La Pomme Illustree N°1 - Janvier

REDACTION
Nibble,Bandit II, Perfect Bugs, Ferox, Or,Azebulon

COUVERTURE
Inconnu à ce jour

REDACTEUR EN CHEF
Pour ce numéro : Perfect Bugs

DIFFUSION
Phoenix Corp.
TDGD, et vous

ILLUSTRATIONS
Se référer à Couverture

PUBLICITE
Phoenix Corp.

IMPRESSION
Phoenix Corp.
A.r.n., Tib et tous les autres que j'oublie...

MATERIEL
IIgs 1,2 Mo, LaserWriter
AppleWorks 3.0
et AppleWorks GS pour la mise en page

SIEGE SOCIAL
Minitel,Rtel,36.15
Bal:PHOENIX CORP

Et voilà, le numéro 1 de la Pomme Illustrée est enfin disponible. Vous allez sans doute noter un changement de style étant donné que c'est moi (Perfect Bugs) qui m'occupe de ce numéro, Nibble étant indisponible et les autres membres du Phoenix manquant sérieusement de temps (moi aussi, mais on va pas chipoter : faut bien que quelqu'un se dévoue). Le N°0 était 100% Phoenix, mais alors celui-ci... Car comme d'habitude (c'est une manie chez nous), on se repose sur ses lauriers (des lauriers, où ça ?), et résultat , Ce numéro est mis en page en quatrième vitesse par votre serviteur pendant le week end du 12 Janvier, soit une semaine avant la Jihailde party. D'ailleurs, à l'heure où j'écris ces lignes, je ne sais meme pas si on sera pret à temps. A noter tout de meme (tout n'est pas noir dans cette vie) que l'impression est une impression laser, ce qui devrait améliorer sérieusement l'aspect général du mag. Par contre, on passe au format A4 : En effet, la demande pour le premier numéro ayant été relativement forte, et le nombre de personnes pouvant faire des photocopies A3 ne courant pas vraiment les rues, on s'est trouvé en rupture de stock (pour tout vous dire, aucun membre du Phoenix n'a pour l'instant d'exemplaire propre), d'où la nécéssité de changer de format, ce qui est maintenant chose faite.. J'espère d'autre part que le nombre de fautes d'orthographes aura diminué de façon significative. Sinon, ne nous en tenez pas rigueur : C'est pas évident de corriger des textes et de faire une mise en page en deux jours. Je laisse maintenant la parole, ou plutot la plume, enfin plus exactement le clavier, à Ferox, lequel a, si je ne me trompe, pas mal de choses à dire.

Happy New Year

Happy new year...

Salut les aminches. Salut les autres ıussi, tiens, tant qu'on y est.

Alors ca y est. Ils ont remis ca. Une ois de plus. Il y a environ trois ;emaines, un groupe d'illustres nathematiciens, aides des masses ›opulaires (des miasmes morbides des refons de la civilisation decadente, levrais-je dire), sont arrives, au bout l'un long cheminement et d'une lemonstration qui n'alla pas sans mal not' collegue de math spe y a rien ;ompris), a l'impressionante conclusion ;uivante: 1990 + 1 = 1991 !!! Et ca leur a ›ris un an, pile 365 jours, pour trouver ca. Et la, tout s'en trouva boulverse, 'euphorie pris une empleur demesuree, la olie s'empara du monde entier, et comme exactement un an auparavant, ils se sont lis que quand meme il fallait feter ca.

Et donc, voici tout un chacun de se nettre a reprendre un bain, de porter son costume du Dimanche en plein Lundi, l'arreter de travailler parfois pendant ›lusieurs semaines, enfin bref, la 'evolution; et causee par quoi ? Par une geniale addition. Einstein lui-meme, nous 'evelant son extraordinaire E=mc2, ne put provoquer un tel bonheur dans le monde entier. Les hommes de toutes categories, dans le monde entier, sortirent de chez eux, allerent faire les grands et petits magazins, s'acheter des cadeaux, de la bouffe de luxe, et autres petites gateries... Ce '1991', si bien decouvert, fut aussi la proie de tous les mass-medias: television, journaux, radio, etc...

Un fait cependant retint mon attention: l'autre jour, prenant une grande feuille de papier et un crayon suffisemment long pour mener a bien ma quete perilleuse, je me mis a la recherche du prochain nombre qui allait sans doute ebranler le monde. Peut-etre presumais-je de mes forces, mais, apres tout, il fallait que j'essaye. Et j'ecrivis la ligne suivante:

```
10 PRINT 1991+1
```

Ceci etait le premier pas. Je ne savais pas pourquoi, mais il me semblait que je tenais le bon bout. Mais ceci, je ne pouvais le deviner seul. Aussi allais-je me poster devant mon clavier, enfoncais-je le button situe sur le cote droit d'un ecran siegeant dans l'environnement, passais-je une main a l'arriere du parallelepipede gris pose a proximite immediate du clavier et de l'ecran suscites, et fis-je basculer l'interrupteur

noir qui y etait present. Un bip sonore se fit entendre, des textes a la signification obscure apparurent a l'ecran, s'effacerent, laisserent place a d'autres non plus lumineux, et un curseur apparut.

Je tapais lentement la ligne ecrite sur la feuille posee a cote de moi, faisant bien attention a chaque caractere entre de maniere a ne pas me tromper, puis j'appuyai sur la touche de validation. Le programme etait termine. Prenant ensuite mon courage a deux mains, je tapai les trois lettres magiques R, U et N.

Mon emotion etait a son comble. J'etais conscient que j'allais peut-etre decouvrir la prochaine cause de revolution mondiale. Je validai mon entree. Je ne pourrai jamais decrire l'angoisse de l'attente que je vecut a se moment la. Allait-ce marcher ?

A suivre.

GAME OVER

Vous allez enfin pouvoir depasser la deuxieme scene de Space Ace, grace a cette solution tapee par votre serviteur.

La rubrique GAME OVER de ce numéro portera sur un logiciel relativement récent mais qui n'a été que très peu diffusé, de par son nombre de disques assez (9 dsks!) élevé : Il s'agit de Space Ace. Ce jeu, assez beau, est d'un intérèt ludique très limité, et de plus tres difficile. Mais il est suffisament beau pour figurer en bonne place dans toute ludothèque. Je précise enfin que la solution ne vient pas de moi, m'est m'a été postée sous forme de photocopie de quelques pages d'une célèbre revue uniquement dédiée aux ST et autres Amiga (Beurk...). De plus, certaines scenes ont ete supprimees dans la version GS. A vous de voir.

Scène 0 : Introduction

Kimberly et Dexter aperçoivent la station spatiale du Commandeur Borf. La falaise s'écroule, Kimberly plonge dans le vide et disparait vers la station.

Scène 1 : Borf arrive vers Dexter. Dès qu'il a tiré deux fois, Joystick à droite. Dès que Borf s'approche, Gauche puis tout de suite Bas.

Scène 2 : Droite pour éviter le bras du robot, puis attendre qu'il se relève et gauche. Attendre que Dexter s'accroupisse et gauche, puis encore gauche dès que le second bras du robot se relève.

Scène 3 : Bas pour faire courir Dexter en dehors des lasers des obots, puis Haut dès qu'il est sur la petite butte.

Scène 4 : Attendre que le vaisseau descende vers la station spatiale de Borf puis Haut pour atterrir. Dexter sort en courant de son vaisseau.

Scène 5 : Feu tout de suite pour tuer le monstre vert.

Scène 6 : Un énorme bras plonge sur le pont où se trouve Dexter. Dès que le bras est en bas, Droite puis lorsque Dexter s'apprète à sauter, juste avant que le bras ne l'écrase, Haut.

Scène 7 : Attendre que la plate-forme soit en bas, puis Droite et à nouveau Droite dès que Dexter est sur la plate-forme.

Scène 8 : Dès que Dexter court vers le monstre, Bas puis tout de suite après, Droite.

Scène 9 : Dexter court vers le monstre, alors Bas puis immédiatement Gauche pour éviter une mort certaine.

Scène 10 : Dès que Dexter est attrapé par le monstre, Feu.

Scène 11 : Attendre que les deux bonshommes bleus apparaissent à l'écran, puis Haut.

Scène 12 : Attendre que les deux chiens surgissent de chaque coté, puis Haut.

Scène 13 : Arrivé à l'intersection, Droite pour échapper aux chiens.

Scène 14 : Une fois que les chiens apparaissent au premier plan, Haut pour se débarrasser d'eux une bonne fois pour toute.

Scène 15 : Dès que les robots apparaissent, Droite pour éviter leurs tirs.

Scène 16 : Laissez Dexter courir jusqu'au fond, puis Gauche.

Scène 17 : Une fois au fond, Gauche pour éviter l'explosion.

Scène 18 : Dexter court encore un peu, puis Droite pour éviter le tir laser.

Scène 19 : Dexter court toujours. Un peu avant l'échelle, Haut pour monter.

Scène 20 : Borf attaque Dexter. Dès qu'il s'apprète à frapper, Feu.

Scène 21 : A nouveau Feu pour parer le coup de Borf.

Scène 22 : Borf tente de donner un nouveau coup. Feu pour parer. Dexter tombe alors à terre et Borf en profite pour essayer de l'avoir. Droite pour esquiver le coup.

Scène 23 : Dexter attaque Borf. Feu dès que l'image apparait pour réussir le coup, puis Bas pour éviter la réplique de Borf.

Scène 24 : Dexter est à terre et doit parer le coup de Borf. Feu dès que ce dernier s'approche !

Scène 25 : Borf donne de vastes coups avec son arme. Haut pour esquiver le premier coup en sautant au-dessus, puis Bas pour esquiver le second coup en plongeant à terre.

Scène 26 : Droite permet d'esquiver le coup de Borf, puis Bas pour sauter sur son dos.

Scène 27 : Les gardiens bleus de Borf courent vers lui pour l'aider. Ne rien faire pendant cette partie...

Scène 27bis :Jusqu'à ce qu'on voit sauter

vers la corde. Gauche pour qu'il l'attrape à temps!

Scène 28 : Haut pour que Dexter saute sur la plate-forme où se trouve Kimberly.

Scène 29 : Droite dès que la plate forme sur la mer de lave, pour sauter avant de s'enfoncer.

Scène 30 : Borf tire avec son rayon Infanto...

Scène 30bis : Dès que le tir arrive, Droite pour éviter de se faire toucher.

Scène 31 : Borf tire à nouveau, au moment où Dexter court vers une intersection. Gauche juste avant le croisement pour éviter le tir.

Scène 32 : Borf tire encore. Dès que le tir arrive, Droite pour l'éviter.

