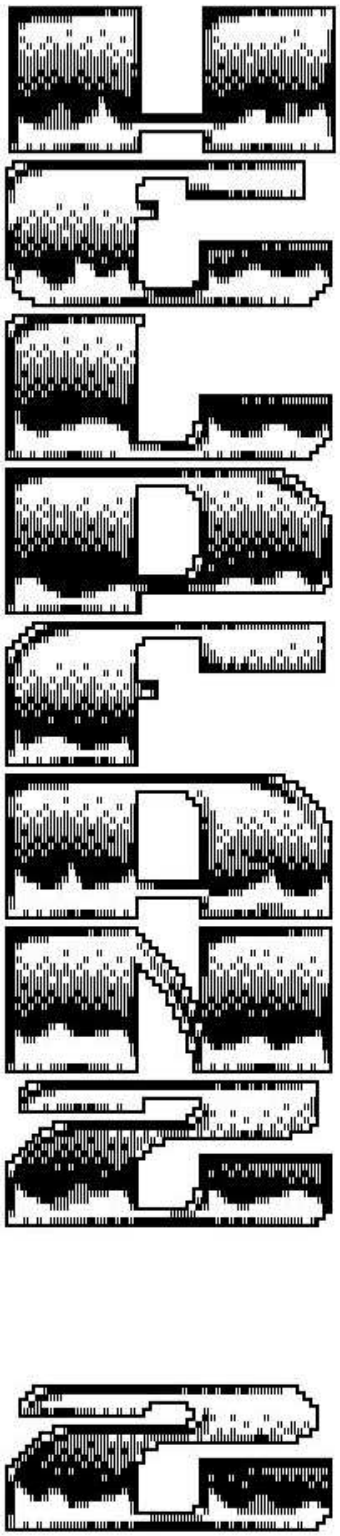
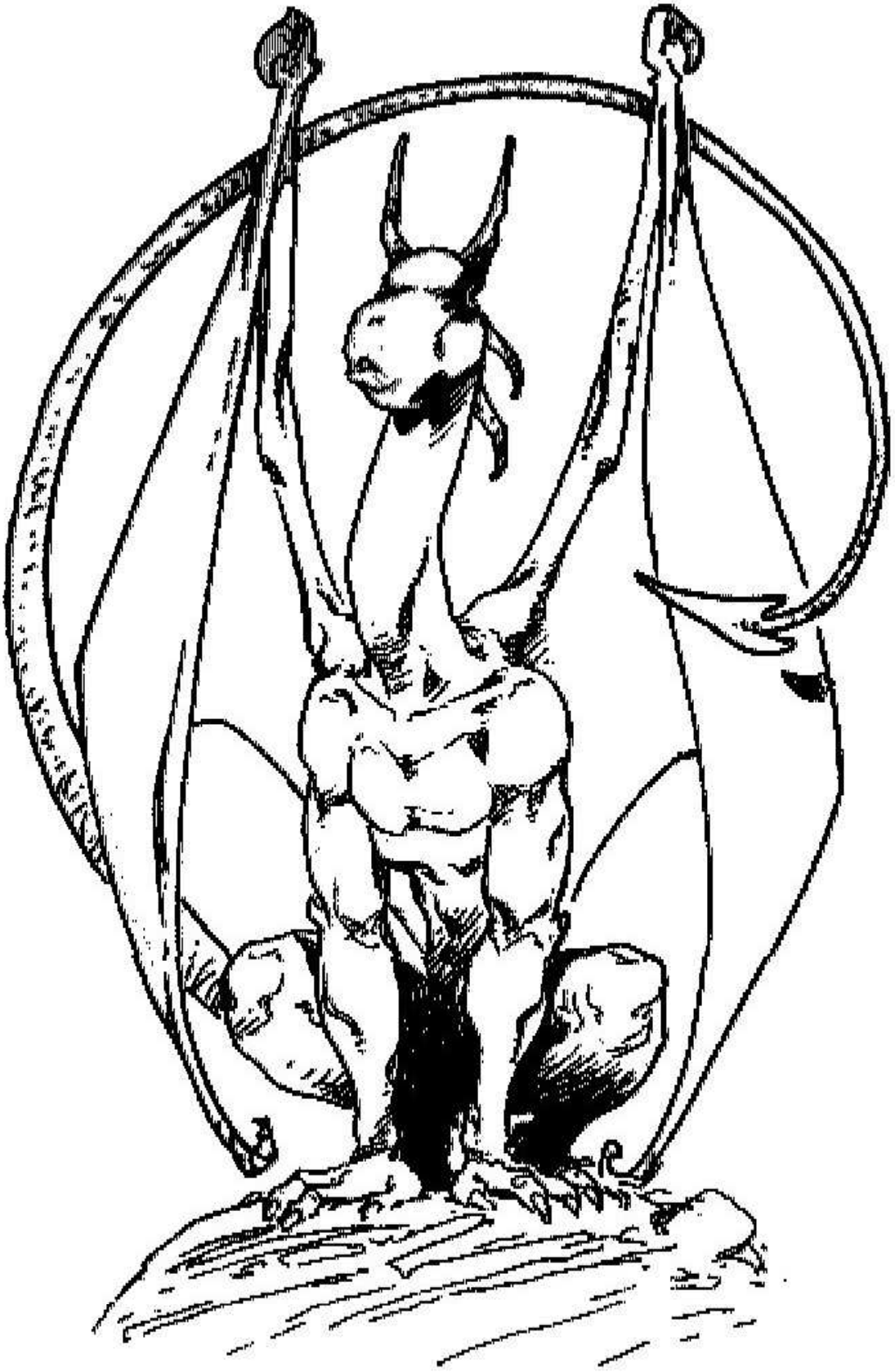
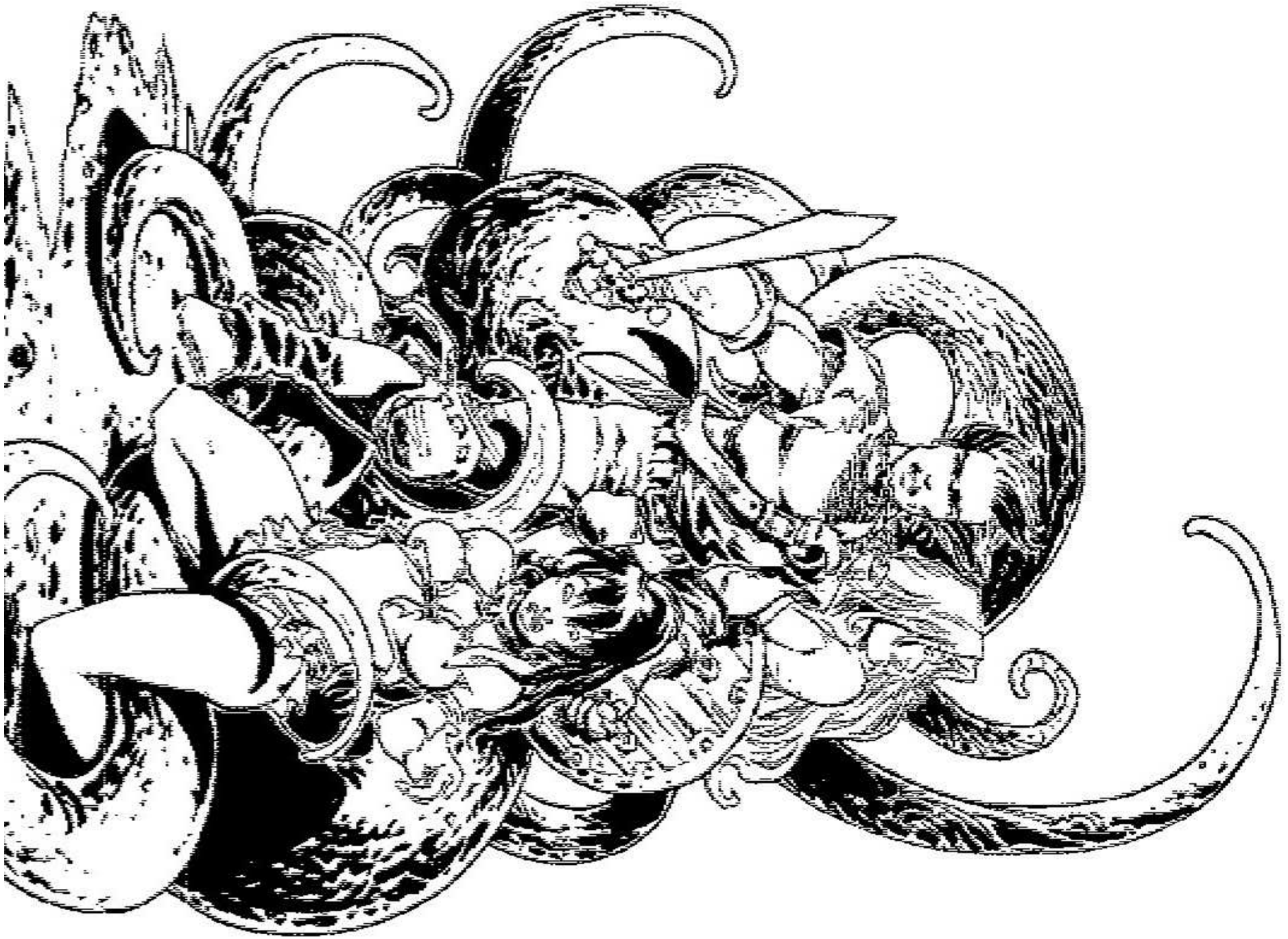






HELPFANZ 2

Eh hop, aprés plus de 2 ans, un nouveau helpfanz est fini... C'est un délai un peu long mais il faut voir la différence entre le premier numéro et celui-ci... La différence c'est la qualité !!! Et quelle qualité !!! Non, sans déconner, cette fois-ci, c'est d'un vrai fanzine dont il s'agit.

Avant de vous laisser lire le contenu de ce numéro 2, je tiens à remèrcier certaines personnes ayants plus ou moins participé à ce numéro: particulièrement Romain pour son article sur l'optimisation et son interview trés intéréssante. Plissken, pour la mème chose. Tom & Jerry pour l'article sur le hacking de musique dans les jeux (6 pages c'est pas rien !!!). Et les autres, ceux qui ont testés ce fanz, ceux qui m'ont encouragé à le faire. Slyder aussi pour sa collaboration en me fournissant les logiciels nécessaires... Et j'en passe, et j'en oublis mais de toutes façon, je remercis tout le monde...

