

VOYAGEZ AU CENTRE DE L'APPLE,
DANS LE ROYAUME DE L'ASSEMBLEUR...
VIVEZ DES AVENTURES GALACTIQUES
DANS GRUDS IN SPACE...

A LES AVENTURIERS DU BOUT DU MONDE

LA REVUE DU MICROCAM



JUIL/AOUT 1985. N° 08

29

MR PERDEVAULT JEAN FRANCOIS
RECHERCHES DEV.

le 0

Ce mois ci, nous vous avons mittoné un numéro double pour cause de vacances. C'est une pratique très répandue chez la plupart de nos confrères, qui s'explique par un ralentissement de l'activité informatique chez les constructeurs, et par le départ en vacances des lecteurs et des journalistes; ce dernier point ne nous concernant pas beaucoup; nous travaillons pour vous pendant que vous brunissez sur les plages, petits veinards !

Puisque précédemment, je faisais allusion à la presse informatique, restons dans ce domaine très particulier.

Certains magazines sont résolument rivés sur un standard ou sur une marque, et puisqu'ils le crient bien fort, en ont fait un argument de vente, de la sorte, que cela ne surprend plus personne.

D'autres essayent, avec plus ou moins de bonheur, de consacrer une part égale à chaque ordinateur. En fait l'information ne peut rester que très générale du fait de la complexité du marché; ce sont les vrais "mordus" de la "micro informatique en général" qui constituent la clientèle de cette presse. N'espérez pas y trouver des trucs, des peeks ou des pokes pour faire marcher votre traitement de texte avec l'imprimante Tartempion!

Vous qui désirez épater vos amis en faisant un exposé à brûle pourpoint sur n'importe quelle imprimante Thaïlandaise, ou sur le dernier né de la gamme Apple, lisez ces magazines qui feront de vous des petits "cablés" infos!!

YR CORNIL.

LES AVENTURIERS DU BOUT DU MONDE

MICROCAM. CRCAM
19 rue du Pré Perché
35040 RENNES

Directeur de la publication : Yves Roger CORNIL

Rédacteur en chef : David CORNIL

Grands reporters :

Arnaud LADAGNOUS : ESCAPE FROM RUNGISTAN
Jacqueline VALENTIN : MYSTERY HOUSE
Patrick LECONTE : EXCALIBUR QUEST

David CORNIL : GRUDS IN SPACE
Yann CHEUREL : LUCIFER'S REALM
Bertrand MORIN : MASQUERADE

Infos club Microcam : Jean-François PERCEVAULT et Jacqueline VALENTIN

INTRODUCTION A L'ASSEMBLEUR/3

Partie:01

L'assembleur n'est pas un langage pour les "pros" contrairement à ce que vous pouvez penser. Ce premier article a pour but de vous apprendre à utiliser un assembleur et à en comprendre les principes de base.

Les prochains articles porteront sur les sujets suivants :

- Les registres.
- Les boucles et les tests.
- Les instructions de branchement.
- Des instructions complémentaires afin de faire de vous un champion de l'assembleur.

I) Différence entre le code objet et le code source.

Lorsque vous tapez en Basic le petit programme suivant vous le faites en mode Editeur. Lorsque vous demandez son exécution vous vous placez donc en mode Execution.

Programme 1

```
10 LET X=10
20 GOTO 20
```

En assembleur les choses se passent à peu près de la même façon. Lorsque vous tapez votre programme vous le faites en Mode Editeur. Voici en assembleur le même programme que précédemment.

Programme 2

```
LOOP   LDX     £$0A    ;0A = 10 en hexa
        JMP     LOOP
```

Ce programme n'est pas directement exécutable . En effet il faut d'abord le compiler pour le rendre compréhensible par votre Apple. Pour cela il faut utiliser un assembleur qui va vous servir à taper le programme 2. Il sera appelé Code Source. Une fois le code Source (Prog 2) compilé vous pourrez alors l'exécuter et voir le fruit de votre travail.

4/INTRODUCTION A L'ASSEMBLEUR

Voici un schéma de la programmation Assembleur.

```
-----  
- Tapez le code -  
- Source (prog 2 -  
-----  
==  
==  
-----  
- Assemblez le code -  
- Source. -  
-----  
==  
==  
-----  
- Exécuter le code - Le code assemblé sera appelé  
- Assemblé. - Code objet.  
-----
```

Maintenant que vous connaissez la différence entre un code source et un code objet, dirigez vous vers votre Apple ou compatible et Bootez la disquette Big Mac Assembleur de Glen Bredon.(Publicité gratuite!!)