Scène 33 : Gauche immédiatement pour que Dexter place le miroir, puis Droite rapidement pour éviter le tir.

Scène 34 : Fin du jeu. Ne comptez pas sur moi pour vous dire ce qui arrive à ce moment là, non mais. Un peu d'effort que diable !!!

Je comptais donner d'autre part la solution de The Immortal, mais je coince à un niveau. Donc je ne donnerai ici que la solution de la scène finale, qui a semble t-il posé de nombreux problèmes :

Après avoir ouvert le coffre, allez au centre de la pièce carrée : Vous tombez alors dans une trappe, pour vous retrouvez face au dragon. Enclenchez le sort Blink.

- Le dragon va cracher 6 fois du feu : Vous devez parez le coup en disparaissant (Grace au Blink).

- Puis il va reprendre son souffle avant de tenter à nouveau de vous carboniser : Vous devez cette fois utiliser le Fire Protection (qui dure plus longtemps), mais attention : Seulement quand celui ci s'apprête à lancer son jet de flammes (Tout est dans le timing !).

- Lorsque le dragon s'arrète de nouveau, montrez lui l'amulette (sans lire les runes !). Mordamir apparait alors et récupère son amulette.

- Enclenchez le sort Statue et tenez vous prêt à vous pétrifier pour survivre aux éclairs que lance Mordamir (là encore, un bon timing est nécessaire), et ceci trois fois de suite.

- Mordamir va alors lancer un sort sonique : Dès que le cercle autour de lui apparait, activer Sonic Protection.

- A nouveau deux éclairs, puis une Mort ailée qui vous fonce dessus pour vous décapiter : Utilisez les trois derniers

Statue.

- A ce moment, il ne vous reste plus qu'un sort : Mordamir s'arrète alors et commence à vous raconter son histoire avant de montrer son amulette : Utilisez alors Magnetic Hands pour la récupérer. Mordamir est ainsi sans protection façe au dragon, et celui ci se fait un plaisir de la transformer en barbecue. Puis il disparait, et vous avez terminé un magnifique jeu.

Voilà, c'était tout pour ce numéro. Si vous avez des question, n'hésitez pas : Bal **PHOENIX CORP** sur RTEL.

Et pour terminer : La date des concours approchant à grand pas, je n'aurais plus le temps de m'occuper de mon GS. Si quelqu'un est disponible pour s'occuper de la rubrique GAME OVER, qu'il me contacte sur notre BAL.

UN PEU DE MATHS

Un specialiste de la question vous propose ici une etude sur les integrales

Informatique mathématique

CALCUL APPROCHE D'INTEGRALES

J'en connais beaucoup qui ont du sentir leur poil se hérisser en voyant l'entête de cet article. Tant pis pour eux, mais aprés tout l'informatique est aussi un des rares domaines d'application quasi-directe des mathématiques; ça peut-être intéressant d'en avoir un exemple, même trés modeste. Ne vous inquiétez pas, je ne vais pas vous défoncez le crane avec de grandes théories, et le niveau requis pour l'utilisation de qui est dit par la suite ne dépasse pas celui de la Terminale (toutes sections confondues...), et peut tout de même être d'un intéret assez intéressant, pour des futurs bacheliers par exemple.

Le calcul intégral est en effet une des pierres d'achoppement de pas mal de résultats, et est introduit en Terminale sous sa forme la plus simple (ne vous en déplaise). Pourtant, le gros problème de ce genre d'exercice c'est de ne pas pouvoir disposer d'un moyen de controle sur et rapide. Or, il existe depuis maintenant trés longtemps des techniques de calcul approché d'une intégrale, facilement programmables sur ur ordinateur ou une machine à calculer et qui permettent des vérifications numériques des calculs formels que l'on peut ètre amené à faire.

NOTATIONS :

Les crochets ne sont pas toujours disponibles : l'intervalle des nombres compris entre a et b sera noté (a,b).

Les traitements de texte scientifiques ne sont pas généralisés en mode texte. L'intégrale de a à b d'une fonction f(x) par rapport à la variable x sera donc notée désormais : **Int(a,b,f(x),x)**.

Exemple : **Int(1,2, p/t ,t) = p * ln(2)** . On aura compris que p est un paramètre, et que la variable d'intégration est ici t (sinon, on relis en faisant un peu plus attention...).

Dans les programmes qui suivront :
En Pascal, on aura défini la fonction f par la séquence :

```
Function f(x:real):real;
Begin
    f:= 1/X ;    (* 1/X c'est pour les
test bien entendu *)
```

End;

Pour la Casio n000, on aura mis dans le programme 9 (ce n'est pas un obligation que ce soit le 9 bien sur, mais je vais pas encore rajouter une lettre) :

1/X -> Y (* 1/X est encore un test, mais les noms X et Y sont ici impératifs *)

Enfin, pour la plupart des tests, je vous conseille d'utiliser la fonction 1/X entre 1 et 2; puisque Int(1,2,1/X,X) = ln(2) = 0,69314718056 .

LA METHODE : Elle est issu de la matérialisation physique d'un calcul d'intégrale (si, il y en a une !) : l'évaluation d'une aire. Je vais tacher de faire une ou deux illustrations, reportez-vous donc éventuellement aux figures mais en bref :

- soit une courbe définie par l'équation y=f(x), et tracée sur le papier dans l'intervalle (a,b). Jusque là vous suivez ?
- Soit A l'aire de la surface S définie par l'axe des abscisses, les deux droites verticales d'équation x=a et x=b, et la COURBE y=f(x).

(Visualisez simplement une plaque avec trois cotés droits et le quatrième courbe).

- A = Int(a,b,f(x),x) (Eh oui...)

Il s'agit donc de trouver un moyen de calculer numériquement une valeur approchée de cette aire pour évaluer l'intégrale.

Le problème est évidemment que l'on n'a pas ici affaire à une surface simple genre carré ou rectangle, la solution consiste à s'y ramener.

Pour cela, on va découper l'intervalle (a,b) en un certain nombre n de morceaux de même largeur ((b-a)/n) et on va considérer que, sur ce morceau (que l'on peut rendre aussi étroit que l'on veut en faisant augmenter n), la fonction f 'varie peu'. On va donc tout bètement remplacer la partie de courbe qui défini le bord supérieur de la surface (réduite au petit morceau) par une droite. On retombe alors sur l'aire d'un rectangle, qu'on va bien sur calculer par la méthode d'approximation dichotomique de la moyenne quadratique des points d'intersections des polynomes d'interpolation de Lagrange et de Bernstein associés au cours du Nil blanc supérieur. On génére bien évidemment une erreur, mais en faisant augmenter n, on diminue cette erreur proportionnellement, en augmentant le nombre de rectangles utilisés pour approcher la surface S. (Un petit dessin éclaircira tout rassurez-vous, à moins que vous n'ayez déjà sauté sur la formule des accroissement finis...) Le calcul ainsi effectué est appelé 'somme de Riemann', et il constitue mème une des méthodes de définition de l'intégrale (quoique pas trés appréciée de nos jours).

L'algorithme est alors trivial : on calcule la somme des aires des

rectangles successifs, en prenant pour hauteur la valeur de f à une des bornes. Voilà les programmes, un en Pascal classique, l'autre dans un langage plus ésotérique, mais que vous reconnaitrez comme étant celui des calculatrices de la série Casio fx (à partir de la 4000) dont l'usage, il faut l'avouer, est assez généralisé.

```
Function Riemann(a b:real;
n:integer) : real;
  var s,dx : real;
      i : integer;
  begin
      s:=0;  (* s est la valeur courante
de la somme de Riemann *)
     dx:=(b-a)/n;          (* dx est le
'pas' d'intégration *)
     for i:=0 to n-1 do
          s:= s + f(a + dx*i);
    s:= s * dx;
    Riemann := s;
 end;
```

Commentaire : la ligne exécutée par la ›oucle peut surprendre à première vue. En ›ffet, au lieu d'ajouter l'aire de chaque ›etit rectangle à s, on n'y ajoute que sa auteur !!! Cela vient du faire que la ›rgeur est toujours la même (dx), et que uisqu'on la met intuitivement toujours n facteur quand on écrit la formule, il 'y a aucune raison de s'en priver du point e vue informatique, d'autant que cela ptimise le programme. Et c'est pour cela u'on à la ligne s:=s*dx. Si ça vous amuse, ›ettez le '*dx' dans la boucle, mais vous voyez bien que cela revient au même, si ce n'est que ça ralentit le programme.

Pour la Casio:(NB: '->' symbolise bien sur la petite flèche des Casio)

```
"RIEMANN (f->P9)"
"DE A"?->A
"A B"?->B
"EN N"?->N
(B-A)/N->I : 0->S : A->X
Lbl 0
Prog 9: S+Y->S: X+I->X
Dsz N: Goto 0
S*I->S
```

L'erreur générée par cette méthode est assez importante. Amusez-vous à faire le test précisé plus haut, avec N de l'ordre de 100. Le programme prend déjà un certain temps sur la Casio, et pourtant, la précision ne dépassera pas le 1/1000. (Juste le 0.69 est garanti). C'est du au fait que l'erreur générée est à peu de chose prés en 1/n (le peu de chose prés reste important pour les puristes, mais cette remarque ne s'adresse pas à eux, na), ce qui est on ne peut plus minable (Si, on peut, d'accord, mais faites pas chier au fond de la classe...).

N'empèche tas de larves, qui si vous vous étiez abominablement planté dans le calcul en vous mélangeant les pattes entre le logarithme et les fonctions rationnelles (hérésie !); ce simple petit programme vous aurez révélé l'erreur.

Et comme vous n'ètes tout de même pas heureux que le pape informatique vous ait

dévoilé votre erreur, et que votre esprit débile ne se sentira pas soulagé tant que votre minable calculateur ne vous aurez pas gerbé les 8 décimales insignifiantes nécessaires au repos de votre âme; on va faire mieux, beaucoup mieux !!!

Je vous vois déjà frétiller d'avance, en train de vous exclamer "Enfin, il va faire sa prière au dieu de l'informatique, il va plonger vers les tréfonds de la machine pour caresser directement le microprocesseur et l'obliger à faire tourner son moulin plein tube dans l'objectif de monter n vers des cimes jamais atteintes, on va peut-être même s'amuser à tripatouiller le hard..."

Et bien non !!! Le dieu de votre prof le plus aborré est plus puissant que toutes les foudres informatiques. Au lieu de vous injectez une double dose de détournement de vecteurs d'interruptions, je vais sournoisement élever brutalement de deux dégrés le niveau de l'interpolation.