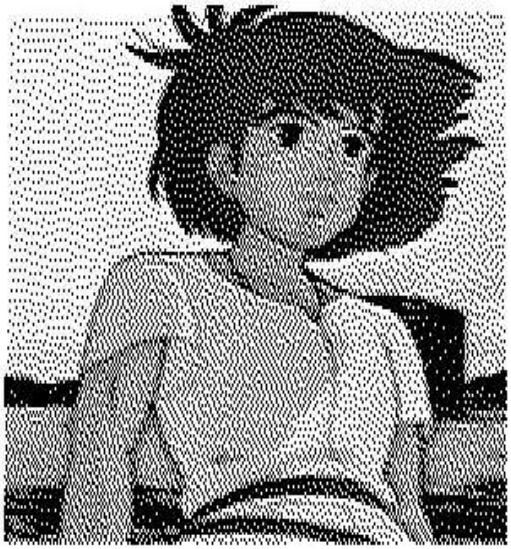
Aussi, sans vouloir vous ennuyer avec ce fanz, il serait bien que d'autres personnes participent si vous voulez un 3ème numéro !!! Car sans articles, pas de fanz ...

Vous remarquerez aussi qu'il n'y a pas de news dans ce numéro, mais je pense que des news de plus de 2 ans n'intéressent personne ... Ici, pas de rubriques bidon, mais de vrai aides (d'ou le nom du fanz !!!), que ce soit au point de vue code ou connaissances... De toute manière, n'hésitez pas à me contacter si vous voulez un article spécial la prochaine fois. Surtout les débutants (s'il en reste...).

Bon, il me reste quelques lignes pour vous communiquer mon sentiment, à savoir que l'activité est trop faible, il faut ce remettre au travail et sortir quelque chose... A ce propos, des projets à trop long termes polluent la scène, trop de choses sont en attente depuis trop longtemps, il faut boucler ses projets au plus vite !!! Pour conclure, je vous souhaite une bonne lecture de ce fanz et espère que celui-ci vous plaira.

Iron

SOMMAIRE



Voici quelques nouvelles de mes projets. Cela peut vous paraitre étrange que je monopolise une rubrique entière à parler de moi, mais à l'origine cette rubrique était prévue pour parler du groupe Hard'OS...

Pour commencer, parlons de helpfanz, que vous tenez dans vos mains. A l'origine prévu sur disc j'ai finalement changé d'avis pour le faire sur papier, le travail d'un fanz disc étant trop élevé pour moi... J'ai donc préféré le papier qui me facilite grandement la vie. Toujours est-il que si vous voulez un 3ème numéro, il serait bon de participer !!!

Je prépare aussi d'autres projets, dont la fameuse Iron mégadémo que certain d'entre vous pourront apperçevoir à l'amstrad expo ou à d'autres meetings. Pour l'instant, pas grand chose n'est fait. L'intro est bien avancée mais il me reste de la place en RAM et je compte donc rajouter d'autres objets 3D... Cette intro est composée d'un graph fullscreen de Slyder; d'un scroll border et de 4 cubes en 3D face pleine (ce n'est pas en temps réel, la technique employée est la mème que celle d'Overflow dans sa preview "All in 3D"). A l'avenir il y aura autre chose. Pour la musique je n'ai pas encore décidé mais ce serait du Shap que ca ne m'étonnerait pas !!! Aussi, deux autres parts sont commencées dont une trés avancée, à base de ligne à ligne; rupture de sprites hard; de 12 wobbler différentiels et d'ondulation écran au mode 2. Dans cette part, le son sera un sample joué avec les DMA... L'autre part est à peine commencé sinon l'effet principal (un wobbler) mais la structure de l'écran n'est elle pas terminée. Aussi, il y aura d'autres parts, certainement une dixaine, dont une au moins à base de rupture verticale, voir mème si j'y arrive, de RVI (encore faut-il que j'y arrive ce qui n'est pas le cas actuellement...). Pour conclure, il y aura un menu dont je n'ai pas encore tout pensé mais une petite idée me dit que ce sera sous forme d'un jeu ou on pourra éventuellemnt mourir !!! Bref le délire. D'autres effets tels que des images fractale seront au rendez-vous (j'ai pas dit que ce serait du temps réél !!!). La plus part des effets seront des effets hard étant donné mon mauvais niveau en soft. Ajoutons à cela trés peu de textes et de la bonne zic bien bourrin et le tour sera joué !!! Pas de date de sortie car ça sortira quand ce sera fini !!!.

Mais ce n'est pas pour cela que pendant ce temps je ne sortirais rien. Au contraire je reste actif. Déja il y a helpfanz 3 qui sortira quand j'aurais assez d'articles. Aussi j'éspère ètre présent aux meetings autant que je le peux et le souhaite !!!

Pour mon jeu: "Du bois dont on fait des pipes" le projet sera revu à la baisse par manque de mémoire... J'avais vu trop grand mais c'est tout de mème dommage d'ètre obligé de réduire !!! Aussi des bugs dans les tests de collision et changement d'écran m'obligent sans arret à revenir en arrière sur des routines vielles de plusieurs mois ce qui n'est pas toujours trés évident...

Enfin, un petit projet de jeu en réseau m'intéresse mais ne débutera uniquement que lorsque tout aura été pensé sur papier... Le concept si ce jeu voit le jour un jour, sera le mème que Klax avec la possibilité d'envoyer des pièces à la con aux autres adversaires voir meme d'autres trucs dans le genre... Mais ce n'est qu'un projet, ne voyez pas ici une promesse.

Voila, c'est à peu pret tout pour les nouvelles, je vous laisse donc parcourir ce fanz et n'hésitez pas à m'écrire pour toute critique constructive; pour m'envoyer des articles, ou tout simplement pour me poser des questions... Voici pour vous mon adresse:

Iron: Durand Amaury
20 rue de la 59ème
division anglaise
14220 Thury Harcourt

De mème, n'hésitez pas à m'envoyer des articles sur le sujet de votre choix. Je me charge de la rubrique hardware, les autres rubriques de ce fanzines sont donc libres et vous pouvez donc si le coeur vous en dit les rédiger vous mème.

Voila, cette fois c'est tout j'ai parlé de ce que je voulais, amusez vous bien et produisez d'ici le prochain numéro plein de bonnes démos que l'on pourra tester la prochaine fois. Aussi et pour finir, si vous avez quelque chose à dire ces quelques pages sont à vous... A bientot !!!

OPTIMISATION

Tout programme (aussi bien démo,jeu,logiciel) se doit d'ètre le mieux optimisé afin d'ètre le plus impréssionnant possible. Cet article traitera donc de l'optimisation (en vitesse) par le biais du sinus-scroll car je pense que c'est l'éffet old-school qui demande le plus d' optimisations possibles. Désolé mais le but de cet article n'est pas de vous apprendre a en faire un ! (Mème si la quasi totalité des routines principales sont plus ou moins expliquées)

Tout d'abord comme la plupart des éffets sont basés sur la lecture de tables, il est trés important d'optimiser au maximum les lectures (où écritures) des tables. Le mieux est de s' arranger pour que la table fasse au plus 256 octets et de la stocker à une adresse de poids faible nul. Les courbes doivent donc boucler au bout de 256 octets. En basic pour calculer une courbe toute simple, il suffit juste de taper (lorsque les angles sont en degrés) :

```
FOR x=0 TO 255:val=amplitude*sin(fréquence*x*360
/256)
```

Quel que soit l'éffet codé, il vaut mieux que les adresses écrans aient été précalculées dans une table ou incrustées dans le code plutot que d'avoir à faire un CALL BC26 à chaque changement de ligne. Pour savoir comment les stocker convenablement reportez vous a QUASAR 19.

Pour les éffets softs qui sont toujours affichés au meme endroit (comme les plasmas, les cercles par interférences ...) il n'est pas trés judicieux de stocker toutes les lignes écran. En effet, si on s'arrange pour que la première ligne de l'effet soit situées sur la première ligne d'une ligne de charactère écran,il suffit juste de stocker la première ligne de chaque charactère et d'ajouter a chaque fois 8 au poids fort de l'adresse pour passer à la ligne suivante. Comme je n'ai pas l'impréssion d'avoir été trés clair, un exemple s'impose (avant le calcul de la ligne suivante HL vaut l'adresse du premier octet de la ligne actuelle, donc il faut utiliser cette méthode éssentiellement dans les cas ou on affiche des sprites d'un octet de large...)

```
        ld B,8
ligne1  ld HL,#c000 ;ligne8  'HL=#f800
          ...
        ld A,L      ;ligne9  ld hl,#c050
        ADD A,B     ;(écran standard)
        LD L,A
ligne2  'HL=#C800
          ...
```

Quand au sinus-scroll, c'est encore une autre méthode de stockage. Il faut créer une table qui comporte autant de valeurs qu'il y a d'octets dans la zone ou le scroll sera affiche. La table fait donc (en words) :

largeur du scroll*nombre de lignes de la zone

Toujours pour des raisons d'optimisation, la table doit ètre stockée à une adresse de poids faible nul. Il est aussi plus judicieux de s' arranger que la hauteur totale de la zone fasse 64 lignes, ainsi chaque colonne dans la table fera 128 octets (2^7).

Un moyen d'améliorer grandement la vitesse d' affichage des sprites non masqués est d'utiliser la pile. Si SP pointe sur le sprite c'est tout con, il suffit juste de faire (HL pointe sur l' écran) :

```
POP DE
LD  (HL),E
INC HL
LD  (HL),D
INC HL
```

Autant de fois que la moitié de la largeur du sprite. Par contre si SP pointe sur l'écran, il ne faut pas oublier qu'un PUSH décrémente SP, donc le sprite sera affiché de la droite vers la gauche. Il est donc plus pratique aussi de stocker le sprite a l'envers.Si HL pointe sur le sprite :

```
LD   E,(HL)
INC  HL  (ou inc l si possible)
LD   D,(HL)
INC  L
PUSH DE
```

...
Si les sprites sont dans les banks ce n'est pas grave car la pile peut lire dans les banks.

Article:
Romain

Pour le sinus-scroll, la pile sert à lire les adresses écran. SP pointe dans la table l'adresse ou sera affiché l'octet de la première ligne du scroll. Si DE pointe sur le scroll, il suffit simplement de faire :

```
POP  HL
LD   A,(DE)
ld   (hl),A
```

Pour afficher cet octet. Mon but n'étant pas de vous donner un algo complet de sinus-scroll c'est à vous de vous débrouiller pour afficher le reste et stocker le scroll en mémoire.

