rend vers la chambre de l'abbé (faites attention, s'il vous trouve dans sa chambre, le jeu est fini !) et ouvre la porte avec la clef prise la cinquième nuit : voilà le manuscrit de Venance. Alors, Guillaume lit le manuscrit à l'aide de ses lunettes :

Guillaume : Secretum finis africae, manus supra XXX idolum age primum et septimum de quatuor.

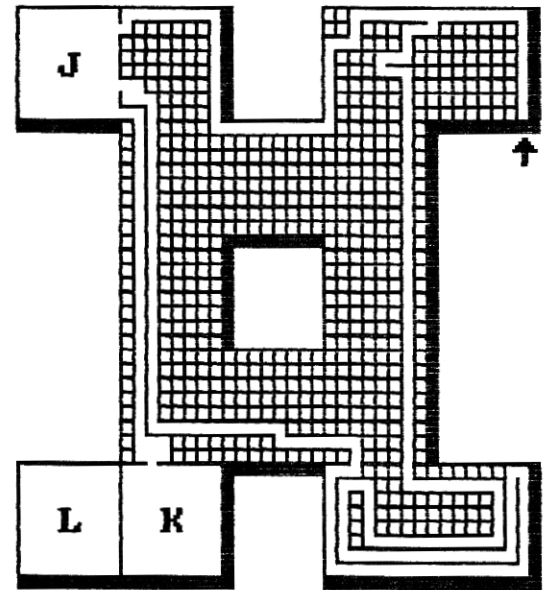
Un peu de latin est toujours utile : le secret du fin d'Afrique, la main sur l'idole XXX touche le premier et le septième de quatuor. "XXX" est choisi aléatoirement à chaque nouvelle partie : "LXX", "XIX" et "XXI" sont les combinaisons possibles.

Maintenant, Guillaume ouvre la porte de l'hôpital avec la dernière clef ramassée (prouvant que Séverin fut assassiné et enfermé par Malachias, car la clef de l'hôpital était sur la table du bibliothécaire) et doit prendre un objet : le gant. La nuit approche mais il y a encore une chose à faire : aller à la cuisine. Une nouvelle lampe à huile est sur la table de la cheminée, et Adson doit la prendre.

### ~nocturne

Arrêtons-nous maintenant pour nous assurer que tout va bien. Adson doit avoir la clef du passage secret et la lampe à huile ; Guillaume doit avoir le gant, les lunettes, le parchemin de Venance, la clef de l'abbé et la clef de Séverin. Tout va bien à bord ?

Guillaume et Adson doivent monter une fois de plus à la bibliothèque ; mais cette fois il est possible de



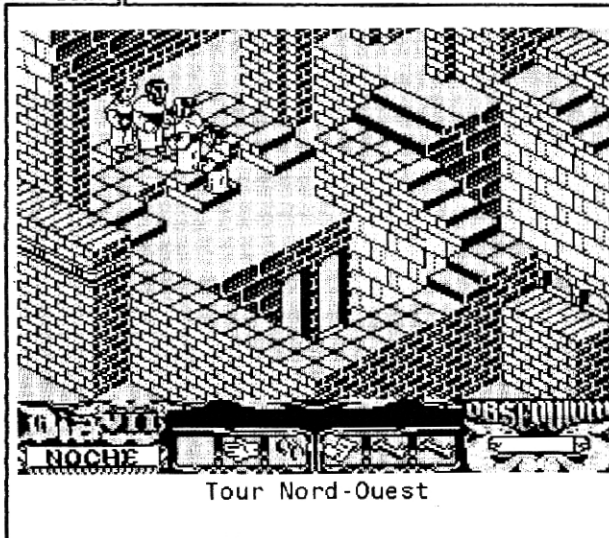
J: Tour nord-est  
K: Tour nord-ouest (miroir)  
L: Tour nord-ouest (final)

traverser les portes de la moitié nord car après la mort de Malachias, personne ne les ferme. Mais l'abbé peut vous trouver, alors utilisez le passage.

Le second chemin de la bibliothèque est assez long et il est possible qu'Adson vous fasse remarquer que la lampe s'épuise peu à peu ; mais il y a assez d'huile pour arriver à la tour nord-ouest.

Un miroir reflète les images de Guillaume et Adson ; trois petits escaliers montent vers la glace. Vous vous souvenez du parchemin de Venance ? Guillaume peut le

reliure en posant les lunettes et en les reprenant. Bon, le bon escalier est celui pointé par la lettre "T" dans le "XXX". Et maintenant, pressez les touches Q et R (premier et septième de quatuor, "Q" et "R"). Après un terrible bruit, le miroir s'ouvre.



Tour Nord-Ouest

## Septième jour

### ~nocturne

Un moine attend derrière une table :

Georges : "SOIS VOS, GUILLERMO... PASAD, OS ESTABA ESPERANDO TOMAD, AQUI ESTA VUESTRO PREMIO." - C'est vous, Guillaume... Approchez, je vous attendais. Prenez, voilà votre récompense.

Georges : "FUE UNE BUENA IDEA, ¿VERDAD? PERO YA ES TARDE !" - C'était une bonne idée, n'est-ce pas ? Mais c'est trop tard !

Alors, le moine aveugle prends le livre et éteint la lumière de la pièce, heureusement, Adson porte toujours la lampe. Georges s'enfuit par la porte du miroir, et Guillaume et Adson courent à sa poursuite. Quand ils arrivent à la tour nord-est, Georges s'arrête :

Adson : "SE ESTA COMIENDO EL LIBRO, MAESTRO !" - Il mange le livre, maître !