Parce que, bande d'ignares, pendant que vous faisiez votre bète découpage en rectangle, et que le père Riemann s'évertuait à essayer de vous faire comprendre que les plupart des fonctions usuelles sont limites uniformes de fonctions en escalier, que cela revenait à approcher la courbe par une interpolation d'ordre 0 (en gros, remplacer la courbe par une droite horizontale autour du point où elle prend effectivement cette valeur); le père Simpson travaillait lui.

Plus sérieusement, vous avez peut-être capté que vous aviez approché la courbe par une droite horizontale. Si vous ètes moins borné que vous n'en avez l'air, vous vous ètes demandé si on ne ferait pas mieux de l'approcher par une droite en biais, ce qui fournirait intuitivement une meilleure approximation. Mathématiquement, cela revient à approximer la fonction par ce que l'on appelle un polynome interpolateur lui facilement intégrable. La droite horizontale correspond au polynome de degré 0 (y=cte), la droite en biais correspondant au polynome de degré 1 (y=ax+b). La méthode de Simpson consiste à pousser l'interpolation jusqu'à l'ordre 2, et donc à approcher la fonction par un arc de parabole (attention, là c'était important, relisez, because le dessin est pas hyper clair). On peut alors vérifier, principalement grâce à la formule de Taylor (pour les initiés), que l'erreur effectuée dans le calcul de l'intégrale est beaucoup plus faible, énormément plus faible, alors que le temps de calcul n'est guère augmenté. La formule est en contrepartie un peu plus monstrueuse et surtout beaucoup plus délicate à intuiter, c'est pour ça qu'on vous la donne. Bon, on passe au programme :

```
Function Simpson(a,b:real;
n:integer):real;
   var s,dx:real;
      i:integer;
  begin
     s:=0; dx:=(b-a)/n;
```

```
    for i:=0 to n-1 do
        s:= s + f(a+i*dx) + f(a+i*dx+dx) +
4*f(a+i*dx+dx/2);
    s:=dx*s/6;
   Simpson:=s;
  end;
```

Commentaires : On a encore opéré une mise en facteur, celle de dx/6 cette fois-ci. Dans la formule, on laisse traditionnellement le 1/6 dans la somme pour faire apprécier l'homogénétité (4+1+1 = 6, n'est-il pas vrai...).

Pour la Casio:

```
"SIMPSON (f->P9)"
"DE A"?->A
"A B"?->B
"EN N"?->N
(B-A)/2N->I: 0->S: A->X: Prog 9
Lbl 0
S+Y->S: X+I->X
Prog 9: S+4Y->S: X+I->X
Prog 9: S+Y->S
Dsz N: Goto 0
S*I/3->S
```

Commentaires : Le pas d'incrémentation utilisée est ici dx/2 (dans I), ce qui explique que la division finale ne soit que par 3 (le facteur 2 restant implicitement inclu dans I). Par ailleurs, par souci d'optimisation, on réutilise la valeur de f calculée pour a+k*dx+dx (k: cf formule) au début du cycle suivant où on a besoin de f pour a+(k+1)*dx. D'où l'appel à Prog 9 seulement 2 fois dans la boucle.

Cette fois-ci, vous allez vous rendre compte de la puissance d'une optimisation mathématique du problème. En effet, même en travaillant en langage machine (et donc au prix d'un travail considérable), l'utilisation de la somme de Riemann pour le calcul numérique (dont ce n'est pas la vocation d'ailleurs) ne permettrait d'offrir une précision honorable (disons 6 décimales) qu'au bout d'un temps certain. Là, avec cette méthode, avec n de l'ordre de 10, vous aurez déjà pratiquement toutes les décimales que votre machine peut sortir.

D'ailleurs, entre parenthèses, j'aimerai faire remarquer aux badabeux fanatiques de précision qu'avec cette méthode, il est ABERRANT de mettre n trop supérieur à 20 ou 30. En effet, la précision théorique affectée à une valeur de n de l'ordre de 100 par exemple est généralement supérieure à celle des calculs de votre machine et dans ce cas là vous travaillez tout simplement dans le vide, et il ne peut en sortir que des problèmes (d'ailleurs, essayez avec 10 000 si vous avez la patience, et vous verrez que la méthode finira par diverger, mais ce n'est pas la faute à Simpson.)

Vous voilà donc armé maitenant pour affronter toutes les traitrises de votre

professeur de mathématiques préféré. A signaler que ce genre de programme ne vous permettra de toute façon pas de trouver un résultat (à moins de tomber sur 1,41421 ou 1,73205 ou 3,14159 ou 2,71828), mais qu'il est un indicateur pratiquement infaillible d'erreur. Du moins jusqu'au bac, aprés, il y a toujours des tordus qui pourraient vous en...er en beauté. A titre d'illustration, j'aurais perdu un point au bac en math si je n'avais pas fait cette vérif. Comme quoi l'article peut servir à certains, et puis il m'aura permis de vous faire chier quelques minutes en période de vacances, et ça, c'est toujours satisfaisant. Voilà, je vais donc conclure en ce qui concerne les lecteurs normaux. A plus, j'espère.

OR

Le coin des matheux :

La méthode associée à la somme de Riemann est évidemment d'une simplicité enfantine (le lecteur normal n'est pas censé lire, donc pas censé non plus rouspéter devant ce genre de remarque même s'il n'a rien compris avant), puisqu'elle constitue la véritable définition de l'intégrale telle qu'elle est désormais définie dans les programmes du premier cycle universitaire (*NDLR:Il existe en plus un autre type d'integrale dit Integrale de Lebegue*), c'est à dire pour les fonctions réglées (et même les fonctions continues par morceux, m'enfin), qui sont, je le rappelle, limite uniforme de fonctions en escalier.

La majoration de l'erreur associée à cette première méthode est donnée par la formule de Taylor avec reste intégral appliquée à l'ordre 1 qui donne :

Int(a,b,f(t),t) = f(a)*(b-a) + Int(a,b,f'(t)*(b-t),t)

Le reste intégral est majoré par le formule des accroissements finis je crois, en tout cas par f'(c)*(b-a)^2 /2. Quand on fait le découpage sur les n intervalles, la largeur (b-a) est remplacée par (b-a)/n, d'où l'apparition de n^2 au dénominateur. Malheureusement, il en reste un au numérateur quand on effectue la sommation, l'erreur commise est donc bien en 1/n. Plus précisément, si M1 est un majorant en valeur absolu de la dérivée de f sur l'intervalle (a,b), on a : Erreur < M1 * (b-a)^2 / (2*n)

La méthode de Simpson est quand même plus élaborée théoriquement, même si son principe (cf interpolation) reste assez simple. La seule démonstration que j'en ai vu jusqu'à présent consistait à appliquer 4 fois le théorème de Rolle à une fonction 'judicieusement choisie'; ce qui n'est pas à mon avis la méthode qui à permis de trouver la formule. Si quelqu'un pouvait me fournir des détails sur ce sujet (et sur Runge-Kutta aussi d'ailleurs)...

La majoration de l'erreur doit s'effectuer comme précédemment mais elle est cette fois-ci en 1/n^4, c'est :

Erreur < M4 * (b-a)^5 / (2880 * n^4)

où M4 est un majorant de la dérivée quatrième de f. (D'ailleurs, f doit ètre C4 dans ma démo, ce qui doit ètre loin d'une hypothèse minimale, m'enfin).

Un dernier mot concernant cette méthode. D'abord elle ne reste efficace que pour des fonctions continues, et mème bien continues puisque la classe doit ètre assez élevée si on veut vraiment rester puriste, or, il y a des domaines où le calcul intégral doit sortir des fonctions usuelles, et où mème la théorie des fonctions réglées reste insuffisantes, tout comme la définition de Riemann. On est alors obligé d'introduire l'intégrale de Lebesgue. (Tout ça, c'est de l'épate, je n'en sais pas plus que vous, mais j'adore en jeter...)

Ensuite, si vous envisagez de l'utiliser pour un calcul d'intégrale généralisée, méfiance, ça peut marcher, mais ça peut aussi ne pas marcher; tout dépend surtout de M4... à vous d'avoir du pif pour intuiter la gueule de la dérivée quatrième. Disons surtout que si la machine trouve comme vous, le résultat est juste, sinon.... dieu seul le sait (avec le prof de math évidemment). Dans ce cas, il vaut mieux se abattre sur un développement en série entière, ou sur un bon vieux calcul pourrin.

Allez, j'arrète le massacre, ciao.

LES RESEAUX, CES INCONNUS

Ethernet, Novell, etc... Autant de noms esoteriques que nous decortique Azebulon

LES RESEAUX ou LA COMMUNICATION ENTRE ORDINATEURS

Branche, cable : la communication est à l'ordre du jour. Malheureusement, nos beaux ordinateurs n'aiment guère communiquer, et le jour ou l'on pourra raccorder une machine au telephone aussi simplement qu'on branche aujourd'hui une prise de courant ou un moniteur video ne semble pas encore arrivé...

Supposons que l'equipement terminal de traitement de donnees de l'agence A d'une compagnie X souhaite echanger des informations avec le terminal de l'agence B.

Supposons egalement que, par des conventions communes dans cette compagnie, les mèmes informations sont presentées et representees de la mème maniere dans les deux agences. Equipées de surcroit du mème modèle du mème ordinateur par exemple de deux APPLE IIGS. Supposons en bref que tout le monde parle rigoureusement la meme langue. Voici la maniere la plus "classique" de s'y prendre. Il faut d'abord trouver un lien materiel pour relier les deux machines. Pour cela, nous utiliserons le reseau telephonique, qui n'est certe pas parfait, mais possede l'avantage essentiel de l'universalité. Il n'a pas ete concu pour l'acheminement des donnee d'informations codées telles qu'elles circulent où sont stockées dans les ordinateurs (*Bien qu'à ma connaissance, Montpellier soit entièrement cablée en fibre optique. Mais ça reste à verifier NDLR*). Les lignes telephoniques sont faites pour véhiculer du son et plus precisement la voix humaine (bande passante 300 a 3400 Hz). Il est donc nécessaire de transformer les signaux "tout ou rien" des machines informatiques en d'autres signaux qui puissent passer par les mèmes lignes qu'une conversation entre des gens.