Au bout de quelques secondes un menu va s'afficher sur votre écran.

```
(E) DITOR  
(S) AVE SOURCE  
(L) OAD SOURCE  
(Q) UIT  
(O) SAVE OBJECT CODE  
(A) PPEND
```

Command -->

La commande E est utilisée pour taper le code Source.
La commande S est utilisée pour sauver le code Source.
La commande L est utilisée pour charger le code Source.
La commande Q est utilisée pour quitter le Big Mac.
La commande O est utilisée pour sauver le code Objet.

Je crois que vous n'aurez aucun mal à comprendre la commande Q (...).

Bien alors maintenant tapez E.

Un prompt symbolisé par un : apparaîtra dans le coin gauche de l'écran.

Tapez maintenant ADD et Return pour rentrer dans l'éditeur de Code Source.

l' apparaît sur l'écran.

Tapez maintenant : <SPACE> ORG <SPACE> \$300 <Return>
 LOOP <SPACE> JSR <SPACE> \$FBDD <Return>
 <SPACE> JMP <SPACE> LOOP <Return>

Et Return 2 fois.

INTRODUCTION A L'ASSEMBLEUR/5

Le prompt doit réapparaître à l'écran. Tapez L(ist) pour la liste de votre programme. En théorie vous devez obtenir ceci :

```
      ORG      $300
LOOP  JSR      $FBDD
      JMP      LOOP
```

Mais votre Source n'est toujours pas exécutable : Il faut le compiler. Tapez ASM et Return. A la question Update Source répondez N. Le listing suivant va défiler.

Programme 3

```
0300:      ORG      $300
0300:20 DD FB          LOOP  JSR      $FBDD
0303:4C 00 03         JMP      LOOP
```

Et voilà, votre programme Source est transformé en code objet. Mais manque de pot il n'existe pas de commande Run en Assembleur. Pourtant votre code Objet est dans la mémoire de l'Apple mais pas à son adresse correcte. Vous devez donc sauver ce code sur disquette et ensuite "Bruner" ce fichier.

Vous avez vu dans le menu la commande O (sauvegarde du code Objet). Pour y retourner tapez O et Return.

Insérez votre disquette formatée dans le lecteur 1 et tapez O.

Le Big Mac vous demandera alors Object :

Tapez par exemple MICROCAM et Return. La disquette se met à ronronner et quelques instants plus tard le menu réapparaît.

Tapez C comme Catalog. Le catalog de votre disquette apparaît. Vous devez y voir MICROCAM. Tapez maintenant BRUN MICROCAM et écoutez...

Vous venez de réaliser votre premier programme Assembleur.

Bravo!!!!

Tapez Control Reset pour retrouver l'assembleur.

Avant de continuer à vous expliquer les autres commandes du Big Mac, analysons le programme que vous venez de taper.

Pour cela retournez dans l'éditeur en tapant E.

Le prompt (:) apparaît. Tapez L(ist) pour voir la liste du programme 3.

```
      ORG      $300
LOOP  JSR      $FBDD
      JMP      LOOP
```

La première colonne est réservée aux labels : ce sont eux qui vous permettront de boucler vos programmes. En effet les numéros de lignes que vous voyez sur le côté ne servent A RIEN dans la programmation Assembleur mais servent à éditer votre programme.

C'est dans les 2ièmes et 3ièmes colonnes que vous taperez vos instructions proprement dites.

6/INTRODUCTION A L'ASSEMBLEUR

----- Analyse du programme 3 -----

ORG \$300 Correspond à l'adresse d'implantation de votre programme. Celle-ci varie si votre programme utilise des graphiques, du basic, etc...

\$300 est utilisée pour des programmes de faible longueur. \$ veut dire que l'on travaille en mode Hexadécimal.

LOOP JSR \$FBDD JSR est l'équivalent Assembleur de Gosub. \$FBDD est l'adresse de la routine qui fait le Bip Bip (Control G). Les gosub ne portent pas sur des numéros de lignes mais sur des adresses mémoires ou sur des sous routines assembleurs.

JMP LOOP JMP est l'équivalent de GOTO. Même conditions que pour JSR.

Si l'on disposait d'un Basic sans numéros de lignes voici ce que l'on obtiendrait :

```
LOOP PRINT"Control G"  
GOTO LOOP
```

Tableau récapitulatif

Commandes Basic	--	Commandes Assembleurs	--
-- GOTO + Num de ligne		-- JMP + Label	--
-- GOSUB + Num de ligne		-- JSR + Label	--
-- Equivalent Basic		-- Adresse de la routine	--
-- Control G		-- \$FBDD	--

Si vous avez bien suivi mes instructions vous devez être en mode Editeur.