Sinon pourquoi avoir calculé les adresses comme si l'effet était affiché dans une zone de 64 lignes ? Et bien, c'est tout simplement parce que c'est beaucoup plus pratique comme ca... Au début de la routine, H vaut le poids fort de l'adresse de la table. On lit une valeur dans la courbe que l'on met ensuite dans L. Après un LD SP,HL c'est la pile qui pointe dans la table la bonne adresse. Il suffit donc ensuite d'éxécuter la routine d'affichage pour afficher une bande verticale. Si on répète 96 (la largeur de l'écran) fois cette routine la totalité du scroll n'aura été affichée que dans la première bande écran. Il faut donc après chaque affichage d'une bande s'arranger pour que SP pointe dans la colonne suivante de la table d'adresses. Comme chaque colonne fait 64*2=128 octets, il suffit donc, en théorie, d'ajouter (aprés l'affichage de chaque bande) 128 à un registre, puis ajouter ce registre à HL. Cette méthode est donc à proscrire tellement elle est couteuse en temps machine. Heureusement pour nous, 128=%10000000 et 2*128=256. Pour changer de colonne la première fois, il suffit seulement de faire SET 7,L (ce qui revient à ajouter 128) et la deuxième fois INC H (ce qui revient a ajouter 256). Pour changer de colonne, il suffit donc d'alterner les SET 7,L avec les INC H.

Un autre moyen d'optimiser au maximum est d'éviter le plus souvent les boucles car les tests bouffent toujours pas mal de temps machine. Il vaut mieux recopier plusieurs fois la routine (lors de l'initialisation du prog, pas dans le source a la main !!) meme si le code final fera plusieurs Kilo-Octets au lieu de un ou deux.

Il faut aussi utiliser les registres secondaires, il sont là pour ça. Grace à eux le nombre de registres est double et l'instruction EXX ne prends qu'un seul NOP. Aussi si ces registres sont utilisées dans une routine particulière, les utiliser nous permet d'éviter de sauvegarder HL,BC et DE avec des POP et de les récupérer avec des PUSH qui prennent beaucoup de temps machine.

Voila, je pense avoir dit l'essentiel. Meme si vous n'avez pas appris grand chose je pense qu'il était bien de rappeler tout ca pour pouvoir faire des trucs rapides.De toute facon, pour qu'une démo old-school soit bien, il faut que les routines soient hyper rapides afin de faire l'effet le plus impressionant, alors que pour le new-school comme, mis a part l'école buissonière, il n'y a encore rien eu de sortit, mème les routines les plus lentes seront impressionantes puisque inedites... Que ceux qui ont l'intention de faire de nouvelles démos sur CPC essayent de coder des effets new-school les plus rapides possibles afin de renouveler un peu la bibliotheque cpcenne que je trouve pas mal répétitive d'une démo à l'autre... Je préfère nettement le style des démos C64 ou celui des dernières démos sur ATARI. Bon, c'est tout pour cet article.

LA RUPTURE

La voici, la voila, la rubrique hardware... Aprés la rubrique soft de ce cher Romain, c'est moi votre pote Iron qui me charge de me taper cette rubrique qui en est donc a son début.

Mais ce n'est pas parce que c'est la première rubrique hardware de ce fanz (normal puisque c'est le premier vrai numéro) que je vais faire un article vide de sens... Au contraire, on va attaquer avec un vrai thème, à savoir la rupture simple. Je sais, certains vont dire que d'autres fanzs en ont parlé avant moi, tel que Quasar CPC dans le numéro 7 par exemple... Alors là, je dis oui mais pour moi, notre Philippe national n'est resté que trop théorique dans ses articles. Moi j'aime pas... Autant le dire tout de suite, celui qui a apris à faire de la rupture dans Quasar ne connais en rien le fonctionnement du CRTC. Alors que, grace à Helpfanz, vous allez enfin tout comprendre... J'éssayerais d'être le plus clair possible dans mes explications bien que cela ne soit pas toujours évidant...

I) Le CRTC, un peu de théorie:

Pour commencer mon explication, je vais tout d'abord vous donner une règle à suivre pour faire de la rupture, à savoir:

(La somme de tout les R4+1)=39

Là, le petit nain roux à rire caractéristique (abrégé: ELIOT pour plus de facilité) me dit que finallement c'est simple... Surtout quand on sait que l'offset est pris en compte à chaque bouclage de C4... Oups, d'un seul coup, plus personne ne comprend... Bon, plus concrètement, considérons les registres (notés: R) et les compteurs internes (notés: C)...

Oui mais c'est quoi les compteurs ? Eh bien, quand on mets une valeur dans un registre, par exemple R4, cette valeur est en fait la valeur de référence de ce registre. Cette valeur constante ne change pas temps qu'on ne le lui dit pas... Par contre dans ce registre (c'est une image...) il y a un compteur qui lui s'incrémente selon certaines régles. Par exemple, C4 s'incrémente à chaque bouclage de C9. On commence à étre un peu concret.

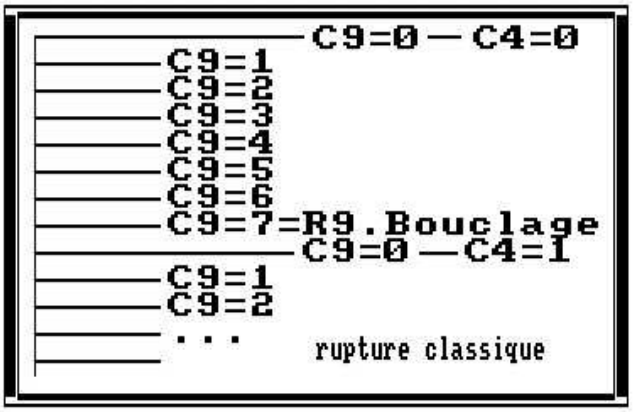
Revenons à la rupture: Le seul intéret de celle-ci est de pouvoir changer d'offset plusieurs fois par écrans. Je disais donc, que l'offset est pris en compte à chaque bouclage de C4. Là, Eliot me répond qu'il suffit donc de mettre par exemple R4=19 et le tour est joué, on a une rupture... Il essaye en Basic et observe son écran qui scroll... Bein oui, c'est normal car déja d'une part, la règle que je vous ai donné plus haut n'est pas réspéctée. Eh oui,

(19+1)+(19+1)=40 et non pas 39... D'autre part, nous n'avons plus de signal VBL qui sert à l'écran pour le synchroniser...

Oui mais comment on fait pour avoir une VBL ? Simple, Pour qu'il y ait une VBL, il faut que C4=R7 (et non pas R4=R7...). Essayons en basic de faire une rupture synchronisée, meme si nous ne sommes pas exactement à 50 Hz. Mettons donc R7=19 et R4=19 aussi. Ca marche, nous avons bien une rupture... Oui mais si vous mettez le border en couleur, vous vérrez une belle VBL au milieu de l'écran. C'est chiant mais il y a une solution !!! Cette solution c'est de changer R7 en cour de balayage... Où de le laisser stable mais s'arranger que pour le premier écran C4 ne soit JAMAIS=R7

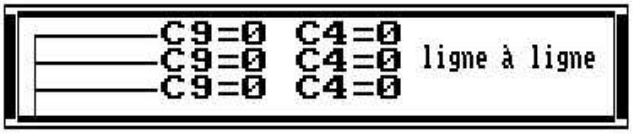
Imaginons: Pour le premier écran, R4=18 et pour le second, R4=19, soit: (18+1)+(19+1)=39 c'est bon, normalement si on ne fait pas de conneries, l'écran sera à 50Hz. Maintenant réglons le problème de la VBL. Méttons dés le début R7=19... Eh oui, pour le premier écran, C4 boucle à 18, donc n'atteind JAMAIS R7, donc pas de VBL. Mais pour le second écran, C4 boucle à 19, ce qui est la valeur de R7... Seul problème, c'est trop tot... Pourquoi ? parce que justement C4 s'incrémente au début d'un bloc de 8 lignes (quand C9 boucle), donc, il manquera 7 lignes à l'écran... Réfléchissez un peu, vous comprendrez facilement !!! Donc, pour avoir le bon nombre de lignes (312 par écran), il faut que la VBL ait lieu aprés que toutes les lignes aient été balayées... Donc, quand C4 du second écran boucle à 0. Donc, concrètement, il faut mettre R7=0 pendant le second écran CRTC, comme ca, à la 313ème ligne... VBL !!!

Donc, pour remettre un peu d'ordre dans tout ca, au lieu de croire que R7 sert à la position Y de l'écran, vous savez maintenant que c'est en fait la position de la VBL... Et comme l'écran se synchronise sur cette VBL, cela influe effectivement sur la position Y de celui-ci !!!



Voyons maintenant le registre R9 dont je parlais tout à l'heure...

Rapidement et concrètement, sachez que ce registre sert au nombre de lignes dans un bloc de lignes. Ainsi, en basic, R9=7 et un bloc de ligne fait donc 8 lignes (ne pas oublier le 0!). par exemple pour faire de la ligne à ligne, on met R9=0, ainsi, 1 bloc de ligne est créé toutes les lignes et C9 boucle donc à toutes les lignes (pixel). Comme on peut changer l'offset à chaque bouclage de C4 et que C4 est incrementé à chaque bouclage de C9, nous pouvons si nous mettons R4 à 0 changer l'offset à chaques lignes. Voyez le schéma numéro 1 pour mieux comprendre...
Mais attention a l'overflow car la mise à 0 de R4 doit ce faire quand C4=0 sinon, le CRTC ne s'y retrouve plus ce qui est plutot logique quand on y réfléchis...



Passons maintenant à l'application car c'est le but... Donc, faisons une rupture... Tout d'abord on ce synchronise sur la VBL qui est signalée par la mise à 1 du bit 0 du port B du PPI (#f5). Un simple RRA (1 NOP) mettra ce bit dans la carry... Avec la petite routine suivante on teste donc le signal VBL:

```
        LD      B,#f5           ;PPI port B
frame   IN      A,(C)           ;on mets dans A
                                ;l'octet du port
                                ;B
        RRA                     ;rotation gauche
        JR      NC,FRAME        ;si pas carry on
                                ;retest...
```

Ensuite, le mieux est de ce synchroniser sur un HALT histoire d'être stable...

Petite remarque, sur CPC+, au lieu de ce synchroniser sur la VBL, il suffirait de provoquer une interruption raster à la ligne 1 et un HALT suffirait à ce rendre en début de balayage...