LES MODULATEURS - DEMODULATEUR (MODEMS):

Tels quels, les deux ordinateurs sont incapables de s'en servir, leurs signaux sont (electriquement) tres faibles et binaires: le plus couramment, c'est une tension légèrement supérieure a 2 volts qui représente le "1", une tension inférieure a 0,8 volt, le "0". Et tout le

reste n'est que combinaison de "0" et de "1". Il va falloir introduire, entre la ligne et ces machines, des boites noires assurant, d'une manière ou d'une autre, la transformation des signaux d'ordinateurs en signaux d'allure "sonore" et vice versa. Ces équipements assurent la modulation et la démodulation du signal; d'ou l'appellation de MODEMS. La modulation consiste en une sorte de gazouillis: le sifflet change de ton quand change le bit de donnees presente par l'ordinateur. La démodulation est réalisée par filtrage; c'est ainsi que fonctionne le moins onéreux des modems nommés coupleurs acoustiques. Les modems de haut de gamme se raccordent directement aux fils de la ligne telephonique, et utilisent des procédés plus fins jouant sur la phase ou sur l'amplitude de l'onde sonore injectée dans la ligne. Personnellement je dispose d'une carte APPLETELL en slot 6 de mon GS (rassurez vous, offerte généreusement par ROLAND MORENO lui meme lors d'une rencontre a Montpellier).

NTERFACES:

A l'interieur d'un ordinateur, on peut multiplier les interconnexions entre éléments: elles sont courtes et relativement faciles a faire grace aux circuits imprimés ou intégrés. Les BUS acheminent couramment des groupes de 8, 16 ou 32 bits en parallele. Quand aux liaisons de l'unite centrale avec les périphériques proches, elles s'effectuent le plus souvent en serie. La norme la plus connue est la RS232C très proche de la V24. La gestion du protocole de cette dernière exige tout de mème un conneteur a 25 broches.

PROTOCOLES DE TELECOMMUNICATIONS:

Ligne transmission, mode de transmission, modems, formats, trames, procédures constituent un ensemble de contraintes strictes importantes et complexes dont l'énoncé constitue le protocole de télécommunication. Schématiquement, le système (simplifié) décrit précédemment comporte outre la ligne de transmission : deux modems, une "couche" d'équipements et de logiciels, qui a mission d'échanger des éléments d'informations simples (bits) avec des modems dont elle assure en outre la commande; une seconde "couche" formée de programmes qui respectent avec les programmes correspondants de l'autre bout, une procédure de liaison soigneusement determinée; enfin des programmes que nous considérerons comme "les applications", c'est a dire qui commencent a se préoccuper de la nature des informations echangées.

RESEAUX LOCAUX: GENERALITES

Un réseau local relie physiquement des stations de travail et leurs périphériques a l'intérieur d'un site géographiquement limité. Ces stations ont pour but de partager entre plusieurs

utilisateurs des ressources communes: grosses unités de disques avec programmes et fichiers, imprimantes, traitements lourds. Le traitement est effectué par chaque station de travail. Un tel réseau doit permettre de : préserver le parc existant de stations de travail (mini, micro etc...); d'éviter les conversions de procédures et de logiciels; d'acceuillir les services télématiques et bureautiques; d'accéder aux réseaux publics: TRANSPAC, TELECEOM 1, RNIS, X400...; d'accéder aux autres reseaux locaux, notamment industriels (pilotages d'atelier, d'automates,de commandes numériques ...); d'integrer le traitement de données diverses: numériques, alphabétiques, images, voix...

Certaines de ces caractéristiques sont encore aujourdh'ui peu répandues. Elles dessinent des evolutions probables.

TOPOLOGIE DES RESEAUX LOCAUX:

Principales topologies rencontrées: Le Bus, L'Anneau ou Boucle,l' Etoile, l'arbre; d'autres structures notamment de reseaux mailles se rencontrent beaucoup plus rarement. Les societes XEROX, DIGITAL EQUIPMENT et INTEL ont mis au point le reseau type ETHERNET qui est l'un des modeles le plus repandu (*Et aussi un des plus 'fragiles', le système utilisé étant une arborescence : Il suffit qu'une station tombe en panne pour que tout ce qui trouve en dessous soit déconnecté du réseau - NDLR*).

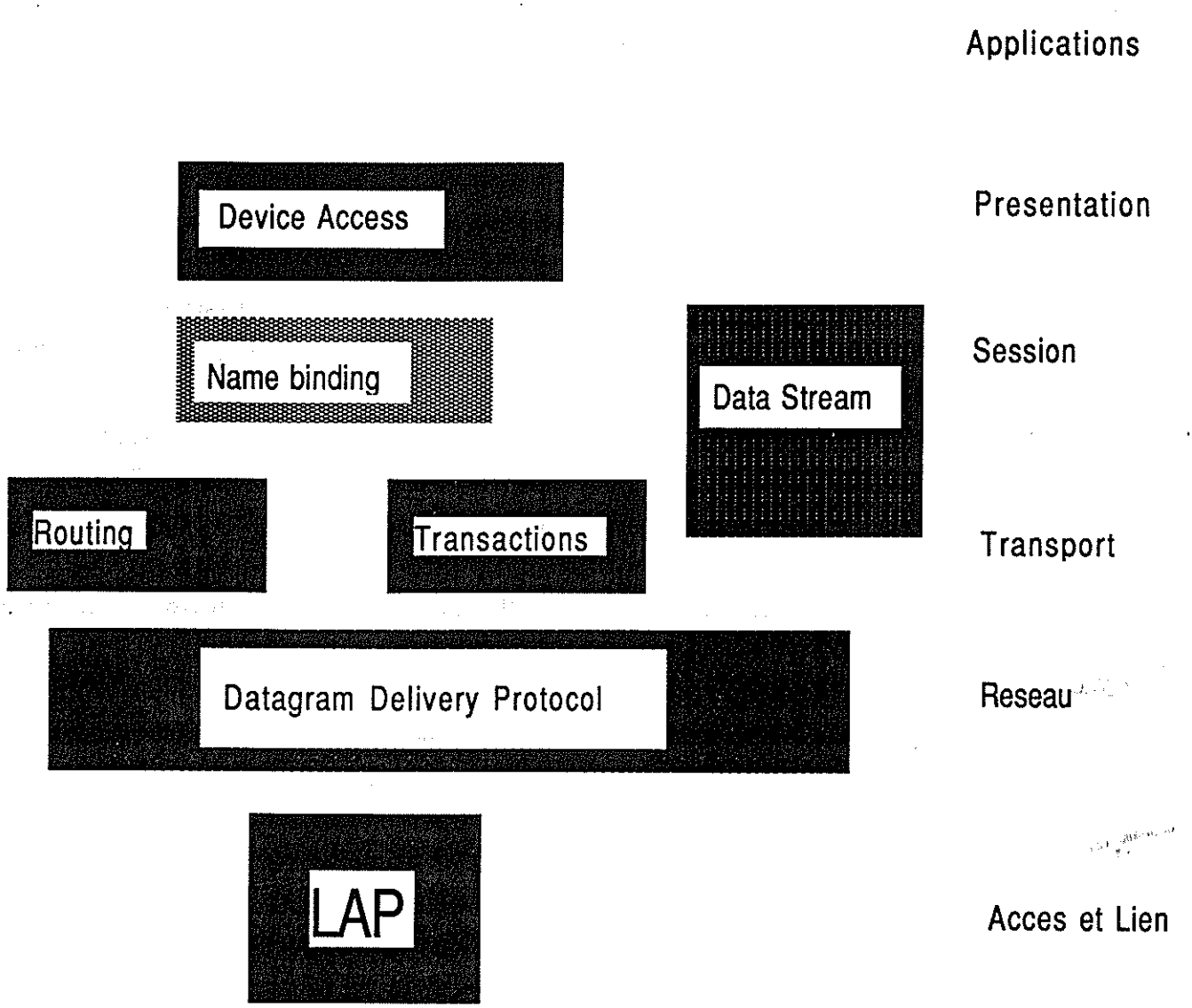
RESEAUX A BASES DE MAC OU DE IIGS:

Plusieurs configurations sont possibles. Le serveur APPLESHARE qui contient toutes les informations à partager. Le serveur 3COM qui dispose de son propre hard ou NOVELL qui demande toutefois un 386. Ces deux serveur sont compatibles APPLESHARE. Existe egalement le reseau TOPS dans lequel tout poste est à la fois serveur et "client". Ici comme je l'ai decrit précédemment, on peut relier ses APPLE à l'aide des liaisons numériques (merci numeris) et le logiciel EASYLINK par exemple. On peut également, si on désire un debit relativement important et régulier utiliser le systeme TRANSFIX; on a alors pas besoin de boitiers car fournis par les TELECOMS; on utilisera toujours EASYLINK par exemple. Le troisieme choix principal sera d'utiliser les liaisons X25 qui sont chères, peu pratiques mais toutefois réalisable a l'aide d'un soft genre SAMBRIDGE. Je vous déconseille d'utiliser ce type. Méfiance egalement lors de l'achat d'un modem; si certains, comme la carte Appletell par exemple, sont directement livrés avec un logiciel performant, il n'en est pas de même pour d'autres. Au sujet des softs, VERSION COM est le plus agreable a utiliser. Je preconise ce même soft lorsque l'on veut relier deux APPLE IIGS ou un IIGS avec une autre marque. Le second soft est TELPLUS qui a des ambitions limitées. En conclusion pour faire communiquer votre IIGS avec l'exterieur, le materiel le plus

adapte (mais pas le moins cher) est la carte APPLETELL asocié a VERSION COM. On trouve maintenant des cartes APPLETELL d'occasion (ex. MICRO'CASE a Monpellier) à près de 2500F.