Reprenons notre programme. Tapez L et Return.

Le Big Mac affichera :

```
1            ORG     $300  
2        LOOP    JSR     $FBDD  
3            JMP     LOOP
```

INTRODUCTION A L'ASSEMBLEUR/7

Nous voulons effacer l'écran et afficher le titre Apple et ensuite continuer avec le Bip Bip du programme 3.

JSR \$FB60 efface l'écran et affiche le titre Apple, il suffit donc de l'insérer juste avant LOOP JSR \$FBDD.

Pour cela tapez I2 et return.
Le Big Mac vous affichera une ligne 2 toute vide : tapez maintenant <SPACE> JSR <SPACE> \$FB60 et return 2 fois.

Tapez L(ist) pour voir votre nouvelle liste modifiée.
Vous devez obtenir :

```
          ORG      $300
          JSR      $FB60
LOOP     JSR      $FBDD
          JMP      LOOP
```

Tapez ASM pour assembler votre programme.
En théorie le Big Mac doit afficher Errors: 000.

Revenez ensuite au menu en tapant Q.

Le menu doit réapparaître.

Votre disquette contenant MICROCAM doit être dans le lecteur 1
Avant de sauver ce code objet, je pense qu'il vaudrait mieux faire une copie de sauvegarde du texte source. Tapez pour cela S et tapez un nom, par exemple S1 comme Source 1.

Pour pouvoir exécuter votre code objet vous devez le sauver d'abord sur disquette en faisant O comme précédemment. Tapez un nom pour votre programme objet, par exemple M1.

Pour pouvoir l'exécuter faites comme pour le fichier MICROCAM.
Je remontre les opérations pour ceux qui auraient déjà oublié.

Tapez C comme catalog et ensuite Brun M1.

Si vous ne vous êtes pas trompé vous devez voir APPLE en haut de l'écran et entendre le Bip Bip discontinu.

Une nouvelle fois bravo, vous avez réussi votre second programme assembleur et appris comment insérer une ligne dans votre code source.

Bien, maintenant que vous savez assembler et insérer une ligne je vais vous apprendre à "deleter" une ligne de code source, ce qui me paraît logique. Pour cela retournez dans l'assembleur en tapant Control Reset. Le menu doit réapparaître. Tapez E pour rentrer dans le mode éditeur et tapez L pour voir le listing de votre programme. Vous devez obtenir ceci :

```
1          ORG      $300
2          JSR      $FB60
3  LOOP    JSR      $FBDD
4          JMP      LOOP
```

8/INTRODUCTION A L'ASSEMBLEUR

Je veux supprimer le bruit discontinu du control G. Comme vous le savez JSR \$FBDD est le synonyme de Ctrl G en assembleur. Il faut donc supprimer la ligne 3 et remplacer le JSR \$FBDD par une instruction NOP signifiant de ne rien faire.

Pour insérer une ligne vous aviez tapé I, donc en toute logique pour détruire une ligne il faut taper D. Tapez D3, Return et L (Return) pour voir votre programme.

```
1      ORG      $300
2      JSR      $FB60
3      JMP      LOOP
```

La ligne LOOP JSR \$FBDD à bien disparue. Mais vous devez la remplacer par une autre à cause du JMP LOOP.

Tapez maintenant I3 et Return.

Tapez ensuite LOOP <Space> NOP (Return).

Tapez L pour voir le nouveau listing ainsi formé. Si vous avez bien suivi mes conseils vous devez obtenir :

```
1      ORG      $300
2      JSR      $FB60
3 LOOP  NOP
4      JMP      LOOP
```

Essayons vite notre programme. Assemblez le et exécutez le. Une dernière fois je remontre les opérations détaillées pour les distraits.

ASM et Return

```
Q  -----
O  -----
TOTO -----
C  -----
BRUN TOTO ---
```

En théorie vous devez obtenir le titre Apple II au dessus de l'écran.

Sinon vous devez avoir fait une erreur quelque part. Cherchez bien!!

Voilà, ce premier article touche à sa fin, mais ne vous jetez pas par les fenêtres je vous retrouve le mois prochain. Vous pouvez peut être penser que vous n'avez pas appris grand'chose mais c'est complètement faux. Si vous avez bien compris toutes les manipulations que je vous ai montrées vous n'aurez aucun mal à comprendre les cours suivants. L'idéal serait que vous vous soyez procuré la doc du Big Mac pour le mois prochain cela me permettrait d'éviter un autre cours sur le fonctionnement de cet assembleur et de passer plus directement dans le vif du sujet, c'est à dire la programmation assembleur en elle même.