Fin

Empoisonné par les feuilles du livre, Georges eut encore la force de frapper Adson et de jeter la lampe à huile sur quelques papiers...

Le petit feu déclenché devint rapidement un grand incendie, réduisant l'abbaye en cendres. Guillaume et Adson quittèrent les ruines et se rendirent à Munich, où finalement, ils se séparèrent.

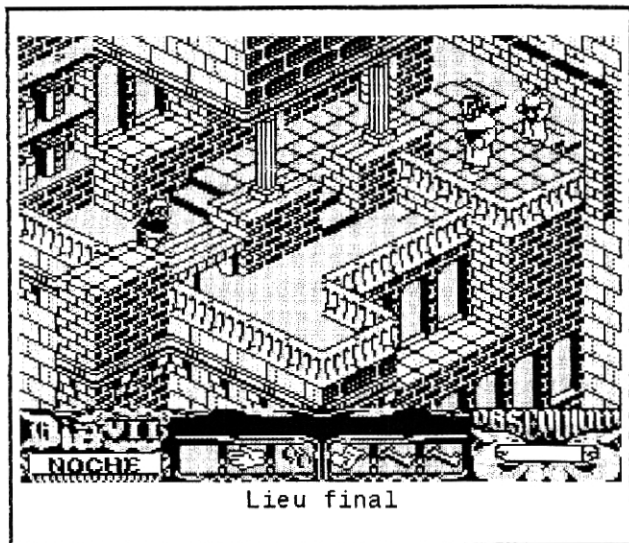
Adson ne vit alors plus Guillaume. Beaucoup d'années plus tard, Adson décida d'écrire tout ce qui s'était passé ; le manuscrit d'Adson Melk fut trouvé par un écrivain italien au XXème siècle, et quelques programmeurs espagnols transformèrent son roman en jeu pour Amstrad CPC en 1987.

CNGSOFT

NDLR : "le Nom de la rose" est un roman écrit en 1980 par Umberto Eco, un écrivain italien né à Alexandrie en 1932. Ce roman mêle questions théologiques et intrigue policière, il a reçu en 1981 deux des plus grandes récompenses italiennes de littérature.

Le réalisateur français Jean-Jacques Annaud en a fait un film à distribution internationale en 1986.

Pour avoir plus d'informations sur Umberto Eco et sur ces nombreux écrits vous pouvez vous rendre sur son site à l'Université de Bologne : <http://www.dsc.unibo.it/dipartimento/people/eco>  
Et voici l'adresse d'un très bon site sur Eco : [http://www.libyrinth.com/eco/eco\\_intro.html](http://www.libyrinth.com/eco/eco_intro.html)



Lieu final

Et il lui montre le livre responsable de tous les désastres de l'abbaye. Guillaume le prend et Georges parle :

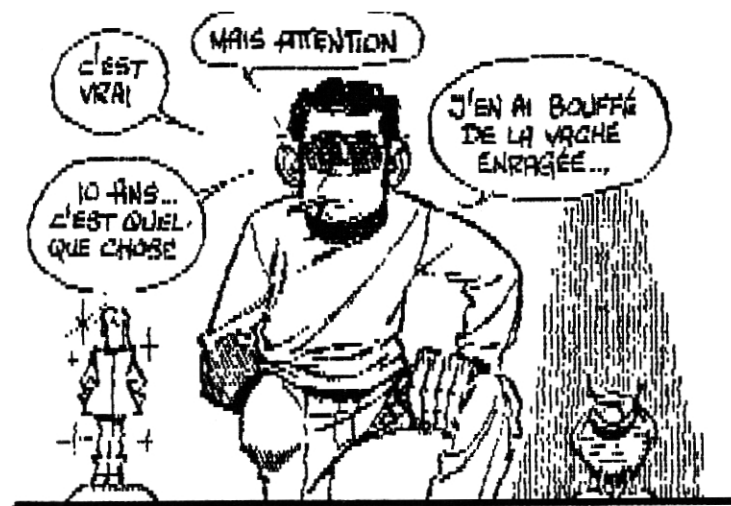
Georges : "ES EL COENA CIPRIANI DE ARISTOTELES. AHORA COMPRENDEREIS PORQUE TENIA QUE PROTEGERLO. CADA PALABRA ESCRITA POR EL FILOSOFO HA DESTRUIDO UNE PARTE DEL SABER DE LA CRISTIANDAD. SE QUE HE ACTUADO SIGUIENDO LA VOLUNTAD DEL SEÑOR... LEEDLO, PUES, FRAY GUILLERMO. DESPUES TE LO MOSTRARE A TI MUCHADO." - C'est le "Coena Cipriani" d'Aristote. Vous comprendrez maintenant pourquoi je devais le protéger. Chaque mot écrit par le philosophe a détruit une partie du savoir de la chrétienté. Je sais que j'ai agi selon la volonté du seigneur... Lisez-le, alors frère Guillaume. Je te le montrerai après, jeune homme.

Adson prend la parole. Il y a une chose que Georges doit savoir :

Adson : "VENERABLE JORGE, VOS NO PODEIS VERLO, PERO MAESTRO LLEVA GUANTES. PARA SEPARAR LOS FOLIOS TENDRIA QUE HUMEDECER LOS DEDOS EN LA LENGUA, HASTA QUE HUBIERA RECIBIDO SUFICIENTE VENENO." - Vénérable Georges, vous ne pouvez pas le voir, mais mon maître porte des gants. Pour séparer les feuilles il aurait besoin d'humidifier les doigts sur la langue, jusqu'à recevoir assez de poison.

Le secret du livre est dévoilé ! Mais Georges ne se rend pas.





**FUTURS' FREWARE DIFFUSION**