LE RESAU APPLETALK : UN RESAU POUR TOUS

Pour communiquer entre ordinateurs, APPLE a mis au point son propre réseau. Votre IIGS peut ainsi parler a un autre IIGS ou à un MAC. APPLETALK est un réseau "ouvert" et ne necessite qu'un dispositif de connexion réduit aux cables et aux boitiers de séparation galvanique. Le fonctionnement du réseau est totalement assuré par un apport logiciel sur chaque machine, qui organise les informations destinées au réseau, sous forme de paquets. Transpac utilise egalement

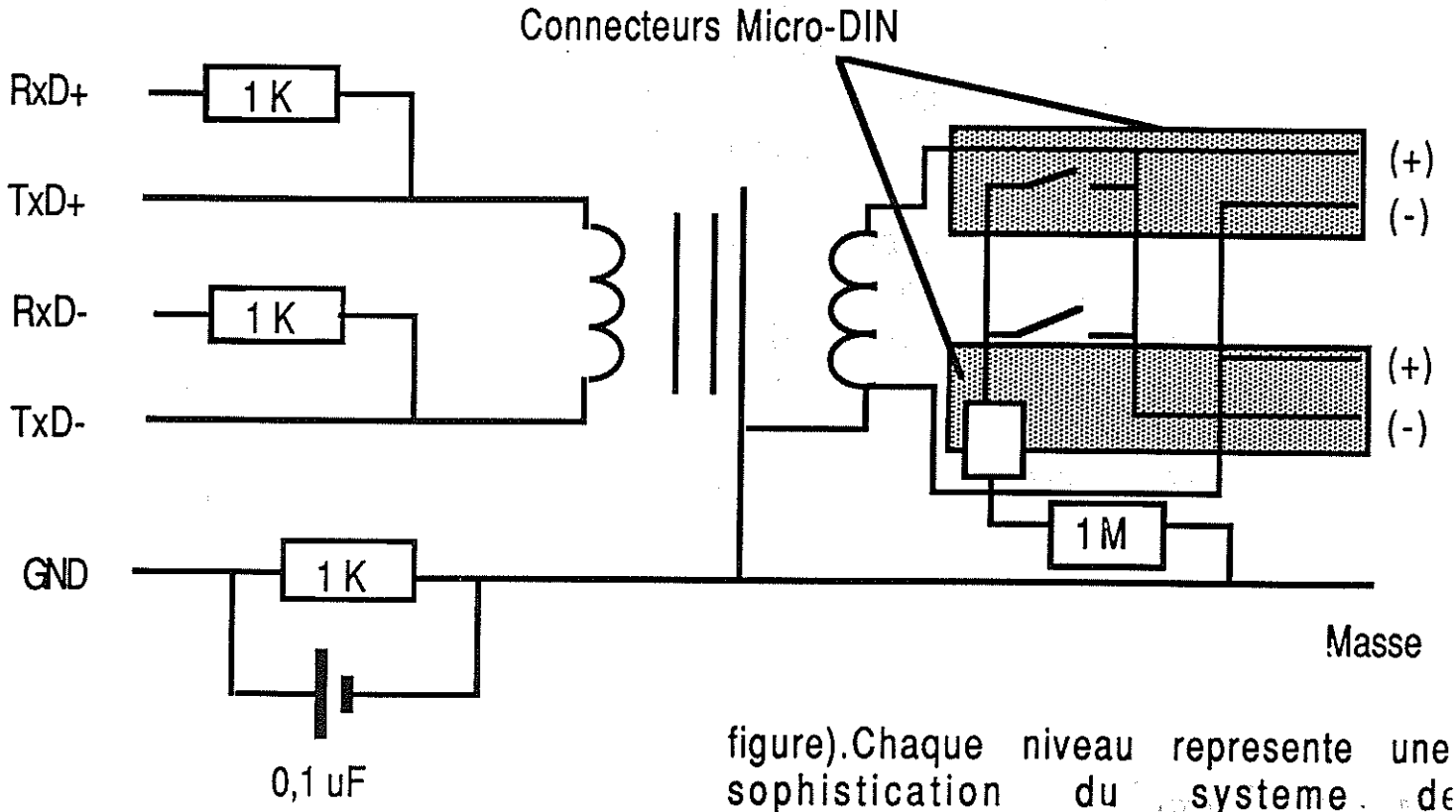


Couches Hard du Reseau

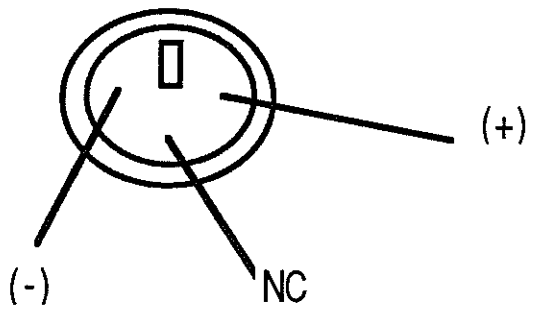
cette technique. Les caractéristiques techniques d'APPLETALK sont les suivantes: structure ouverte; 7 couches logicielles; 32 postes maxi; distance maxi de transfert: 300M; vitesse: 240Kbits soit 30000 carac/sec; protocole CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance); Format d'encodage SDLC; filaire torsade; Le reseau APPLETALK utilise une architecture aussi complexe que celle d'ETHERNET chez IBM. La difference se situe au niveau de la vitesse de transmission (230,4 KBauds contre 10MegaBauds pour ETHERNET).

L'architecture du reseau APPLETALK est divise en 7 niveaux de protocoles (voir figure).Chaque niveau represente une sophistication du systeme de transmission et peut contenir un ou plusieurs protocoles. Ces derniers utilisent generalement les protocoles des niveaux inferieurs. Chacun d'eux represente la definition d'une règle de transmission réalisée par une couche de logiciels présente dans les lecteurs de gestion du reseau.

D'un point de vue "hard", le réseau se presente sous la forme d'un boitier de



Connecteurs Micro-DIN



connexion contenant un translateur qui assure l'isolation galvanique des postes entre eux. Le plus important réside dans le MAC. Il s'agit du boitier de communication ZILOG SCC 8530 qui assure des fonctions aussi importantes que la reconnaissance du numéro de noeud, détection de porteuse etc... Au niveau physique, les données sont transmises sur un bus synchrone RS-422A en utilisant une modulation "FM0" qui consiste a representer un zéro par un changement d'état entre deux changements d'horloge et un "un" par une abscence de changement d'etat. Elles sont transmises sous forme de paquets de taille variable allant jusqu'a 600 octets. Ces paquets sont envoyes suivant le format HDLC/SDLC. Il convient de noter que ce systeme autorise l'insertion ou la suppression de postes sans causer de probleme electrique. Si APPLETALK est un reseau aussi complexe qu'ETHERNET, ceci est du au systeme d'allocation dynamique des numéros de noeuds et de prises. Cependant, ce dispositif complexe evite à l'utilisateur d'avoir a configurer son reseau. Si APPLETALK possede un logiciel interne aussi sophistiqué, c'est dans le but de simplifier la tache de l'utilisateur.

LA FIBRE OPTIQUE: SUPPORT DE L'AVENIR

En BUS, en ETOILE, en ANNEAU avec ou sans jeton, le réseau necessite un support de transmission. C'est un cable coaxial, une paire de cuivre torsadée ou une fibre optique. Chacun possede des avantages mais aussi des inconvenients. L'un est cher, l'autre n'est pas fiable ou ne supporte pas les hautes vitesses ou les grandes distances. La fibre optique, issue d'une technologie recente reste marginale. Elle revet un intérèt croissant par rapport au cable traditionnel. Le pricipe est de remplacer les conducteurs electriques classiques, c'est à dire remplacer le courant electrique modulé par un faisceau optique. Ce systeme permet entre autres une grande immunité aux bruits electriques electromagnetiques. La distance entre deux postes et egalement augmentée: ainsi, deux GS pourront etre distant de 5 Kilometres au lieu de 300 metres. Seul le prix est actuellement un obstacle. Voila, la place me manque afin de faire une analyse plus detaillee de tout ce qui est réseaux informatiques. Beaucoup de choses restent a dire notamment sur APPLETALK. J'espère, chers lecteurs, que cet article (un peu long il est vrai), ne vous a pas semble trop technique et par de la meme trop rébarbatif. AZEBULON

Pour tout contact ou info suplementaire ecrivez-moi dans ma BAL AZEBULON sur RTEL

La Trois Dimension Sur GS

Une legende racontée par Perfect Bugs

Lorsque Modulae est sorti (merci le FTA), tout le monde a été émerveillé par la fluidité de l'animation. Certains Amigamen (d'excellents amis au demeurant) ont été tout simplement dégouté par la vitesse de calcul : Le GS avec ses 2,8 MHz rejoignait allègrement Amiga et autres ST avec leurs 8 MHz, sans compter tous les coprocesseurs de l'Amiga. Ceci montre bien que comparer les vitesses brutes de deux machines n'a pas beaucoup de sens si les microprocesseurs sont différents, car les algorithmes utilisés sont pratiquement les mèmes (à l'exception du Fill mode utilisé pour quelques animations). Enfin bon, là je m'écarte de mon sujet. On va parler aujourd'hui des différentes techniques d'animation 3D en assembleur (seul langage permettant une vitesse suffisante).

Pour commencer, les principes de base: Pour faire une animation, cela se conçoit aisément, il faut pouvoir faire subir des transformations aux objets. Généralement, si ces objets sont solides (et on se limitera à ce cas), les seules transformations possibles sont des translations et des rotations. Pour une translation, pas de problème, il suffit d'ajouter aux coordonnées du point une certaine valeur dépendant du vecteur de translation.

Exemple : Le point de coordonnées (1,2,4) subit une translation de vecteur (2,1,3). Le point résultant est alors le point (3,3,7).

Jusque ici, pas de problème. En revanche, là où le problème se complique, c'est lorsque l'on veut faire tourner les objets dans l'espace. Autant vous dire que les rotations autour d'un axe quelconque de l'espace ne sont pas évidentes du tout, on les laissera donc pour le moment de coté. On va se limiter à des rotations autour des axes du repère. Pour cela, on va utiliser ce que l'on appelle des matrices de rotation, qui sont en fait des tableaux 3*3 (dans ce cas particulier). La raison pour laquelle on ne peut faire (de façon simple) tourner les objets autour d'une droite quelconque est que derrière ces matrices se cache tout un ensemble mathématique basé sur les espaces vectoriels, c'est à dire un espace uniquement composé de vecteurs, sans

origine. Donc toute transformation dans cet espace s'effectuera sur des vecteurs, alors que l'univers dans lequel nous vivons, chétifs mortels (mais qu'est ce que je délire moi ?), est un espace affine, c'est à dire comportant une origine. Enfin, bon bref : C'est pas simple... Rassurez vous, je ne vais pas détailler la théorie. Sachez simplement que pour obtenir la matrice ci dessous, il faut multiplier trois matrices 3*3 entre elles, ce qui est loin d'ètre évident.

Donc, étant donné trois angles alpha, beta et gamma, la matrice de rotation autour des trois axes suivant ces trois angles est de la forme (on notera CS1 pour Cos(alpha), CS2 pour Cos(beta), CS3 pour Cos(gamma), et SN1, SN2 et SN3 pour les sinus correspondants).

CS2 * CS3	CS2 * SN3	-SN2
CS3*SN1*SN2 - SN3*CS1	SN3*SN1*SN2 + CS1*CS3	SN1*CS2
CS1*SN2*CS3 - SN3*SN1	CS1*SN2*SN3 - SN1*CS3	CS1*CS2

On notera qu'avec alpha = beta = gamma = 0, on obtient des 0 partout sauf sur la diagonale où il y a des 1. On appelle cette matrice la matrice Identité, car elle laisse tous les points invariants. Une petite définition en passant, qui peut servir : L'inverse d'une matrice (quand il existe) est la matrice qui fait subir à tout point de l'espace une transformation inverse à celle de la matrice d'origine. Pas très clair ? Je m'explique. Imaginez un point M de l'espace, qui subit une transformation quelconque. On obtient un point transformé (qu'on appelera dorénavant image de M) qui, s'il subit la transformation inverse, redevient le point d'origine. Dans le cas général, c'est pas évident de calculer une telle matrice. Mais ici, comme on a affaire à une rotation, il est évident (enfin, presque) de voir que l'inverse correspond à une matrice où tous les angles sont opposés aux angles de départ, soit si on appelle P(alpha,beta,gamma) la matrice de rotation et P' son inverse : P' = P(-alpha,-beta,-gamma).