Machinalement vôtre,

Xavier Pouyollon.

LA CHUTE DES CHIFFRES ROMAINS

2ème PARTIE: 9 + 6 = F
=====

En pratique l'ordinateur comprend un certain nombre de symboles (alphabet, chiffres, signes spéciaux (+, -, *, / (<' % & \$ £ " !) < ...)) grâce à une table de correspondance appelée table des CODES A.S.C.I.I (American Standard Code for Information Interchange).

En effet, bien que seuls 2 états soient repérables par l'appareil (symbolisés par 0 et 1 d'où l'utilisation du système BINAIRE) les quelques 127 caractères nécessaires pour communiquer avec l'Apple ont été codés sous forme de 7 positions binaires + 1 position de contrôle.

Ainsi, à chaque caractère utilisé dans le langage courant est associé une valeur numérique normalisée.

Au caractère 'A' du clavier est associé le nombre binaire '01000001'. Devant la lourdeur du binaire et pour coïncider avec la notion d'octet (unité d'information de 8 caractères binaires (Binary digIT ou BIT)) le code A.S.C.I.I. retient un système de numération particulier : le SYSTEME HEXADECIMAL. Ce dernier permet de représenter un octet par 2 chiffres.

En effet, avec 8 positions et 2 états, il est possible d'avoir 2^8 possibilités différentes allant de :

```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 0 1 0
0 0 0 0 0 0 1 1
à -----
1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1
```

soit 256 possibilités.

Avec 4 positions et 2 éléments, le nombre de possibilités est de $2^4 = 16$. Des lors n'importe quel octet est représentable par 2 chiffres HEXADECIMAUX, c'est à dire en base 16.

MAIS COMMENT COMPTER SUR 16 DOIGTS ???

Comme nous ne connaissons que 10 symboles numériques (0 à 9), il est nécessaire d'inventer d'autres signes pour compter jusqu'à 16 avec un seul élément. Par définition, en BASE 16, les éléments sont : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

La correspondance entre les 3 bases connues est la suivante :

systeme --->	binaire	décimal	hexadécimal
BASE --->	2	10	16
	0	0	0
	1	1	1
	10	2	2
	11	3	3
	100	4	4
	101	5	5
	110	6	6
	111	7	7
	1000	8	8
	1001	9	9
	1010	10	A
	1011	11	B
	1100	12	C
	1101	13	D
	1110	14	E
	1111	15	F

Dans un tel système le caractère 'A' du clavier codifié 65 en décimal et 01000001 en binaire correspond à 41 en hexadécimal :

0100 0001
4 1

Pour visualiser la codification hexadécimale dont les nombres sont par convention précédés du signe \$ dans les ouvrages, faites les manipulations suivantes :

```

$ CALL -151 (RETURN)      appel du langage machine
(moniteur)

* 300          (RETURN)    liste du contenu de l'adresse $300
(300 en hexadécimal)

*              (RETURN)    la touche 'RETURN' permet
d'afficher le contenu des adresses suivantes

* 3D0G        (RETURN)    retour a l'adresse où se situe le
basic APPLESOFT

$              prompt du basic APPLESOFT

```

Les codes A.S.C.I.I. peuvent être visualisés facilement grâce à l'ordre BASIC CHR\$() qui donne le caractère A.S.C.I.I de la valeur numérique fournie entre parenthèses.

Exemple : PRINT CHR\$(65) affiche la lettre 'A'
correspondant au code A.S.C.I.I 65 en decimal

A l'inverse, l'ordre basic ASC(" ") donne le code decimal A.S.C.I.I. correspondant au premier caractère de la chaîne mentionnée entre guillemets.

Exemple : PRINT ASC("ALORS") affiche 65 (code
décimal de 'A').

A vos programmes pour lister l'ensemble des codes ASCII de 0 à 127 ... et pour établir des systèmes de numération différents !

Vous vous trouvez dans la salle de navigation d'un superbe vaisseau intergalactique, à votre première commande frappée au clavier l'ordinateur vous annonce que vous avez reçu un message dans la salle des communications. Vous quittez la salle de navigation en allant vers l'ouest. Vous vous retrouvez dans une salle communicant avec la salle des transmissions radio au sud, la salle des ordinateurs au nord (dont l'utilité n'a pas encore été démontrée !), et bien entendu la salle de navigation.