Donc, sur old, on ce synchronise sur un HALT, cependant, certains CRTC capricieux (comme par exemple les CRTC 4 et 3) attraperont le premier HALT, celui envoyé en meme temps que le signal VBL, ce qui nous ferait une différence de 52 lignes pixel entre certains CRTC... Il faut donc éviter ce premier HALT dans la plus part des cas ce qui n'est pas bien compliqué, un simple :

```
        LD      B,32:DJNZ $
```

suffira. Esuite on balance notre HALT et cette fois c'est certain, nous sommes au deuxième HALT de l'écran (oui je sais je ne devrais pas dire de l'écran mais on s'en fout...). Ensuite, on décide de la taille de nos écrans... Disons 18 blocs de lignes pour le premier et 19 pour le deuxième (19+20=39 c'est toujours ok !). Occupons nous du CRTC maintenant. On commence avec R7 que nous allons mettre par exemple à 23 et pas à 255 ce qui est plutot naze non ? Ensuite c'est R4 que nous méttons donc à 18. Reste à attendre un peu histoire que C4 boucle à 0 avant de mettre la valeur de notre 2ème écran à 19... Au deuxième halt, à peu près 6,5 lignes on peut meme dire que c'est presque exactement tout les 6,5 lignes selon ce que le system va faire pendant une interruption... Bref, il nous faut 19 lignes déja parcourues... Donc, 19-6,5=12,5 lignes à parcourir encore. Donc, 6,5+6,5=13 ce qui est bon. Nous attendrons donc encore 2 HALT. HALT:HALT... Voila, c'est bon, le premier écran à été balayé... Nous pouvons donc remettre de bonnes valeurs... Nous mettons donc R7=0 pour que le signal VBL soit balancé à la fin de cet écran et R4=19 pour la taille de l'écran... Un petit test de touche et le tour et joué, voila, nous avons fait notre première rupture...

Maintenant, rajoutons la règle finale à tout ça: à savoir que la règle que je vous ai donné plus haut n'est pas tout à fait éxacte. La règle finale pour avoir un écran à 50Hz est donc:

(E signifira somme)

(E(R4+1))*(R9+1)+E(R5)=312 lignes.

Aussi nous vérrons la prochaine fois en exclusivité la rupture ligne à ligne et la rupture verticale, si il y a un 3ème numéro bien entendu... Cela ne dépend que de vous.

Récupérer une musique dans un jeu CPC (c) Tom et Jerry GPA 1993,2001 !

- Publié dans "Le petit Electro Jack illustre", numéro 9, 1993
- Retapé et enrichi le 30/06/01 pour Iron !
- Dédié a Trebor 45, alias Robert Mathieu.

A la demande de Trébor 45, je vais vous faire un petit topo sur l'art et la manière de récupérer une musique dans un jeu commercial ou une démo. Ayant pas mal pratiqué cet exercice (cf les compilations Music Pack !), je peux vous faire part de mon expérience dans le domaine (cela fait un peu ancien combattant !).
Soyez indulgent envers mon style d'écriture, il n'est pas simple de faire un article sur ce sujet sans répéter 30 fois les mots musique, player, etc...

Avant de parler de la marche à suivre, il faut préciser deux choses :

- une interface style Hacker est chaudement recommandée (jadis disponible pour cpc et cpc plus chez Duchet Computers). A défaut, procurez-vous une Ramcard et récupérez la rom du Hacker ! Les bases évoquées ici peuvent être utilisées avec un moniteur comme le Super Monitor Razormaid d'Antoine, mais cela compliquera beaucoup les choses...

- une connaissance minimale de l'assembleur Z80 est nécessaire, toutes les musiques de qualité (sauf celles de Michel Windogradoff !) étant programmées avec ce langage. Si vous ne pratiquez pas, il est toujours temps de vous y mettre !

1) Un peu de théorie...

Une musique en assembleur se compose de deux parties bien distinctes :

* Le player : en francais, la routine qui "joue" la musique. Il comporte quasiment systematiquement trois sub-routines, celle initialisant la musique (Init), celle la jouant (Exec), et celle l'arretant (Stop).

* Les datas, qui comprennent non seulement les notes, mais aussi les redéfinitions des sons (en gros, les commandes ENV et ENT en Basic).

Ces données peuvent éventuellement étre compactées (ex : AY player de Madram).

Les musiques dans les jeux sont en général jouées sous intérruption. Cela signifie que le player est appelé à intervalle régulier par l'intermédiaire d'une fonctionnalité spéciale du microprocésseur Z80, le coeur de nos CPC.
Deux cas de figure peuvent se presenter :

* Le programmeur utilise une intérruption programmée (en passant par le vecteur &BCD7). C'est en pratique trés rare.

* Le programmeur utilise l'intérruption système RST &38 en la détournant ou en se l'appropriant totalement (la quasi-majorite des cas pour les jeux, et quelques rares démos).

Parfois, les musiques ne sont pas jouées par l'intermédiaire d'une intérruption (beaucoup de démos dont les Music Pack sont concernées). L'intérruption système RST &38 est alors "bloquée" et ne sert qu'à faire de la synchronisation vidéo pour utiliser des techniques nécéssitant un timing précis, comme de la rupture ou des rasters.

Le code d'une musique peut se trouver n'importe ou en mémoire. C'est là un des avantages de l'assembleur, mais dans ce cas précis, cela nous complique la vie. Les données peuvent aussi bien se loger dans la RAM Basic que dans la RAM vidéo, les zones réservées au système ou dans les banks mémoire supplémentaires pour les cpc ayant plus de 64ko. Cela peut donc poser problème pour les trouver. Heureusement que le CPC n'a pas 4mo de mémoire en standard !

En général, le player et les datas d'une musique sont rangés dans la mémoire de facon contigue, mais certains programmeurs sadiques trouvent une déléctation suprème à disperser le tout. D'autres poussent mème le vice jusqu'à morceler les données un peu partout ! Ne les maudissons pas et supposons qu'ils sont contraints d'agir

ainsi pour économiser de la place mémoire. M' enfin, cela n'arrange pas nos petites affaires !

Vous l'avez compris, récuperer une musique suppose donc de rechercher son player et ses données !

2) La quête du player

C'est la partie la plus délicate de notre périple. Il nous faut localiser la routine jouant la musique. Les différentes techniques présentées permettent dans la majorité des cas de la trouver.

2.1) Rechercher les opérations effectuees sur le RST &38.

Pour pouvoir utiliser à ses fins ce vecteur d' intérruption, le programmeur doit modifier son adresse de saut (JP &B941 sur un CPC 6128). Il doit donc changer le contenu des cases mémoire &39 et &3A par une ou des afféctations mémoire. En recherchant les chaines héxadecimales &39,&00 ou &38,&00, nous avons donc une chance de trouver des bouts de code dans ce style :

```
LD HL,&xxxx     LD HL,&39       LD A,&C3
LD (&39),HL     LD (HL),&xx     LD (&38),A
                INC HL          LD HL,&xxxx
                LD (HL),&xx     LD (&39),HL
etc...
```

Evidemment, il y a toujours des originaux pour faire autrement, en utilisant par exemple la commande LDIR ou les registres d'index.

Si vous trouvez quelque chose ressemblant a ce que nous venons de voir, notez l'adresse pokée en &39 et allez voir ce qu'il y a. La plupart du temps, vous allez trouver une routine qui sauvegarde tous les registres du Z80 avec des commandes PUSH, puis lance quelques CALL aprés de petits tests (JR Z, etc..). Exécutez en mode direct ces CALL aprés les avoir éventuellement étudiés. Si un bruit continu sort des entrailles de votre CPC, c'est presque gagné ! Notez l' adresse du CALL et testez cette hypothétique musique en tapant une de ces routines :

a) La musique est jouable directement

```
boucle  CALL &BD19
        CALL music ' (l'adresse que vous
                avez notée, on suit ?)
        JP boucle
```

b) La musique n'est pas jouable avec les interruptions ou est logée à un endroit réservé au système :

```
        DI
        LD HL,&38
        LD (HL),&C9
boucle  LD B,&F5
boucle2 IN A,(C)
        RRA
        JR NZ,boucle2
        CALL music
        JP boucle
```

Si vous entendez une musique, le player est trouvé !

2.2) Rechercher des bouts de routine sauvegardant les registres du Z80.

Si la méthode RST &38 s'est révélée vaine, tout n'est pas perdu. Je ne vais pas expliquer dans les détails la notion d'interruption, mais une contrainte qu'elle pose au programmeur va nous intéresser. Ce dernier est obligé de sauvegarder certains registres du Z80 avant de faire quoi que ce soit. Comme en prime, le codeur est un animal bète et discipliné qui reproduit ce qu'il a appris, il va sauvegarder ces registres dans un ordre donné :

```
PUSH AF         &F5
PUSH BC         &C5
PUSH HL         &E5
PUSH DE         &D5
```

On recherche les valeurs correspondant a ces mnémoniques et le tour est joué ! Puis l'on recommence ce que nous avons précédemment fait (analyse et tests des CALL). Attention, une sauvegarde des registres ne correspond pas forcément à une routine jouant ensuite une musique...

2.3) Le Hacker à la rescousse !

Vous avez lamentablement échoué dans vos précédentes recherches. Il existe une solution pour vérifier définitivement que la musique n'utilise pas l'intérruption RST &38, le mode ALTERNATE du Hacker. Cette commande permet de faire tourner dans les bank supplémentaires du cpc (version 128ko obligatoire donc !) un programme. Contrairement à ce que prétend la documentation de cette interface, cette commande tourne aussi sur cpc 6128. Enorme avantage, une fois le programme stoppé par le bouton magique, les banks ne sont pas initialisées. Il suffit de connecter ensuite la bank &C4 (bank 1) et de désassembler en &4038 pour trouver l'adresse de saut de l'interruption &38 du programme arrété, sympathique non ? A partir de là, si cette adresse semble intéréssante, on recommence la procédure décrite en 2.1).

Si vous ètes l'heureux utilisateur d'une Multiface II, vous n'avez pas besoin de cette astuce, un arret du programme et l'édition de la ram en &38 suffit.

Cette méthode n'est évidemment utile que si le jeu contenant la musique convoitée n'occupe pas les bank supplémentaires...

2.4) Rechercher la routine envoyant des données au PSG.

Dans toute musique, le programmeur est obligé d'utiliser une routine qui envoie des données au procésseur sonore du CPC. Il existe bien dans le firmware du cpc un vecteur système servant à cela (&BD34), mais il n'est jamais utilisé. Les programmeurs de musiques préfèrent utiliser leur propre routine. Cette dernière est grosso modo toujours la mème et se déroule comme suit (exemple : routine d'une musique Soundtrakker).

```
LD B,&F4
OUT (C),A
LD B,&F6
IN A,(C)
OR &C0
OUT (C),A
AND &3F
OUT (C),A
LD B,&F4
OUT (C),C
LD B,&F6
IN A,(C)
AND &80
OUT (C),A
OUT (C),C
```

Le but n'est pas ici d'expliquer le fonctionnement de la routine. Il suffit de savoir qu'un bout de code dans ce style sert soit à envoyer des données au PSG, soit à lire l'état du clavier, ou lire des données provenant du lecteur de cassette (je vois déja les puristes s'insurger !). En cherchant des chaines héxadécimales 06 F4 (LD B,&F4) et 06 F6 (LD B,&F6), on a donc des chances de trouver ce bout de programme. A partir de là, il faut "remonter" dans le code, trouver quelles routines l'utilisent, pour finalement retrouver le player ! C'est certe compliqué, mais cela fonctionne souvent.