Reste maintenant à savoir comment utiliser cette matrice. Pour un point (soit trois coordonnées), on multiplie la matrice par le vecteur coordonnée correspondant. Mais attention, la multiplication de matrice n'est pas évidente. Voici ce que cela donne pour une matrice 3*3 :

```
! a   b   c ! !x!   ! ax + by + cz !
! d   e   f ! !y! = ! dx + ey + fz !
! g   h   i ! !z!   ! gx + hy + iz !
```

Si on pose x'=ax+by+cz, y'=dx+ey+fz, z'=gx+hy+iz, le point M'(x',y',z') est le point image de M(x,y,z) par la rotation. Comprendo ?

Ben voilà, on a les bases. Maintenant, il faut appliquer tout cela. L'apparition de cosinus et sinus dans la matrice oblige en théorie à utiliser des réels. En fait, on peut créer une table de cosinus et de sinus où chaque terme est multiplié par 16384, avec le bit 15 utilisé comme bit de signe. On se ramène ainsi à des entiers. Mais attention, il y a un problème lors de la multiplication : En effet, quand on multiplie deux sinus ou cosinus entre eux sous cette forme, on obtient un résultat sous la forme a*b*16384*16384 (ici a et b représentent le nombre réel, avant multiplication par 16384). Il faut donc diviser par 16384 tout résultat d'une multiplication de deux nombres de la table. Lorsque l'on a obtenu la matrice (rangée quelque part en mémoire, et qui est d'une longueur de 3*3*2=18 octets), il ne reste plus qu'à appliquer la formule de multiplication pour tous les points de l'objet pour obtenir l'image de l'objet par la rotation. Attention cependant, si vos points sont codés sur un octet, le résultat sera sur 24 bits, et si vos points sont codés sur deux octets, le résultat sera sur 32 bits. Il ne faudra pas oublier de diviser par 16384 (qui est une puissance de deux, ce qui simplifie grandement les choses. Attention toutefois au bit de signe lors des rotations et décalages à droite) pour obtenir le vrai résultat. Une routine est donnée que calcule la matrice et transforme les points (sans toutefois diviser le résultat par 16384, au cas où vous auriez d'autres choses inavouables à faire subir à ce malheureux objet). Normalement, elle marche...

Encore faut il afficher tous ces objets. Pour la 3D en fil de fer, pas de problème, il suffit d'adopter un codage de l'objet par segment, d'appliquer la rotation, de projeter sur l'écran (on verra ça à la fin) et d'afficher les segments. Mais pour la 3D face pleine... Alors là, accrochez vous.

En fait, avec notre machine, il existe deux grandes méthodes, ayant chacune un tronc commun. De toutes façons, à la base, il faut adopter un codage de l'objet qui non seulement distingue les segments, mais aussi les faces. Le plus simple est à mon avis de coder tout ça sous forme de liste de polygones, chaque polygone étant composé de segments. Une autre contrainte dont on verra la raison
plus loin impose que les points d'un polygone soient codés dans un sens précis, c'est à dire soit dans le sens trigonométrique, soit dans l'autre sens, et ceci en considérant le polygone vu de l'exterieur de l'objet (voir schéma). Mais une fois ce sens choisi, il doit être le même pour tous les objets. Ceci permettra plus tard d'éliminer automatiquement les faces orientées de telles façons qu'elles soient invisibles par l'observateur. Chaque point (pour un objet non baroque) est donc répété plusieurs fois : Un certain nombre de fois pour chaque face (tout dépend du nombre de face en ce point) et

deux fois dans chaque codage de façe (Pas très clair tout ça...).

Une fois ceci fait, on procède de la même manière que pour la fil de fer, mais on s'arrète une fois la transformation effectuée.

-- Interlude --

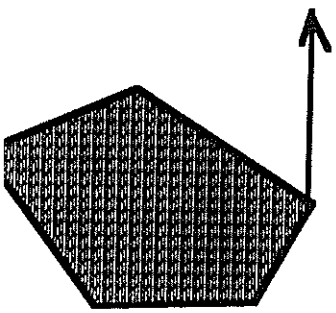
Il est maintenant nécessaire d'introduire deux notions mathématiques, à savoir le produit vectoriel et le produit scalaire.

Si on considère deux vecteurs a et b, le produit vectoriel ab est un vecteur normal (orthogonal) au plan défini par les deux vecteurs a et b. La formule de c = ab en coordonnées cartésiennes est la suivante :

```
      !x1!      !x2!              !y1*z2-y2*z1!
a = !y1!  b = !y2!     c = ab = !z1*x2-z2*x1!
      !z1!      !z2!              !x1*y2-x2*y1!
```

En ce qui concerne le produit scalaire, noté a.b, on a la formule suivante : a.b = x1*x2 + y1*y2 + z1*z2 (il faut noter que le résultat est un nombre réel).

-- Fin de l'interlude --



Cette fleche represente le produit vectoriel des deux vecteurs consecutifs de la face

Reprenons. Prenons deux segments consécutifs dans le codage d'une face. Il est facile de transformer chaque segment en vecteur (chaque coordonnée du vecteur étant la différence de la coordonnée correspondante du deuxième point et du premier point). En faisant le produit vectoriel, on obtient un vecteur normal au polygone (ou la surface, c'est comme vous voulez), et pointant vers l'exterieur de l'objet (ou l'interieur, selon l'orientation des faces choisie au départ). Bien, maintenant, il suffit de faire le produit scalaire de ce vecteur par le vecteur correspondant à la direction d'observation. On obtient un nombre réel qui est positif si les deux vecteurs sont dans la même direction et négatif s'ils sont opposés (encore une fois, le signe dépend de l'orientation de départ). On peut ainsi facilement marquer une façe comme étant invisible pour l'observateur (si on a choisi une orientation telle que le produit vectoriel pointe vers l'exterieur de l'objet, la façe est invisible si ce vecteur et le vecteur d'observation sont de même sens, soit un produit scalaire positif - Voir schéma -). Mais attention, cette étape ne supprime que les façes totalement invisibles à cause d'une orientation. On peut s'arreter là si l'objet est convexe (c'est à dire que pour deux points quelconques de l'interieur de l'objet, le segment reliant ces deux

points est entièrement dans l'objet. Une sphère est convexe, un anneau n'est pas convexe). Il ne reste plus alors qu'à projeter puis afficher les faces marquées comme visibles. Mais si l'objet n'est pas convexe, d'autres étapes sont nécessaires. Je vais maintenant distinguer deux méthodes.

Dans la pratique, cette face n'est pas dessinee

Dans la pratique, Dans la pratique n'est pas n'est pas

Etape 1

On redessine par dessus : les parties cachees disparaissent

On redessine parties cachees

Etape 2

L'algorithme du peintre

Voici un nom bizarre pour une méthode particulièrement simple qui est utilisée en particulier par les Amiga, leur coprocesseur y étant particulièrement bien adapté. En fait, elle consiste à afficher les faces en commençant par celles les plus éloignées de l'observateur. Ainsi, les parties cachées sont automatiquement supprimées par le remplissage des faces qui justement les cachent (oulà ! Pas très français tout ça. Enfin bon...). Elle nécessite en revanche une routine de remplissage de polygone ne tenant pas compte du fond. Pour classer les faces, un simple tri à bulle sur les cotes (coordonnées suivant l'axe de profondeur, c'est à dire généralement Oz) maximales (ou minimales, selon l'orientation de l'axe) de chaque point de la face. Et après, il ne reste plus qu'à projeter et à afficher selon la méthode décrite plus haut. Un conseil : Faites votre affichage dans le banc 01 avec l'autoshadowing coupé, puis enclenchez l'autoshadowing et faites un rafraichissement de l'écran. Cette méthode à tout de même deux inconvénients majeurs : Sa relative lenteur pour les objets simples et surtout, due cette fois à la structure du GS, l'impossibilité de faire une animation plein écran fluide à cause du balayage (du moins dans l'état actuel des choses). En revanche, elle a deux avantages énormes : D'abord, elle permet une gestion des sprites (Je pense d'ailleurs que c'est la méthode utilisée par le FTA pour le partie 'Free Flight' de Modulae, car avec toutes ces étoiles, je ne vois pas trop comment ils peuvent faire en utilisant le mode Fill. Mais je peux me tromper. Si c'est le cas, j'aimerais beaucoup que le FTA me le dise et m'explique la méthode utilisée). D'autre part, pour les objets composés de nombreuses petites faces, elle est très efficace, car le temps est proportionnel au nombre de faces, tandis que la méthode utilisant le mode Fill (tout au moins celle que j'ai developpé) prend un temps qui augmente de façon exponentielle suivant le nombre de faces.