<S> Vous remarquez un écran à côté duquel se trouve un bouton vert. Votre curiosité naturelle vous oblige à l'enfoncer : push green button. C'est un message du commandement général basé sur la terre qui requiert de toute urgence votre aide. Voilà résumé ce message : la flotte galactique basée sur le front des Baranocks a épuisé son fuel et seul un cargo basé sur Pluton est à même de ravitailler votre armée; le fuel (l'héliotropanithe) se trouve sur Saturne. Le haut commandement pense donc que vous êtes le seul commandant capable de transporter le fuel de Saturne à Pluton dans un temps qui permettrait d'éviter la défaite de nos forces. Bien entendu vous pouvez refuser la mission (et aller vous coucher) mais votre sentiment patriotique, votre amour de la liberté et bla -bla-bla, vous pousse à accepter. D'autant plus que la perspective de gagner un million de dollars en cas de succès n'est pas pour vous déplaire. L'ordinateur vous donne les coordonnées de navigation jusqu'à Saturne : 64-18-52, alors que les coordonnées du teleport jusqu'à la mine sont : 77-34-40.

L'ordre de mission précise néanmoins que vous pourriez avoir certaines difficultés à obtenir du fuel puisque des mineurs se sont rebellés récemment... Il n'y a pas que sur terre que la C.g.t, comprenez Confédération des Gruds au Travail, fait des ravages !!

<N D D N> Vous vous trouvez dans le magasin, et déjouez donc le premier piège de ce jeu en ne prenant rien ! Tout est inutile. Rendez vous dans la salle auxiliaire de navigation.

<S W> Sur le clavier, entrez les coordonnées de Saturne : set 64-18-52. L'ordinateur annonce à sa façon qu'il s'engage en direction de la planète. Allez à l'ouest, dans la salle du téléporteur. Entrez au clavier les coordonnées de la mine : set 77-34-40. L'ordinateur annonce ensuite qu'il vient de se positionner en orbite autour de saturne. Entrez dans le teleport : go teleport. Les miracles de la technique vous transportent jusqu'à saturne; c'est maintenant que commence la véritable aventure, une aventure dans un pays inconnu et révolté...

SATURNE

<N N N N> Vous vous trouvez près d'une entrée de grotte, en face de vous, deux gardes, dont l'un possède une corde. Vous vous emparez par la force : steal rope. En commétant cet acte indigne d'un aventurier, vous avez déclenché la colère des gruds. Fuyez.

<S W> Devant une entrée de la mine, un gruds à l'air indifférent surveille l'entrée; à ses pieds vous ramassez une pièce de monnaie : get coin. Vous pourriez acquérir divers objets du magasin (sur la planète), mais il s'agit là encore d'un piège, ne compromettez pas vos chances en dilapidant votre argent. Je vous suggère de faire une autre utilisation de la pièce; suivez moi.

<E S S E S E> Vous vous trouvez devant la demeure de Lord Deebo. Votre première pensée se fixe sur la porte. Pas la peine de l'ouvrir, de la défoncer (voyou!) ou de passer par la fenêtre, contentez vous de vous comporter en gentleman, frappez : knock door. Le maître d'hôtel vous ouvre, vous répondez par un oui à sa question, et donnez la pièce : give coin. Le maître d'hôtel vous invite à rentrer.

<E> Vous vous trouvez face à Lord Deebo qui réclame son tribut, vous tapez : give tribute. Ce qui correspond en fait à une formule de politesse puisqu'il le refuse aimablement. Il vous demande : "Avez vous un vaisseau spatial ?" Vous répondez par l'affirmative. Il vous invite à le suivre dans le living room.

<E> Après un essai infructueux de votre part à engager la conversation Lord Deebo commence à parler. Il vous affirme qu'il peut vous aider à trouver le fuel que vous cherchez (mais comment a-t-il deviné ?); que cela ne le dérange pas de travailler avec des humains (comment croyez-vous qu'il soit devenu si riche ?). Pour l'argent il ferait n'importe quoi, même traiter avec le arler. Tout ce que vous avez à faire est de prendre cette note et de la porter à son partenaire, Mr Green, sur Venus. Les coordonnées de navigation sont : 96-17-14; les coordonnées pour le teleport sont : 82-13-64. Ce fameux partenaire doit vous remettre un "money maker" comme celui qu'il pointe du doigt mais qui n'est qu'une contrefaçon. Lorsque vous serez revenu il vous aidera à trouver votre fuel. En signe de confiance il vous confie un secret important : "la clé de l'un des murs de passage est dans l'une des caves inusitées." Il vous remet une lampe électrique et une note, que vous prenez : get flashlight, get note.