2.5) Le désassemblage pur et dur !

Le programmeur de jeu dont vous convoitez la musique est un ètre raffiné ; dernière solution, la plus laborieuse, il vous faut étudier le programme depuis son adresse d'éxécution. Pas de recète miracle, seule de la persévérance et un peu de chance vous permettront d'arriver à vos fins...

Bien, à partir de maintenant, je considère vous vous avez trouvé le player (pas de panique si ce n'est pas encore bien clair, nous vérrons deux exemples par la suite). Avant de passer au néttoyage de la musique (c'est à dire ne garder que les données utiles), il nous faut trouver les adresses d'init du ou des thèmes qui la composent, et pour les perféctionnistes, celle qui l'arrète. En pratique, on peut se passer fort bien de cette dernière sous Basic, un CALL &BCA7 faisant l'affaire (pitié, pas de PRINT CHR$(7), ce que l'on voit souvent dans des vieilles mauvaises démos allemandes).

Là, pas de recette miracle, il faut noter que souvent, cette adresse n'est pas bien loin de la routine jouant la musique. On retrouve alors un appel à cette routine "prés" de la portion de code détournant le RST &38.
A vos désassembleurs !

Pour les musiques ayant plusieurs thèmes, en général, les programmeurs utilisent un registre (souvent A) pour définir quel air doit ètre

joué. Parfois, il faut poker des valeurs en mémoire (les musiques de Tiny Williams, par exemple). Bref, il faut parfois avoir du flair !

3) Le grand ménage

Il faut maintenant récupérer les données associées au player que vous avez débusqué ! La encore, pas de méthode miracle, mais quelques recommandations à suivre donnant souvent de bons résultats.

La première chose consiste à parcourir la mémoire à partir du player pour trouver une zone de 00 conséquentes ou une routine n'ayant rien à voir avec la musique. Cette technique primaire est assez éfficace car souvent, le player et les données sont situées l'un aprés l'autre, particulièrement si votre musique se trouve dans une bank de mémoire supplémentaire, ou dans la zone réservée au système (&A67B a &BFFF pour le 6128).

Sinon, il faut procéder par tatonnement, en sauvegardant tout d'abord une large zone mémoire, puis en la "dégraissant" (remplir des bouts avec des 00) au fur et à mesure. Je vous le concède, ce n'est pas bien passionnant ni trés épanouissant techniquement, en plus, c'est souvent long ! La longueur usuelle pour une musique simple varie entre &A00 et &1000, mais peut ètre plus grande chez certains éditeurs (Krisalis, par exemple !).

Petite anecdote, la musique la plus lassante (pour ètre poli) que j'ai néttoyée est celle de Turbo out run. Elle ne comporte pas moins de 8 blocs de données différents, situés un peu partout dans la mémoire. De quoi devenir dingue!

Voila, aprés ces explications, "ripper" une musique de jeu devrait vous sembler facile. Euh.. non ? Bon, deux exemples vous montreront que c'est plus simple que cela en a l'air.

4) Ripper la musique de 'Outrun'avec le Hacker

Nous allons récupérer ensemble la musique d'un jeu trés célèbre, Outrun. C'est vrai, la conversion sur nos micros de cette borne d'arcade est vraiment mauvaise. La musique est néanmoins assez mignonne sur cpc, en plus, elle est facile a récupérer !

a) La recherche du player.

Aprés avoir chargé le jeu, on l'arrète avec le hacker au niveau de la page de présentation. On commence la recherche avec la première méthode (cf 2.1)).

* Taper 'S' puis la chaine hexadecimale 38 00
=> Le Hacker trouve trois fois cette chaine aux adresses &107, &4CA4, &83BE.

* Désassembler un peu avant ces adresses (&100, &4CA0,&83B0 par exemple) grace à la commande 'D' et étudier ces zone de code.

Oh miracle, la première adresse (&107) a bien une action sur le RST &38 en pokant l'adresse &17C en &39.

* Désassembler en &17C.
=>Nous trouvons d'abord deux instructions qui sauvegardent des registres (ca ne vous rappelle rien ?), puis deux appels à deux routines :

```
CALL C,&8000
CALL &99B
```

* On execute le premier CALL en mode direct.
=>Le CALL &8000 se traduit par un son strident et continu. Le player est trouvé ?
Vérifions...

* Taper la routine suivante en &BE80 puis faire un CALL &BE80

```
boucle  CALL &BD19
        CALL &8000
        JP boucle
```

Normalement, vous devriez entendre la musique d'Outrun ! Le plus dur est fait.

b) Recherche de la sub-routine d'initialisation

On revient a l'endroit ou nous avions trouvé une instruction travaillant sur le RST &38 (en

&107).
* Désassembler à nouveau à partir de &100 et chercher un CALL éffectué pas trop loin de la case mémoire &8000 (adresse d'éxécution précédemment trouvée).
On trouve le code suivant :

```
LD A,&1
CALL &80B9
```

Cet appel se fait a priori dans la zone mémoire ou se trouve la musique... Testons cette routine à l'aide d'un CALL (ne pas oublier de mettre le registre A a la valeur 1). Evidemment, rien ne se passe. Il faut éxécuter la routine jouant la musique pour en vérifier le résultat. Décidement nous avons beaucoup de chance, nous avons trouvé du premier coup la bonne adresse, la musique est jouée à partir de son deéut. Comme nous savons qu'il y a deux thèmes sur Outrun, cherchons un éventuel autre appel à la routine d'initialisation (CALL &80B9 pour ceux qui ne suivent pas !) Aie, c'est le desert total, on n'a rien trouvé de plus...

A priori, vu ce que nous avons trouvé précédemment, il semblerait que c'est le registre A qui détermine la musique à initialiser. Il ne nous reste qu'à faire des éssais avec différentes valeurs pour ce registre, en partant de 0. La chance (ou le talent ?) nous sourie encore, car un CALL &80B9 avec A=0 initialise l'autre thème du jeu. Les plus curieux pourront lancer la routine avec A=3 ou A=4, par exemple, vous entendrez alors des bruitages !

c) Le nettoyage

* L'éxécution de la musique étant en &8000, nous allons éditer la mémoire à partir de cette adresse et "remonter" (commande 'e' puis flèche vers le haut).
=>Nous voyons une petite zone de &00 avant l'adresse d'éxécution. Cela permet de supposer que l'adresse d'éxécution correspond au début de la musique.

* Cherchons maintenant la "fin" du morceau, en déscendant dans la mémoire (flèche vers le bas). En &8F80, nous retrouvons une autre zone de &00.
=>Nous allons travailler à partir de ces deux bornes.

* Sauvegarder la zone dans un fichier (OUTRUN.MUS, par exemple).
* Retourner sous Basic pour éffacer la mémoire
* Revenir sous le prompt du Hacker, puis charger en mémoire le fichier OUTRUN.MUS.
* Tester la musique en l'écoutant jusqu'à la fin. Dans notre cas, tout se passe bien.

* Nous pouvons donc recommencer l'opération en éffaçant une petite portion de la zone RAM occupée par le fichier (par exemple, de &8E00 à &8F80).

=>Dans cet éxemple précis, tout rognage du fichier OUTRUN.MUS se traduit par des pertes de données (sons bizarres, voix ne jouant plus, ou avec des rythmes décalés, etc...). La longueur définitive de notre fichier est donc:

```
&8F7F - &8000 = &F80
```

Victoire, la musique d'Outrun est mise en fichier et utilisable dans vos créations les plus folles ! Il faut quand meme noter que cette extraction est facile. Pour quelqu'un de bien entrainé, la chose est réglée en moins d'un quart d'heure. Cela constitue néanmoins un éxcéllent éxercice pour les profanes...

5) Ripper la musique de "Dizzy down the rapids" avec le Hacker

Et hop, une commande spéciale d'Iron. Nous allons voir maintenant un deuxième exemple. La victime est cette fois-ci un jeu de Codemasters dans le style de Toobin, avec l'incontournable Dizzy ! La description sera un peu plus lapidaire que pour Outrun, mais devrait suffire pour y arriver.

a) La recherche du player

* Arreter le jeu sur la page du menu avec le Hacker puis rechercher la chaine hexadecimale 38 00 en mémoire :
=>On trouve 3 adresses : &6833, &7297, &AA76

* Désassembler un peu avant ces adresses (&6820, &7290, &AA70)

=>Rien ne semble correspondre a du code.

* Rechercher la chaine héxadécimale 39 00 en mémoire :
=>On trouve une seule adresse : &2F43

* On désassemble en &2F30.
=>On trouve du code intéressant :

```
&2F3F   LD HL,&2F0E
&2F42   LD (&39),HL
```

* Désassembler en &2F0E.
=>Il y a une routine sauvegardant des registres, puis lançant avec des tests trois CALL : &2C31, &23E6 et &6F17.

* On éxécute les trois adresses :
=>Rien ne se passe pour les deux premières. Avec &6F17, un souffle se fait entendre...

* Tester l'adresse avec la routine suivante en &BE80 :

```
boucle  CALL &BD19
        CALL &6F17
        JP boucle
```

Oh, une belle musique retentit !

b) Recherche de la sub-routine d'initialisation

* On reprend le désassemblage en &2F30 en cherchant un CALL dans la zone des &6F17.
=>Rien.

* On étudie les CALL situés aprés &24F2 (la commande pokant le RST &38) en désassamblant à chaque adresse, toujours pour rechercher un éventuel appel dans la zone prés de &6F17.

  En &2F6B, on trouve un CALL &3363, qui éxécute le code suivant :

```
CALL &6E3D
RET
```

* On essaie cette adresse. Il ne se passe rien. Par précaution, on désassemble en &6E3D. La routine réssemble pourtant à une routine d' initialisation. Elle est cependent un peu courte, et est suivie d'une autre routine, en &6E44.

* On recherche des appels a cette routine(&6E44)
=>deux adresses sont trouvées : &335C et &3361. Tiens, c'est trés proche de ce que nous avons trouvé plus haut...

* En désassemblant (on n'arrète pas ici !) un peu avant &335C, on trouve ceci :

```
&335A   XOR A
&335B   JP &6E44
&335E   LD A,&1
&3360   JP &6E44
```

=>Hum, ca ressemble à une routine d' initialisation, ca... On teste !
Trouvé ! En prime, on a aussi la routine d'arret qui est &6E3D ! En général, on trouve plutot d' abord la routine d'initialisation.

c) Le néttoyage de printemps.