Utilisation du mode Fill

Alors là, on s'accroche, parce que ce qui précède, c'est rien du tout à coté de ce bordel (surtout à expliquer !!). Comme on l'a vu plus haut, un polygone est constitué de segments : Le problème se ramène donc à trouver les parties du segment qui sont visibles et celles qui sont invisibles. Pour bien expliquer tout ça, je vais mettre en place une méthode qui est à proscrire. On verra plus loin comment l'améliorer. Bien, maintenant, on considère un segment particulier d'une face particuliere. Ce segment n'est visible que si il est devant toutes les façes. On peut pour simplifier considérer qu'il est visible à au moins une de ses extrémités (cette hypothèse est très restrictive). En associant un flag à ce segment décrivant sa position (devant ou derrière) par rapport à toutes les façes, on peut facilement tester s'il est visible ou non : En effet, ce segment ne peut etre rendu visible ou invisible qu'a l'intersection (en projection) avec un autre segment. Il suffit donc de considerer dans l'ordre tous les pixels du segment, de tester les éventuelles intersections et de mettre à jour le flag (si le segment était derrière une façe et recoupe à nouveau un segment de cette façe, il devient alors à nouveau visible - pour cette façe !!). Je sais, c'est pas très clair, mais étudier le schéma ci-contre et tout s'éclaircira. Il est bien évident qu'une telle méthode consomme énormément de temps machine. Il est indispensable de l'améliorer. Pour ce faire, au lieu de considérer indépendamment chaque segment de l'objet, on va considérer ce que dans certains bouquins, on appelle une 'Scan Line' (Rien à voir avec le SCB). On prend la ligne la plus haute de l'objet. Sur cette ligne, il y a un certains nombre de sommets d'aretes : On transfère toutes ces aretes dans une table qui regroupe en fait les aretes dites actives. Puis on descend d'une ligne, et on vérifie si une ou plusieurs des aretes de la table active se terminent, auquel cas on les évacue. On vérifie de mème si d'autres aretes sont susceptibles d'entrer dans la table (car commencant à la ligne considérée). Une fois ceci fait, on sait que pour une ligne donnée, il n'y a qu'un nombre limité de segments actifs, lesquels se trouvant tous dans la table : Il suffit de comparer deux à deux leur cotes et de mettre à jour les flags en conséquence (S'il y a N segments dans la table, cela fait au maximum N/2 tests. En effet, si un segment est derrière un autre, cet autre segment est naturellement devant le premier !). En considérant les parties visibles de chaque segment, on obtient toute une série de segments qui, s'ils sont dessinés, donnent une image en fil de fer de l'objet. Reste à mettre de la couleur. En utilisant le mode Fill, c'est relativement (!) simple : Il suffit d'affecter un autre flag à chaque segment déterminant la couleur. Comme on a choisi une orientation fixe pour le codage des

segments, on sait si le segment est à gauche de la façe ou à droite (Mettons que l'on ait choisi le sens des aiguilles d'une montre comme orientation. La façe est tournée vers l'observateur - c'est forcément le cas, car on a déja supprimé plus haut les autres façes - Si le segment descend, cela signifie qu'elle est à droite. Si il monte, cela signifie qu'il est à gauche). S'il est à gacuhe, le flag prend la couleur de la façe, et s'il est à droite, il prend la couleur de la dernière façe coupée. Il faudra quand mème générer toute une série de segments une fois tous les calculs terminés qui remplaceront les segments de l'extrème droite par des segments de la couleur du fond. Et voilà, en activant le mode Fill et en dessinant nos segments, on a l'objet en trois dimension façes pleines. Si de plus on a pris la précaution de sauvegarder la table des segments de l'image précédente, on obtient une animation très fluide, car il suffit juste d'effacer les anciennes lignes puis de redessiner les nouvelles, ce qui supprime la nécessité de faire des STZ sur toute la page écran.

Compléments

Et oui, on a presque fini. Mais je vais aborder (juste aborder) ici trois autres petites choses : Primo, les rotations autour d'un axe quelconque de l'espace. Secundo, un algorithme de remplissage de polygones convexes rapide. Et tertio, comment faire une routine de ligne rapide.

En ce qui concerne les rotations, je serai bref. Le principe est assez simple la réalisation l'est moins (pourquo croyez vous que le FTA ne l'ait pas fait ?) Dans la théorie, il faut faire ur changement de repère, c'est à dire ramene l'axe de rotation à un des axes du repère d'origine. Pour ce faire, on prend un poin quelconque de l'axe qu'on ramène à l'origine par une translation : On fai subir ce sort à tous les points de l'obje Puis, par une rotation idoine, on fai correspondre l'axe de rotation avec mettons, l'axe Oz du repère. Enfin, e appliquant une matrice de rotatio correspondant à une rotation d'angl gamma autour de Oz (alpha=beta=0), pui en appliquant à l'objet le transformations inverses (une rotatior plus une translation), on obtient l'image de l'objet. Pas simple, mais joli à voir.

Pour le remplissage, je me limite à des polygones convexes. Je pars du plu haut sommet du polygone, et je trace une une les lignes horizontales (voir schéma). Or tracer des ligne horizontales est particulierement simple Il suffit de mettre quelque part e mémoire une série de

```
STA 2000,X
STA 2000,X+2
...
STA 2000,X+80
```

En donnat une valeur idoine à X et e

démarrant à un endroit particulier de la liste, on obtient ainsi ce que l'on désire.

Et enfin, la routine de ligne : Alors là, c'est très simple. Il suffit de générer une table à deux entrées, qui donne l'incrément en nombre de pixels horizontaux et verticaux en fonction de la variation en X et en Y.
Ainsi, si on donne deux points, on calcule DeltaX et DeltaY, on regarde dans la table, et il suffit de tracer: Facile, non ?

Voilà, c'étati tout pour cette fois. Si vous avez des problèmes (et il y en aura), n'hésitez pas à me contacter.

A LA QUESTION : Y AURA T'IL UN PROCHAIN NUMERO

JE REPONDS : OUI, PEUT ETRE. MAIS POUR CELA, NOUS AVONS BESOIN DE REDACTEURS, DE GRAPHISTES, ETC...

FAIRE UN MAGAZINE A QUATRE N'EST PAS PARTICULIEREMENT EVIDENT, ET DE PLUS, AU FUR ET A MESURE QUE L'ECHEANCE DU BAC (POUR CERTAINS) ET DES CONCOURS (POUR D'AUTRES) APPROCHE, LE TEMPS VA MANQUER DE FACON DRASTIQUE.

DONC SI VOUS AVEZ UNE IDEE : N'HESITEZ PAS, CONTACTEZ NOUS

Hypocrisie

Un petit coucou !! ou Sequence Hypocrisie !!

Yo , bonjour ou bonsoir je passe longuement dans ce canard pour, genre, le remplir d'un delire delirement absurde et logique pour la gloire des fideles professionnels de SEGA, MINTENDO et L'AMIGA, (ces MacGAMERs, (c) William COMTE)...

Heu, c'est une fausse frayeur. Enfin la vrai realite de mon passage dans ce canard (longue vie, Peace) est :

Un petit neckneck (?) a tous ceux que j'aime que j'aimerai, qui m'ont aidaient a concretisser le premier numero de La POMME Illustree. Je vais me devertir (car l'informatique c'est ludique !) a les repertoirier :

Ceux qui ont repondus aux nombreux messages de propagande sur minitel:
(Par ordre chronologique):
RJP / DR. MADDOX / WEAKY / GSMANIAC / HERVE 52 / CRACKOS / DEDE GS / OR / KANGOUROU / CHTULHU NOW / ARTHUR / ARAGON GS / GRAND SOT / KLAW / ZARDOS / BUGABOO / SHE / LORD OF PRODOS / BEEBOP / JULES / JO / BLAST / ZORG / JIHAILDE / AZEBULON / SANGUINE / QUARKS / PLANO / ELECTRON / BASTIEN** / TYM (BRAINSTORM) / PI / KRYPTON / ARN / MINI 13 / ALBO / EHEH-GS / MARCOVIL / TPS / RICKY LA BRICOL / HULK / AL / BOOSKOP / CANA / BELOU / JAD / FW / DOM-GS / SNOOZE / GS2 / PAD / LUS / GS Lorraine Club / Hyper Pomme Paris / ISILDUR / JCB&P.39 / HYPNOSS / GRAND SOT / ZIBOR / DAG MENTAR / KWISARTZ / MAB / DAX / APPLE GANG / HELICOSOFT / DOUME ... (
Quant on aime La POMME Illustree on ne compte pas (De l'humour , Nul!!)
Certains attendent toujours leurs numeros comme Gs Lorraine Club (20 exemplaires). X'cusez moi.

Les redacteurs:
Perfect Bugs, dans le role du pseudo-Manager deborde.
Bandit II, dans le role du 'J'ai pas le temps !'.
Ferox, dans le role du tres-sauvage tres-anarchiste tres-faignant.
Nibble,
Les Photocopieurs:
Bozo, le genereux Mega Merci
Criss, le courageux

Sur-encheres :
Un pote hyper-Pommien Parisien ayant apporte les 1er numeros a Sausheim.
Tout les message de paix et d'encouragement (Weaky, OR, Brainstorm, Azebulon...).

'oila je crois en avoir fini avec hypocrisie... je vais peut-etre exploker /hy I quitte the APple Illustree: ien si tu veux mec j'ai fais le numero 0 en semaines tout compris (du papier blanc la distribution a Shausheim), avec le ıultiscribe Apple 2e et aucune onnaissance sur la PAO, alors impression olonnes par colonnes, puis collage, /pexage... une fois les articles reunis. ion, jusque la ca va, (mais bientot ca 'iras pas, j'connais l'histoire), ce loisir ı'a permis de voir des papillons 3-D a elices clignotantes pendant les cours et ɜs controles, mes profs n'aimant pas ma oesie mathematiciennes et physiciennes . Ils m'ont dedicaces des 3/20 ! comme je e tiens pas a faire une 2eme edition de remiere S, je vais p'etre bosser (joie de ıa mere prefere !).
Voila c'est tout, Bye (apres les edicaces)

Cette Article est dedicace a :
lathalie (), Maryelle Fabien, Christelle :hristophe, Virginie, Anne-Sophie, ;eraldine, Carinne, Gerome, Ludo, Sylvain, lachel Jean-Guillaume, Les spaghettis a a bolognaises du re

LISTE NON EXHAUSTIVE ...

Liste des erreurs du GS

Eppluche

Liste non exhaustive ...

65816

Dans la table qui donne la liste des instructions en hexadecimale par ordre croissant, il manque l'intervalle 3B a 4F.

DRIVE

Les erreurs sur le 5 1/4 et le prodos 8 ne sont pas interessantes.

3.5.1 Le lecteur 5" 1/4 /* pour rappel seulement */

Phase 0	$C0E0	Off	$C0E1	On
Phase 1	$C0E2	Off	$C0E3	On
Phase 2	$C0E4	Off	$C0E5	On
Phase 3	$C0E6	Off	$C0E7	On

Selection du Drive 1 $C0EA
Selection du Drive 2 $C0EB
Mise en route du moteur $C0E9
Arret du moteur $C0E8

Q6	$C0EC	Off	$C0ED	On
Q7	$C0EE	Off	$C0EF	On

3.5.2 Le lecteur 3" 1/2

Dans cette partie on va parler uniquement du lecteur non-intelligent l'Apple 3.5. En effet celui-ci est controlable directement par le GS, alors que l'Unidisk 3.5 dispose de son propre processeur (65C02) et de sa propre RAM/ROM.

Le drive 3.5 utilise les memes commutateurs logiciels que le drive 5 1/4. On doit donc le selectionner avant toute chose car le 5 1/4 est valide par defaut.

Pour selectionner le Drive 3.5 on doit effectuer les operations suivantes:

* Valider les slots internes en 5 et 6 du GS

```
LDA  $C02D
AND  #$9F
STA  $C02D
```

* Etre en vitesse rapide, et ne pas repasser en vitesse lente au demarrage du moteur:

```
LDA  $C036
AND  #%11111011  ;Pas de passage en
```

AND #%11111011 ;Pas de passage en
itesse lente
;sur mise en marche du
ıoteur
ORA #$80
STA $C036

Ecrire un $40 en $C031 pour selectionner
ɜ 3.5, tete 0.