<W W W N W S> Vous pénétrez à l'intérieur du teleport pour retourner à votre vaisseau : go teleport. Dans la salle auxiliaire de navigation, vous rentrez les coordonnées de Vénus : set 96-17-14. L'ordinateur vous indique qu'il se dirige vers Vénus et vous signale qu'il y a un vaisseau non identifié au 66-12-15; nous reparlerons de ce vaisseau... Dans la salle du teleport, vous rentrez les coordonnées : set 82-13-64. Vous rentrez dans le teleport, et partez à la découverte de Vénus.

VENUS

<N> Vous remarquez de l'autre côté de la rivière un panneau sur lequel est écrit KURTZ. En fait, ce panneau n'a aucune utilité, si ce n'est de vous faire perdre du temps, ne cherchez pas à traverser la rivière et revenez plutôt sur vos pas.

<S E> La proximité des marécages vous incite à attacher votre corde au premier arbre venu : tie rope to tree.

<E D> Vous ramassez un pistolet dans la vase : get gun; et vous vous empressiez de remonter.

<U W> Vous détachez votre corde : untie rope; et vous la ramassez : get rope.

<N E N> Vous vous trouvez devant un venusien (drole de bête, n'est ce pas) qui peut être dangereux si l'on ne sait pas s'y prendre. La manoeuvre consiste à l'effrayer, pour cela vous tapez : shoot venusian. Le gros tas de graisse s'endort; vous pouvez passer...

<N W> Mr Green vous accueille avec un 357 magnum, et parait peu enclin à bavarder autour d'une tasse de thé, vous décidez donc de revenir en arriere et de déposer votre pistolet: e, drop gun, w. Vous remettez la note à Mr Green : give note; celui ci vous demande de prendre le money maker, ce que vous faites : get money maker.

<E> Get gun.

<S> Fine venusian.

<S W S W> Vous voilà revenu au teleport, que vous empruntez immédiatement : go teleport. La manoeuvre maintenant habituelle consiste à rentrer les coordonnées de Saturne, et du teleport. Rendez vous sur Saturne..

SATURNE

<N E S E> Vous frappez : knock door; et vous entrez en continuant à l'est. Vous offrez le money maker à Lord Deebo : give money maker. Deebo ne semble pas très fier de vous, puisqu'il vous rapporte des mauvaises nouvelles : "Mr Green m'a dit que vous êtes venu fouiner sur sa planète, il dit que vous avez pris l'un de ses revolvers." Deebo est désolé, mais un tel comportement ne l'invite pas à partager un secret avec vous. Vous vous débrouillerez donc seul. Deebo vous remercie tout de même en mettant en marche sa machine qui crache un petit caillou, d'une inestimable valeur; vous le prenez : get rock; et partez.

<? ? ?> (Attention, un point d'interrogation n'équivaut pas à un déplacement !!!). A l'aide du plan, sinon ça sert à quoi que je me décarcasse, vous vous rendez dans la grotte où vous trouverez des objets bien utiles pour poursuivre votre périple...

A suivre...