* On étudie le code un peu avant &6E30. A priori, le début de la musique est bien en &6E3C, car le "visuel" de la mémoire avec Edit (permet de voir une plage de la ram sous forme de valeurs héxadécimale) réssemble à un sprite (valeurs se reproduisant à des intervalles constants).

* On se déplace maintenant vers le haut de la mémoire. A partir de &7C3E, il y a des zéros partout !
=>On suppose alors que la musique est située entre &6E3D et &7C3D. Il n'est en fait mème pas nécéssaire de faire des tests, car nos précédents désassemblages nous ont appris que la routine d'arret de la musique allait en &7C14. Si on désassemble à partir de là, on trouve une routine qui se termine en... &7C3D!

Voila, la musique est prète a ètre sauvegardée. Cet exemple est, vous l'avez constaté, un peu plus complexe que pour Outrun. Il constitue un bon exemple de ce à quoi l'on est en général confronté.

Avant de commencer cette rubrique je tiens à signaler que le texte de cet article a été trouvé sur un site internet et a été recopié sans autorisation de son auteur n'ayant pas pus le joindre. Je tiens aussi à m'excuser pour la faible qualité des transfères des images mais la résolution des photos du site était trop petite pour passer correctement sur CPC. Il me reste à vous souhaiter une bonne lecture...

En Novembre 1971, aprés avoir créée le premier circuit intégré de mémoire vive de 1024 octets, Intel met en vente le premier microprocesseur de l'histoire, le 4004, conçu par Ted Hoff. Le 4004 est un processeur 4 bits fonctionnant à 108 KHz, qui adresse 640 octets de mémoire et effectue 60000 opérations par seconde. En fait, c'est un processeur plutot destiné aux calculatrices et ne fut pas utilisé dans d'autres applications.

Il fallut donc attendre la sortie du 8008 en Avril 1972, une variante du 4004 comme son nom l'indique, fonctionnant à 200 KHz et adressant 16 Ko de mémoire, pour voir apparaitre le 1er micro-ordinateur de l'histoire, et qui de plus est français ! Le Micral de la R2E fut commercia-lisé par François Gernelle en 1973, et le terme "micro-ordinateur" fit la première fois son appa-rition dans un journal Américain en Juin 1973 dans un article consacré au Micral.

Les premiers micro firent timidement leur appa-rition: on peut parler par exemple du kit à monter soit-meme dénommé Mark-8 conçu par Jonathan Titus qui parut dans le magasine Radio Electronics en Juillet 1974.

En Mars 1975, une réunion d'amateurs d'ordina-teurs dénommée Homebrew Computer Club effectue une démonstration de l'Altair, un micro lui aussi basé sur le nouvel Intel 8080 (2 MHz, 64Ko de mémoire adressable et 640000 opérations par seconde). L'Altair dont le nom provient d'un épisode de Star Trek "Voyage To Altair" sera vendu dès Avril 1975 pour 39$ avec 256 octets de mémoire.

Moi en mon temps y'avais des p'tain d'machines... c'est pas comme maintenant y font plus que d'la merde... Jeunes cons tiens !!!

En Mars 1976, Steve Wozniak (26 ans, ingénieur chez Hewlett-Packard) et Steve Jobs (21 ans, technicien chez Atari) finissent leur ordi-nateur qu'ils baptisent Apple I. Ils fondent la société Apple le 1er Avril 1976 et vendent leur ordinateur au magasin Byte Shop pour 666.66$ avec un microprocesseur MOS 6502, 256 octets de mémoire morte (ROM), 8 Ko de RAM et une sortie vidéo sur téléviseur: 175 exemplaires se vendront. Pour 75$ supplémentaires, une carte permettant de brancher le magnétophone permettra de charger le Basic en mémoire, ce qui contribua au succès de la machine. En Avril 1977, Apple Computer présenta l'Apple II, une version améliorée du premier modèle intégrant une carte graphique, un clavier, un port pour manette de jeux, 8 slots d'extension, 4 Ko de RAM et 16 Ko de ROM intégrant le Basic microsoft. Il s'en vendra 35000 exemplaires durant la première année.

En janvier 1977, Commodore, alors fabriquant de calculatrices qui venait de racheter MOS Technologies (le fabricant du célèbre 6502), lance PET (Personnal Electric Transactor), un ordinateur à base de 6502, 14 Ko de ROM conte-nant une version du Basic Microsoft, 4 Ko de RAM, écran monochrome et lecteur de cassette intégré. Il sera vendu au prix de 800$.

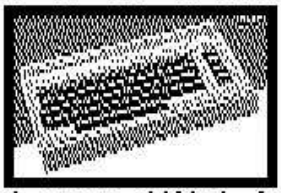
C'est alors au tour de Radio Shack de Tandy de présenter sa première solution informatique. Le TRS-80 est basé lui sur un Zilog Z80, possè-de 4 Ko de RAM et est vendu pour 600$ avec écran, clavier et lecteur de cassette. 10000 exemplaires s'en vendront le premier mois !

Mais les premiers micro-ordinateurs, bien que primitifs dans leurs possibilités (pas de gra-phismes, son austère, mémoire trés limitée) n'en était pas moins trés chers. Il fallut atten-dre 1980 qu'un génial inventeur du nom de Clive Sinclair (qui gagna plus tard le titre de Lord en mérite de ses créations) mette au monde un petit micro-ordinateur dont le principal avanta-ge sera sans nul doutes un prix accessible à toutes les bourses !
Le ZX80 pouvait s'acheter en kit à monter soit-meme pour un gain d'économies supplémentaire. Muni d'un petit kilo de mémoire vive il n'en comporte pas moins un Basic agréable, mais qui malheureusement ne gère pas les nombres à virgu-le ! De plus le microprocesseur gère le rafrai chissement de l'écran, qui a trés facheuse ten dance à clignoter dés que l'ordinateur travail!

Ce petit défaut sera vite rattrappé par le ZX81, sorti comme son nom l'indique en 1981. c'est au contraire du ZX80, un ordinateur trés ré-pandu ! Mais le prix des composants baissant sans cesse et les progrés techniques aidant, les graphismes colorés et les sonorités de plus en plus réalistes allaient faire leur entrée dans la micro-informatique. Alors concentrés

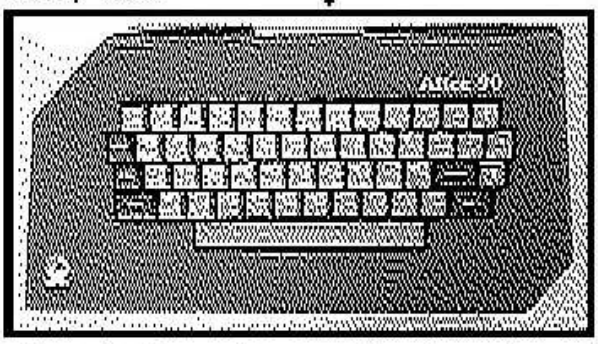
sur l'informatique sérieuse (et certes ennuyeuse !), CBM alias Commodore Business Machines lan-cèrent fin 1981 le Commodore VIC-20, une petite machine familiale au prix accessible, dotée de graphismes colorés, capable de faire du son et possédant un port cartouche (une première !) Seule sa petite mémoire vive, 3 Ko, lui causera du tort.

Les choses prirent une toute autre tournure en 1982: Sinclair introduisit son ZX Spectrum, petit ordinateur d'initiation au succés phénomé-nal en Europe, doté de graphismes fins et d'une généreuse taille mémoire (jusqu'à 48 Ko). Mais la véritable révolution se dénomma Commodore 64 Cette machine mythique fit bien du tort à ses congénères 8 bits pendant de très nombreuses an-nées à venir ! Le Commodore 64 est en effet doué pour les graphismes avec une vraie résolution graphique de 320 x 200 points en 16 couleurs, du moins au maximum, gère 8 sprites et possède un syn-thétiseur sonore telle ment extraordinaire qu'il est encore utilisé de nos jours par certains créateurs de musique élec-tronique ! (Cf: (http://sidstation.com/).



Emboitant le pas à Sinclair et Commodore, d' autres constructeurs se hatèrent de mettre sur le marché une gamme de micro-ordinateurs 8 bits tous incompatibles entre eux, et dont les spéci-ficités eurent tot fait d'engendrer le chaos dans la tete des acheteurs potentiels !

Ce fut donc en 1983 les Oric 1, le Tandy Color Computer et son équivalent Européen le Dragon 32 et d'autres modèles au succès limité et donc peu significatifs Matra/Hachette Alice, Mattel Aqua-rius, etc.)



Meme les Français ne restèrent pas insensibles à ce raz de marée avec Thomson et ses TO7, TO7-70, et finalement MO5.

1985, marqua une certaine rupture dans l' univers des micros 8 bits : le marché était satu ré, essouflé, et seules des machines à bas prix telles que les Amstrad CPC 464 et 6128 ce ven-daient. Celles-ci n' étaient révolutionnaires que dans leur fabuleux rapport Qualité/prix. Ainsi, Un CPC464 en 1984 avec lecteur de cassette et moniteur monochrome coutait moins cher qu'un Thomson MO5 nu !

Arriva alors un évènement de taille : l'intro-duction par Apple du premier Macintosh ! A une époque ou le monde professionnel n'avait pas les yeux tournés vers autre chose que les IBM PC et leur DOS morne et triste, le Macintosh bien qu' en noir et blanc affiche une facilité d'utilisa-tion extraordinaire et des logiciels déja trés au point (Mac Paint, Mac Write). Plus réservé aux professions journalistiques, télévisuelles et artistiques, le Macintosh poursuit toujours son chemin de nos jours.

1986 sonna définitivement le glas pour les ordi-nateurs 8 bits. Sortirent des usines d'Atari les premiers Atari 520ST, de puissantes machines 16 bits construites autour du célèbre Motorola 68000, et livrées en standard avec une très res-pectable mémoire vive (512 Ko pour une machine familiale à l'époque ou memes les pc profession-nels n'en affichaient pas d'avantage !).

Il fut suivi de trés près par les Commodore Amiga 1000, eux aussi construits sur une archi-tecture 16 bits et le meme microprocesseur. Mais les Amiga eurent toujours pour eux d'avoir des atouts graphiques et sonores bien au dessus des Atari ST ! Commodore redésigna peu aprés l' Amiga 1000 en Amiga 500 au boitier trés ressembl-ant à l'Atari 520ST et essentiellement moins cher. De nombreuses évolutions d'Amiga sortiront jusqu'aux Amiga 4000, puissantes stations multi-média.

Les dernières machines 8 bits sortirent en re-tard des usines Thomson (TO8D, puis la gamme réellement obsolète des TO9 / TO9+), mais le standard MSX lui continuait de faire un carton au Japon, avec les évolutions graphiques et div-erses extensions incluses, puis les MSX2+ avec des modes graphiques très élaborés,et finalement les MSX2 Turbo-R, puissantes machines bi-proces-seurs, animés par un processeur RISC (un R800 de la société ASCII à 28 MHz) et le classique Z80A.