.5.2.2 Allumage extinction du moteur

$C0E9	Allumage du moteur
$C0E8	Extinction du moteur

.5.2.3 Selection de drive 1 ou 2

C0EA Selectionne le Drive 1
C0EB Selectionne le Drive 2

.5.3 Registres propres au 3.5

.e registre d'interface disque: $C031

'ermet de selectionner le lecteur 5 1/4
ıu le drive 3 1/2, permet pour
:e dernier de selectionner me tete
ıdressee pour les operations de
ecture ecriture.

3it 7 (SEL) a 1 selectionne la tete 1 /*
;EL n'etait pas decrit !!! */
a 0 selectionne la tete 0

3it 6 a 1 selectionne les drives 3 1/2
a 0 selectionne les drives 5 1/4

.e lecteur 3 1/2 utilise aussi les memes commutateurs logiciels que le lecteur 5 1/4. Ces commutateurs situes de $C0E0 a C0EF portent les memes noms que lors de leur emploi pour le 5 1/4 mais par contre leur fonction est totalement differente.

En effet les commutateurs Q6 a Q7 de concert avec la mise en marche du moteur permettent d'acceder a cinq registres differents.

Q7	Q6	Moteur	Operation
Off	Off	On	lecture registre de donnee
Off	On	xx	lecture registre de status
On	Off	xx	lecture registre handshake
On	On	Off	ecriture registre de mode
On	On	On	ecriture registre de donnee

Apres selection d'un registre, l'ecriture a n'importe quelle adresse impaire de l'IWM ($C0E1 a $C0EF) ecrira dans ce registre, la lecture d'une adresse paire de l'IWM ($C0E0 a COEE) lira ce registre.

Registre de Mode:

Attention le lecteur doit etre arrete et non seulement eteint.

/* la signification des bits 4, 2 et 1 est

verse dans le texte original */
l'erreur vient en fait du 'HARDWARE
ANUAL REFERENCE' premiere edition */
d'ou proviennent ces informations top
ecretes... */

Bit 7 Reserve ne pas modifier
Bit 6-5 Reserve ecrire toujours 0
Bit 4 Vitesse de l'horloge de
cture
1:8 Mhz
0:7 Mhz valeur pour le Gs
Bit 3 1:Les cellules de bits de
octet font 2 Usec cas des
lecteurs 3.5
0: Les cellules bits de l'octet
nt une longueur de 4
Usec, valeur
utilisee pour les devices
onnectes au Smartport et
les lecteurs 5 1/4.
Bit 2 si a 0, le lecteur apres sa
eselection, restera allume
une seconde.
Bit 1 a 1 protocole de Handsake
synchrone (autres cas)
a 0 protocole de Handsake
ynchrone
(lecteur 5 1/4).
Bit 0 1:Mode latch valide, la
onnee reste valide pour la
duree d'un octet. (cellule de 2
Isec alors 16 Usec,
cellule de 4 microsec alors 32
Isec)
0: Mode latch invalide, la
onnee lue
reste valide pendant 7 Usec.

Le registre de Status:

Bit 7 test de la protection
ecriture, (lecteur 5 1/4) resultat
acces registre pour le 3 1/2.
Bit 6 Reserve
Bit 5 1:le lecteur 1 ou 2 est
selectionne et le moteur est
allume.
0:pas de lecteur selectionne
Bit 4-0 Idem registre de mode.

Le registre de Handshake:

Bit 7 1 le registre de donnees est
pret pour les donnees.
0 le registre de donnees est
plein.
Bit 6 1 le dernier octet ecrit a ete
ecrit correctement.
0 un octet a ete rate et non
ecrit sur le disque.
Bit 5-0 Reserve.

Le registre de donnee:

Suivant l'etat des commutateurs Q7 Q6 on
accede a l'octet lu a partir
du disque, ou on ecrit un octet sur le
disque.

Un deuxieme jeu de registres peut etre
accede grace aux commutateurs
correspondant aux phases, et a la
selection de la tete. (SEL)
Lecture de ces registres:

* le texte original parle des phases 3, 2
t 1 a positionner */

)7 doit etre Off et Q6 On le lecteur doit
tre valide avec le moteur
llume. Puis verifier que Phase 3 est Off.
)n positionne ensuite les
hase 2,1,0 et SEL suivant le registre
uquel on veut acceder. Une fois
eci effectue on peut lire l'information
lans le bit de poids fort de
C0EE. Apres lecture du registre on peut
cceder a un autre registre
implement en repositionnant les phases
t SEL. A la fin des operations
le lectures de registres, repassez en
node normal en accedant a Q6 off.

* le tableau qui suit est completement
lecale dans le texte original */

Phase2	Phase1	Phase0	SEL	Registre	
Off	Off	Off	0	DIRTN	direction ete
Off	Off	Off	1	CSTIN	Presence l'une disquette
Off	Off	On	0	STEP	
Off	Off	On	1	WRTPRT	protection criture
Off	On	Off	0	MOTORON	moteur en narche
Off	On	Off	1	TK0	tete sur iste 0
Off	On	On	1	TACH	tachometre
On	Off	Off	0	RDDATA0	flot donnees tete 0
On	Off	Off	1	RDDATA1	flot donnees tete 1
On	On	Off	0	SIDES	lecteur simple ou double face
On	On	On	1	DRVIN	lecteur installe

Ecriture dans ces registres:

/* encore des erreurs dans les numeros des phases */

Verifiez d'abord que Phase3 est off, ensuite passez On phase0 et Phase1, puis positionnez SEL a 0. Mettez ensuite Phase1 et Phase0 dans l'etat necessaire pour acceder aux registre, et positionnez Phase2 suivant la valeur a ecrire (On pour 1, Off pour 0).
Maintenez Phase3 a On, pendant aux moins 1 Usec, mais moins de 1 msec (sauf si vous ejectez une disquette) puis remettez le a Off. Soyez sur que vous ne changez ni Phase1-2 ou SEL pendant que Phase3 est On, et que Phase0 et Phase1 sont On avant de mofifier SEL.

NOTE:Comme dans le cas d'une lecture le lecteur doit etre d'abord selectionne.

Phase1	Phase0	SEL	Registre	
Off	Off	0	DIRTN	Sens de

deplacement de la tete
Off On 0 STEP Deplace la tete d'une piste
On Off 0 MOTORON allume le moteur
On On 0 EJECT ejecte le disque.

DIRTN: indique le sens de deplacement de la tete a 1 on se deplace vers la piste 0, a 0 vers la piste 79.

CSTIN: A comme valeur 0 quand une disquette se trouve dans le lecteur.

STEP: Mettre ce registre a 0 cause un deplacement de la tete en fonction de DIRTN. Quand le deplacement est termine (a peu pres 12 msec) le lecteur remet STEP a 1 et on peut recommencer.

WRPTR: a 0 si le disque est protege contre l'ecriture.

MOTORON: 0 allume le moteur 1 l'eteind. Ne fonctionne que si le drive est selectionne et qu'une disquette est presente.

TK0: a 0 si la tete se trouve sur la piste 0.

EJECT: ecrire 1 dans ce registre ejecte la disquette. Attention il faut maintenir Phase3 On pendant au moins 1/2 seconde.

RDDATA0:
RDDATA1: donne le flot de bits instantane venant de la tete 1 ou 0.

SIDES: 1 car lecteur double faces.

DRVIN: 0 si un lecteur est connecte.

Exemple d'acces aux lecteur:

Ejection d'une disquette lecteur 1

```
LDA  $C036
AND  #$FB
ORA  #$80
STA  $C036

LDA  $C02D
AND  #$9F
STA  $C02D

LDA  #$40
STA  $C031
```

;Selection du drive

```
LDA  $C0EA
LDA  $C0E9
 JSR  WAIT      ;Attente
;
LDA  $C0E6     ;Phase3 Off
LDA  $C0E1     ;Phase0 On
LDA  $C0E3     ;Phase1 On
```

```
    LDA  #$40
    STA  $C031      ;SEL a 0

Parametres pour l'ejection

    LDA  $C0E1      ;Phase0 On
    LDA  $C0E3      ;Phase1 On
    LDA  $C0E5      ;Phase2 On (valeur 1)

    LDA  $C0E7      ;Phase3 On
    JSR  WAIT       ;attente d'une demi-sec
    LDA  $C0E6      ;Phase3 Off

Deplacement du bras

SEEK
Allume le moteur et se deplace vers une
piste
Cyl :Piste voulue
CurCYl:piste actuelle
Drive:Lecteur concerne

* on suppose que les initialisations ont
deja ete faites */
* 3"5 selectionne, moteur en marche, etc
*/

SEEK  LDX  Drive      ; 0 pour drive 1 et 1
pour drive 2
      BIT  CurCyl,X   ; doit contenir une
valeur >= $80 a l'initialisation
      BPL  no

      JSR  Recall     ;Recalibre
      BCS  erreur
no    JSR  DriveOK

      SEC
      LDX  Drive      ;Calcul le nbr de
      LDA  CurCyl,X   ;pistes a deplacer
      SBC  Cyl
      BEQ  fin

      LDY  #$01       ;Piste plus petite
      BCS  inf
      LDY  #$00       ;No pst plus grand
      EOR  #$FF
      ADC  #$01

inf
      TAX
      TYA
      JSR  EcrREG     ;Sens de deplacement
      JSR seek_n

fin   LDX  Drive
      LDA  Cyl
      STA  CurCyl,X
      CLC
      RTS

erreur
      LDA  #$02       ; Cette facon de traiter
l'erreur vient de la ROM
      ORA  Error
      STA  Error
      RTS

seek_n
      LDA  #$04       ;Registre SEEK
      JSR  EcrREG

Loop  JSR  Lit2REG    ; une grosse erreur
!!!
      BPL  Loop       ;attente fin de seek
      DEX
```

```
     BNE  seek_n      ;encore une piste
    LDX  #$3C

tmp  DEX
    BNE  tmp
    RTS
DriveOk

     LDA  #$00        ; code inutile !!!
    STA  $6A
    LDA  #$5D
    STA  $6B

    LDA  #$08      ;MOTORON
     JSR  LitReg
    BPL  Ok

    JSR  Ecr2Reg

     LDX  Drive
     LDA  $11,X       ; mettre au depart a $19
!!!
    JSR  WAIT

    LDA  #$19
    STA  $11,X
    JSR  WAIT

Ok   RTS


Wait
    PHA
     JSR  Wait2
    PLA

Wait2
    STA  $18
```