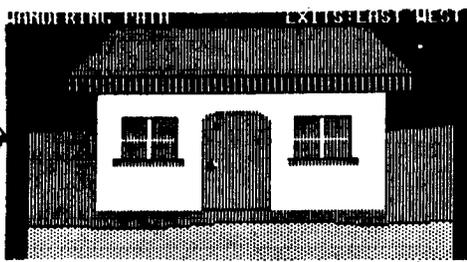
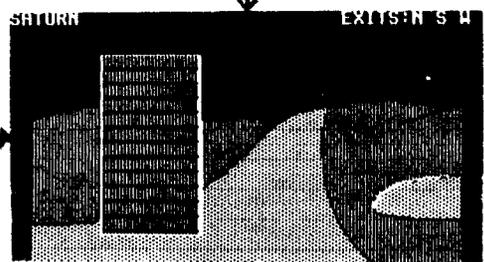
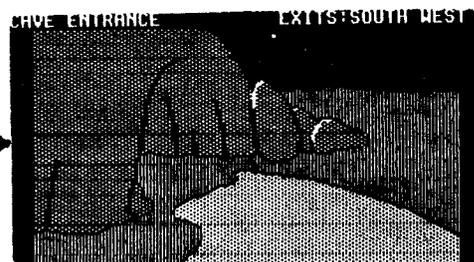
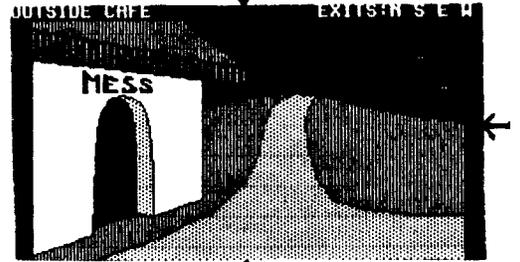
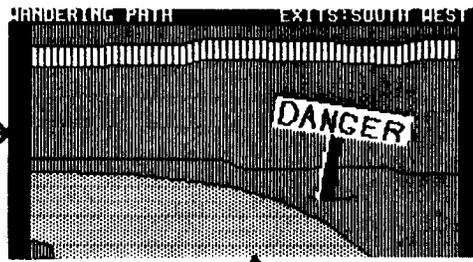
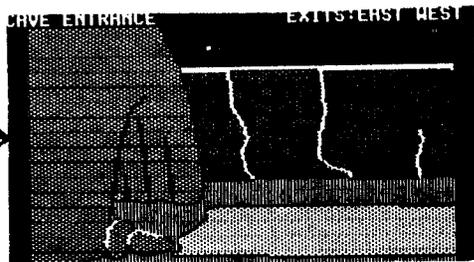
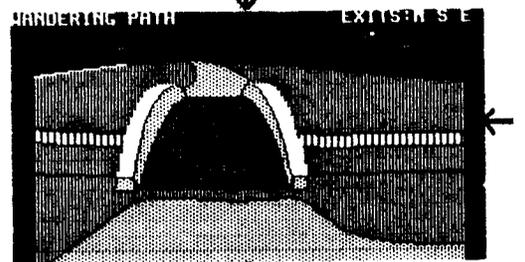
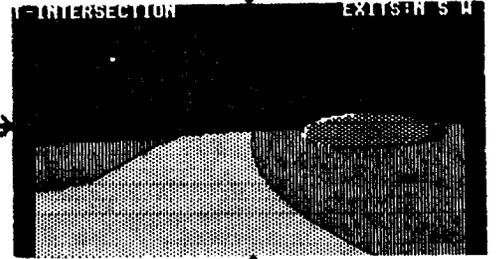
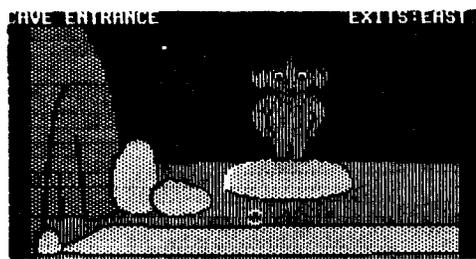
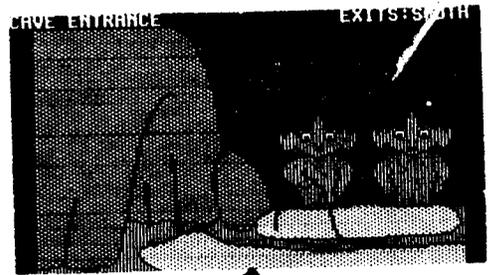
David CORNIL.

pour votre correspondance, une seule adresse :

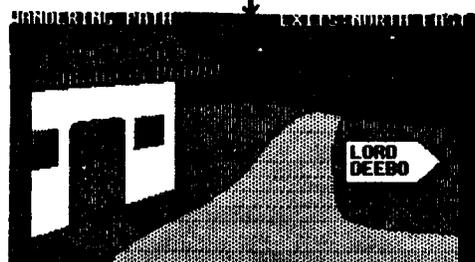
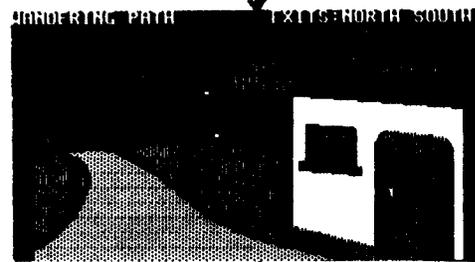
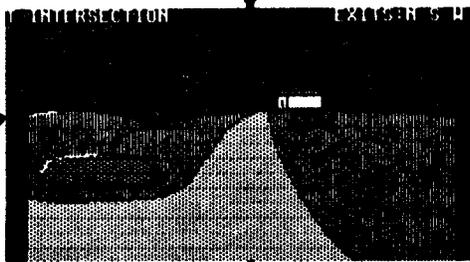
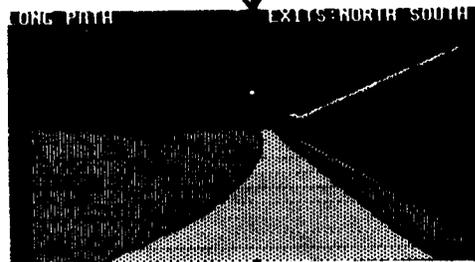
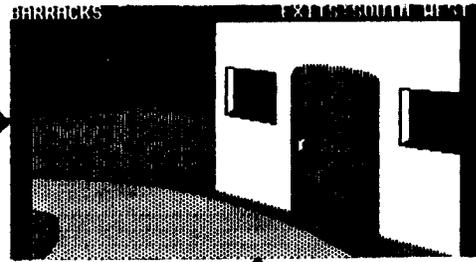
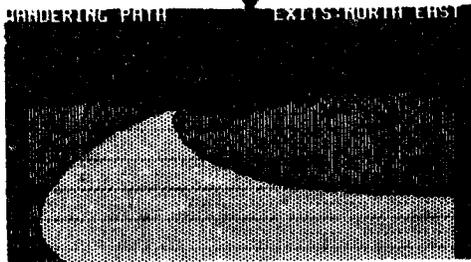
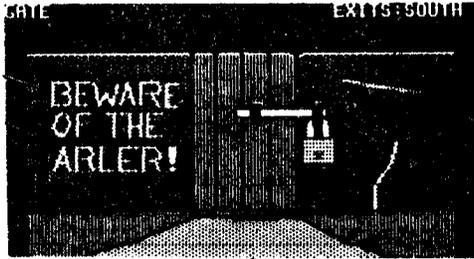
**LES AVENTURIERS DU BOUT DU MONDE
MICROCAM. CRCAM
19 rue du Pre Perche
35040 RENNES**