Pendant ce temps, les PC prirent petit à petit le marché pourrissants chaque jour un peu plus le monde de l'informatique devenue dérisoirement commerciale et sans interet...

Une légende raconte que de nos jours encore il existe de fières combattants de la dernière heu-re préférant toujours leurs vielles machines à l'ideuse PC ... CPC FOR EVER !!!

Article realise par: samurai@mo5.com ... Conclusion par Iron.

Les émulateurs sont devenus depuis quelques temps une mode. En éffet,beaucoup d'émulateurs éxistent, certains émulent des consoles comme la game boy, la mégadrive ou bien des ordinateurs tel que le CPC, l'atari, l'amiga, le spectrum et d'autres.

A quoi sert un émulateur,qu'est ce que c'est ?

Un émulateur est un programme ,que ce soit sur pc ou mac (nd:Iron:où Amiga...), qui reproduit le fonctionnement d'une autre machine,ainsi il est possible de jouer aux jeux mégadrive sur un pc (ndIron: ou Amiga...).
L'intérét ? Il est multiple. On peut découvrir ou redécouvrir une machine oubliée, ou bien cela peut éviter de se ruiner dans l'achat de consoles ou ordinateurs anciens.

Les émulateurs ont souvent des avantages par rapport au machines émulées,par exemple certains émulateurs CPC possèdent la possibilité d'ajouter des roms, comme l'on ferait avec une ramcard sur CPC. Pour les émulateurs CPC,il y a aussi la possibilité de changer de configuration pour pouvoir tester des démos sans avoir tous les types de crtc chez soi.
Il existe souvent plusieurs émulateurs pour une mème machine, alors c'est à nous de choisir le bon. Sur CPC c'est pareil, il existe plusieurs émulateurs comme cpcému, arnold, winape, no'CPC, caprice...

Alors lequel choisir ?

He bien je vais tenter d'apporter une réponse en testant certains émulateurs.
Voici la configuration du PC qui va servir à tester les émulateurs :
PC intel céléron 800Mhz,128Mo de Ram,Windows Me.
Seul winape et arnold et seront teste car je ne possède pas cpcemu et caprice ne fonctionne pas avec windows millenium. Personnelement, c'est Winape32 que j'utilise.

Pour tester ces 2 émulateurs je vais lancer des démos et décrire ce qu'il se passe pour ensuite faire un bilan et dire lequel des 2 semble le mieux. Je ne sais pas si c'est la meilleure méthode, mais elle me convient.

WINAPE32 :

Ofe 1: Elle passe sans probleme,a part le son qui crachote un peu.

Ofe 2: Les rasters verticaux se prennent le balayage et un problème par moment au niveau du son.

Ofe 3: Elle ne passe pas , la rupture verticale n'est pas émulée,on voie juste un bout de raster et la vitesse de la musique est divisée par 2.

Voyage 1993

Le test de CRTC foire un peu : on m'indique un (ndIron: sur CPC déja c'est pas toujours ça...) 464 crtc 0 alors qu'il s'agit d'un 6128+ qui est émulé.
Part third axis :problème au niveau des rasters.
Hard'n'heavy : marche nickel, si ce n'est que l'on ne voie rien quand on écoute le sample (shift+9 ou control+9)

Big'o'full'o dem'o (Targhan)

1ère part:pas de problème.L'écran n'est pas trés bien calé
2ème part: la démo est ralentie,on ne voit pas correctement les gros rasters, que des bandes de couleurs.
3ème part:on ne voit pas les 2 textes, et le son merde un peu.

Ultimate méga démo(Face hugger)

1) Passe sans problème,mais la musique bugge : gros souffle, j'ai déja rencontré ce problème avec le digitraker sur Winape.
2) ok
3) ok
4) synchro verticale, désyncronisé, zik jouée toute les 2 vbls.
5) part en 320*256 en 7 couleur (turrican): ok
6) 3d, ok
6) animation / ok
7) multiball . ok
8) end part . ok

Plasma de one

Ok. Le scroll est foireux, mais j'ai déja rencontré ce problème sur plusieurs cpc. Ce problème ne peut donc pas ètre imputé à l'émulateur.

Digital orgasm

1) test de crtc : 6128 464Ko de ram crtc 0.
2) ok. le son par moment foire un peu.
3)sur mon cpc+ on ne voie que le bas,ici on voit tout. (ndIron: normal:CRTC 0...)

Méga sound amiga by cjc of ccc

Le son crachote.

Ecole buissonnière by madram/ovl

Aucun problème.

Bactro overflow

Part 1: beuh, elle passe pas: zik ralentie, écran buggé.
part 2: quelques saccades, le reste ca va.

Ze meeting 2000 d'Iron.

Le son merde un peu

Simply the best

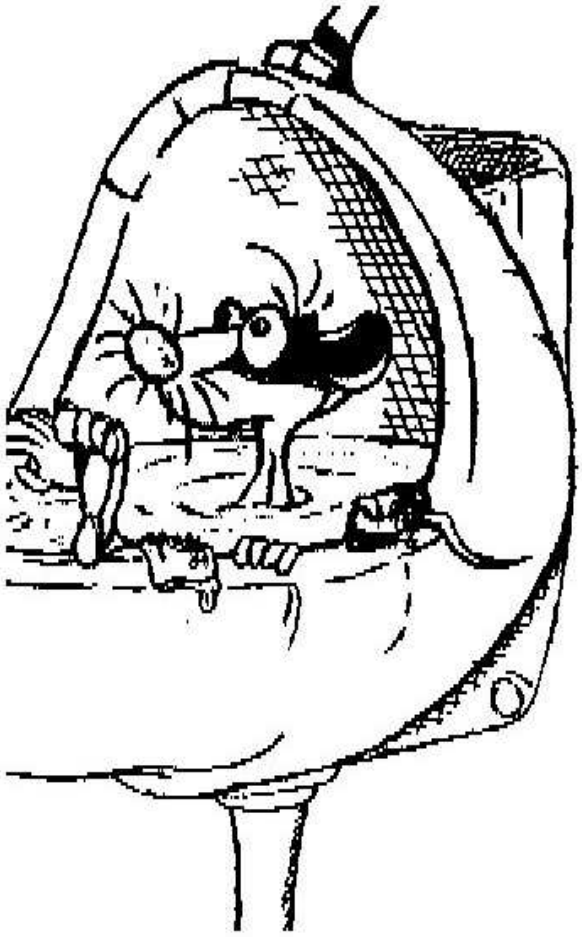
Saccades, son foireux, on voit pas de plasma, juste un changement de couleur.

Conclusion pour Winape 32:

Le CPC est trés bien émulé, il y a juste des lacunes au niveau sonore et quand il y a certaines techniques hard employées, comme la rupture verticale, il ne les émulent pas correctement.
(ndIron: par exemple, le registre 8 du CRTC n'est carrément pas émulé... On dit au revoir au mode entrelacé et aux split border SNIF !!!)
Il faut noter que pour pouvoir s'en servir il faut séléctionner 16bit pour la couleur dans le menu propriété de Windows et pas couleurs vrai, sinon, l'émulateur ne fonctionnera pas corrèctement.

Pour Arnold, les mèmes tests avec les mèmes démos ont servis à le tester. Voici mes conclusions:

Arnold n'émule pas correctement le cpc, surtout au niveau du son, cependant, certains types de ruptures passent trés bien... A noter que Arnold permet de générer des YM5 ce qui est plutot sympa. Enfin le principal reproche qu'on peut lui faire, c'est son énorme lenteur qui fait plutot pleurer quand on voit la différence de vitesse entre un CPC et un PC (beurk !!!). Il faut espérer que personne d'autre que les CPCistes eux memes n'utilisent cet émulateurs, ils pourraient croire qu'un CPC est réèllement aussi lent... Voila, ce test s'arrète ici, je vous dis à la prochaine...



A l'occasion, vous remarquerez que Plissken a lui mème proposé son article sans que je ne le lui demande. Aussi, si comme lui vous avez des idées d'articles, n'hésitez surtout pas à nous les envoyer. Pour plus de facilité lors de la conversion, voici quelques régles à respecter:

- Textes en ASCII avec 48 caractères de large
- Pas de fautes d'orthographe...
- Pas d'attaques personnelles dans un article

Voila, et merci d'avance. Merci à Plissken !!!

INTERVIEW ROMAIN POUR HELPFANZ 2:

1) Salut Romain, je te laisse te presenter...

Salut, je m'appel Romain Giot, j'ai 17 ans et je suis donc encore au lycée en terminale. Je suis codeur chez Bénédiction depuis pas trés longtemps et j'éspère réster sur la scène encore pas mal de temps.

2) On entend pas mal parler de tes préviews trés impréssionantes... Mais sur quoi travailles tu éxactement ?

J'ai l'intention de faire (comme la majorité des codeurs qui débutent, je pense ...) une démo de plusieurs parts. Je ne sais pas vraiment qu'est-ce qu'il y aura comme éffets de présentés, mais en tout cas ce sera éssentiellement du soft car c'est plus facil de faire quelque chose de nouveau ou de mieux que ce qui a put ètre fait étant donné que la plus grande majorité des démos sur CPC sont à base de Hard. En fait, je n'ai pas réellement avancé sur ce projet depuis le Croco 3 (c'est déja loin maintenant !). En poussant la réflexion un peu plus loin, je pourrais mème dire que je n'ai toujours pas commencé à coder cette démo et qu'il serait temps que je m'y mette parce que pour le moment c'est plus un projet qu'autre chose ! Mon intention principale est de faire une démo du style de celle que l'on peut voir sur C64, donc avec un minimum de scroll hard et puis des éffets qui n'ont pas été vus souvent sur CPC (plasma, déformation de graphs, loupe ...)

3) On observe un renouveau du coté de l'intéret que les codeurs portent envers les jeux, pas mal de personnes se mettent à en coder... Qu'en penses-tu ? Cela te donne-t-il envie d'en faire un jour ?

Je trouve ca super que de nouveaux jeux vont bientot sortir, mais pour ma part je ne suis plus vraiment un grand joueur sur CPC... Peut-ètre que j'en coderais un dans un futur trés éloigné, mais pour le momemt je n'en pas l'intention. Je trouve que faire un jeu demande énormément de temps et que c'est quand mème assez dur (je recherche toujours la simplicité). La somme de travail a fournir est beaucoup trop importante pour moi !

4) Tu fais parti des derniers arrivants sur la scène; que penses-tu de celle-ci, quelles ont été tes impréssions lors de ton 1er meeting ?

Je ne vois pas trop ce que je pourrais dire sur la scène, si ce n'est qu'elle est trés acceuillante et sympathique. Ce qui est génial c'est qu'il y a de trés grands passionés qui en font parti depuis plus d'une dizaine d'années ! Je ne pense pas que ce soit le cas sur les autres machines. Le seul reproche que j'ai à faire est son inactivité. J'ai une trés mauvaise mémoire alors mes souvenirs sont pas mal vagues. C'était le Croco 2 ou j'est (!!!) pu voir autant de cpcistes pour la première fois. Je me sentais un peu perdu, mais comme j'avais déja vu les participants du WACWE! je n'ai pas été vraiment (un peu quand mème) inquiet de la folie qui planait dans la salle.

5) Je reproche souvent l'inactivité des codeurs. Quel est ton avis la dessus ?

Je suis tout à fait d'accord avec toi, la scène est inactive, mais mème en faisant cette critique nous restons toujours autant inactifs. Je pense que tout le monde pourrait faire l'éffort de sortir un petit quelque chose tout les ans, mème si il n'y a rien de génial, tout en continuant a bosser sur un projet plus important qui sortira plus tard et sera trés impréssion-nant. Le fait de ne rien sortir doit aussi ètre la recherche de la simplicité car je pense que tout le monde code de nouveaux trucs de son coté mais a la flegmme de les finaliser puisque c'est vraiment trop chiant.

6) CPC ou CPC+ ? Certains détèstent (a tord) le +. Et pour toi qu'en est-il ? Penses-tu que les critiques de cette machine soient réellement fondées (méssage subliminal: non) ?

Je ne connais pas trop le plus étant donné que je n'en ai pas et que les productions spéciales + se comptent sur les doigts d'une main. Par contre il est dommage que peu de personnes codent dessus car avec les sprites hards et la rupture + on peu faire pas mal d'effets (à base de précalculs ou non) beaucoup plus impréssionants que sur cpc old (sans s'

occuper de la palette +) et innovants.

7) Quelles sont selon toi les meilleurs démos CPC ? Sur quels critères te bases-tu ?

1.ECOLE BUISSONIERE	(pas la peine de se justifier)
2.THE DEMO	(la part de LONGSHOT est vraiment impréssionante. Le délai entre le debut du projet et sa finalisation l'est encore plus.)
3.S&KOH	(pas la peine de se justifier non plus)
3.ULTIMATE MEGA-DEMO	

En troisième position j'hésite beaucoup car je trouve la S&KOH vraiment trés impréssionnante au niveau de sa téchnicité (personne n'a fait mieux) et la ULTIMATE qui a du en décoiffer plus d'un lorsqu'elle est sortie. Je la trouve pas mal impréssionnante aussi (mais pas pour les mèmes raisons).

8) Penses-tu te tourner plus vers le soft ou le hard ? Quels sont tes projets ?

On dirait la question 2... Pour le moment je bosse surtout sur le soft parce que je trouve ça quand mème rudement plus facil que le hard. Le hard ne m'attire pas trop d'une part parce que j'ai pas envie d'apprendre à quoi servent tous les registres , je ne connais que les reg 4,9,12 et 13 pour les autres j'ai toujours besoin d'une doc. L'autre raison est le problème de compatibi -lité d'un CRTC a l'autre. C'est pourquoi pour le moment je me suis limité à la rupture simple et ligne à ligne. Pour les projets ce sont toujours les mèmes qu 'à la question 2...

9) L'entrée dans un groupe (Bénédiction) t'as t'elle apportée quelque chose ? Avez-vous des projets dans le groupe ? A part toi et Eliot, y'a qui et y font quoi ?

Je sens la question piège... Il est vrai que de faire parti ou non de Bénédiction n'a pas changé grand chose au niveau du rythme des sorties du groupe. Pour moi, c'est plus motivant de faire parti d'un groupe, je ne passe pas forcément plus de temps devant mon CPC, mais je ne code plus que pour moi maintenant. Peut-ètre que ça me forcera à faire les trucs le mieux possible, qui sait ? Sinon pour les membres du groupe, y'a THE VILLAIN le boss, ANTITEC codeur et musicien, RAINBIRD codeur et graphiste ARNO codeur et graphiste, ELIOT codeur et musicien, et moi codeur seulement. Bien sur qu' il y a des projets (ca ne veut pas dire qu'on bosse dessus..). Une démo avec une part d'ELIOT, une autre d'ARNO et l'intro de moi (C'est d' ailleurs en faisant cette intro que j'ai appris à coder), DIGITAL PRESS 4, l'EUROPEAN MEGA DEMO, une mini-démo d'un mini meeting qui date de plus de 6 mois, la part d'ELIOT pour le Croco et surement d'autres trucs que j'ai oublié. Je préfère ne pas me prononcer sur les dates de diffusion.

10) Le refus de trouver un pseudo serait-il une forme de rébéllion contre la société actuelle, l'américanisation et la mondialisation ?

Je ne pense pas qu'avoir un pseudo est la chose la plus importante. J'en aurais un, un jour, mais pour le moment je n'en ai pas l' utilité étant donné que je ne sort rien et puis je m'en fous un peu.

11) Sinon c'est quoi tes gouts musicaux; culinaires (de cuisine quoi !) ???

Pour la musique j'écoute plein de trucs. J'aime bien Bob Marley, Jimi Hendrix, Black Sabbath, Noir Désir, Tryo, Led Zepplin, Deep Purple ... En fait j'écoute pas mal de truc sauf du rap, de la techno et de la musique classique. Pour ce qui est de la bouffe j'aime tout sauf le lait, le chocolat et le fromage. Curieusement j' aime bien les yahourts, le nutella et la raclette.

12) Que penses-tu du retour des codeurs du moyen age tel qu'Overflow ?

J'sais pas je suis arrivé sur la scene à peu prés en mème temps que lui, je n'ai pas pu sentir de différence !! Par contre je ne vois pas pourquoi tu as mis la question au pluriel, parce que c'est le seul "ancien" codeur qui à

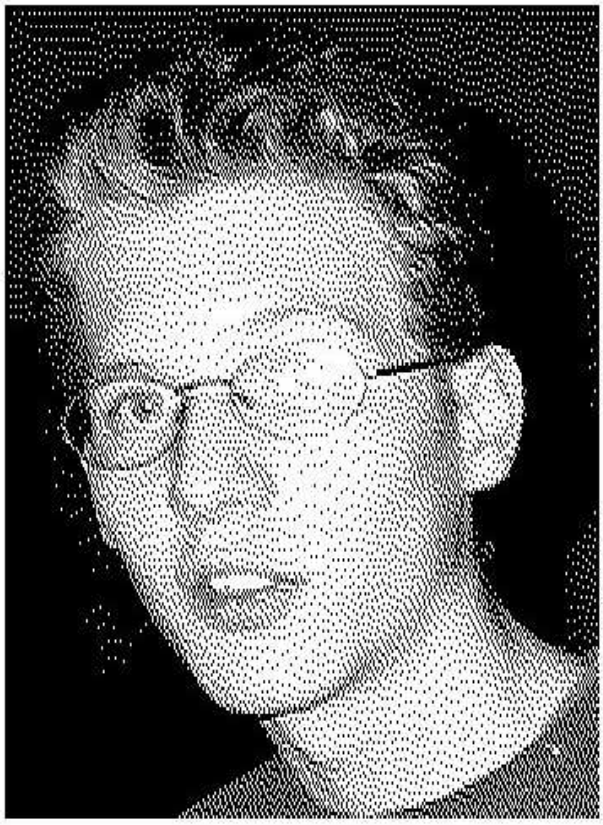
fait un "réèl" retour.

13) Et les OFE démos, quel est ton avis la dessus; soupçonnes-tu quelqu'un ?

Je sais pas. Je soupçonnerais bien OVERFLOW qui fait semblant de mener son enquète (en croyant qu'on ne penserait pas que c'est lui alors qu'en fait c'est réèllement lui) pour trouver le coupable et en foirant volontairement ses ruptures (en pensant que comme ca personne n'aura l'idée de le soupconner).

14) Merci d'avoir répondu a cette interview, je te laisse t'exprimer sur ce que tu veux ...

Ca c'est cool mais je n'ai rien à dire si ce n'est qu'il est impératif que de nouveaux trucs sortent.



Romain au Croco numero 2

Bon, eh bien voila, helpfanz 2 c'est finit... Je tiens à remercier tout les participants surtout Romain, Plissken et Tom & Jerry pour leurs articles. Pour conclure ce numéro, je tiens juste à préciser quelques trucs:

Premièrement, mème si le fanz a été imprimé sur PC, celui-ci a été entièrement réalisé avec un CPC. J'ai en effet utilisé le génial Oxford PAO pour le réaliser. Compte tenu que je n'ai pas d'imprimante, il m'a fallu me résoudre à utiliser un PC pour l'imprimer. Malheureusement et je le regrète, aucun émulateur n'émule le port imprimante d'un CPC en redirigeant l'information vers le port imprimante d'un PC. Ainsi, il m'a fallut déja trouver un émulateur qui gérait graphiquement le mode 2 pour enfin faire des captures d'écran des articles sur Oxford PAO pour enfin tous réunir en fichiers. Ainsi, je peux vous donner les fichier JPG au besoin.

Je tiens aussi à m'exprimer sur le triste phénomène qu'est internet (le premier qui dis "l'internet" j'lui en met une Ok ?). En effet, depuis déja pas mal de temps, beaucoup d'entre vous avez céssé le swapping qui faisait vivre la scène. Maintenant nous pouvons nous rendre compte des dégas en voyant la perte d'activité des fanzines directement due à la mailing list (quelqu'un pense le contraire ? Non ? Ok!). Vos discutions sont devenues plates et sans interet. Posez vous la question de savoir à quoi vous sert la mailing list, vous verrez alors qu'elle ne vous apporte que trés rarement quelque chose. Pour ma part, le swapping continu et quelques personnes tels que Romain et les frères Thomasette répondent encore au courier. Pour vous donner un exemple, moi et Romain nous sommes écrits 20 lettres entre le 12 septembre 2001 et le 9 mars 2002. Et ses lettres n'étaient pas vides puisqu'on y parlait entre autre de plasma; dots tunnel; 3D mappée; sinus scroll; DMA son; fractal; rupture; optimisation; et bien d'autres choses... En résumé, 1 à 2 lettres par semaines et toutes remplies de débats auquels sont venus ce greffer d'autres personnes tel qu'Overflow pour le bump. C'est aussi ça l'activité: discuter d'effets encore jamais faits sur nos bonnes vieilles machines où tout simplement de nous aider entre nous. Pour conclure, je vous dirais donc que maintenant vous savez de quoi parler plutot que de vous demander chaque jour quel temps il fait. IRON