CAVDS IN SPACE

LE PLAN



GRUDS IN SPACE



Introduction aux langages de Programmation

Programmer les premiers ordinateurs n'était pas une tâche facile . Pour dire à l'ordinateur ce qu'on voulait qu'il fasse , l'utilisateur devait :

- connaître le détail des circuits internes de l'ordinateur,
- expliquer tout cela avec un langage ne contenant que des 0 et des 1.

Ainsi pour aider l'utilisateur, on a imaginé des langages dits "évolués" c'est à dire des langages qui :

- ne dépendent plus du type de machine sur lequel ils seront exécutés (ceci n'est pas tout à fait vrai),
- sont beaucoup plus proches de l'utilisateur non initié.

Nous allons donc expliquer succinctement ce que sont les bases de tout langage de programmation "évolué".

CARACTERISTIQUES DE BASE

Valeurs et types

Un programme est une suite d'instructions qui permet de résoudre des problèmes le plus souvent à base de calculs. Plutôt que des valeurs l'ordinateur manipule des représentations de valeurs, qui sont des configurations de bits, d'octets ou de mots de la mémoire. Devant la diversité de ces valeurs, on est obligé de spécifier leurs types (ex:décimal ou entier). Cette contrainte n'est qu'apparente et se révèle en définitive très profitable pour le programmeur car ceci le conduit à définir précisément toutes les valeurs qu'il emploie, et donc à mieux contrôler son programme.

Notez, cependant, que certains langages présentent des contraintes plus ou moins importantes. On a donc deux sortes de langage qui donnent deux démarches différentes :

- synthétique avec le BASIC, où beaucoup de ces valeurs et types ne sont pas spécifiés mais pris par défaut.
- analytique avec le PASCAL, où il est impératif de tout déclarer.

Nous énumérerons donc ci-dessous quelques types courants, et les notions de base sur leurs représentations, pour autant qu'elles soient utiles au programmeur.

1°) Caractères

Pour représenter des caractères appartenant à un "alphabet" donné (souvent l'alphabet usuel, éventuellement augmenté des majuscules et minuscules, ainsi que des chiffres décimaux, signes de ponctuation, du blanc et de quelques caractères spéciaux comme <,à,é,>,% ,etc...), on utilise un code.

Les plus utilisés aujourd'hui sont le BCD, EBCDIC et surtout au niveau de l'APPLE et autres, l'ASCII (prononcez "ASKI").

Suivant ce code il y a correspondance entre une certaine information et une certaine configuration binaire. Dans le cas de l'ASCII un caractère comme le A est représenté par 8 bits soit un octet. Par exemple lorsqu'on appuie sur la touche A du clavier, le clavier envoie dans les circuits de l'ordinateur l'octet 01000001 et on voit apparaître sur l'écran la lettre A.

2°) Textes

Un "texte" ou "chaîne de caractères" (en anglais STRING) est une suite finie de caractères, comme:"BONJOUR". Notons qu'un objet de ce type est un cas particulier de "texte", seul l'usage en est différent. La représentation des textes en mémoire ne sera pas évoquée ici.

3°) Logiques

Une valeur "logique" est l'une des deux valeurs "vrai" ou "faux" qui interviennent surtout dans les instructions et les expressions conditionnelles, c'est à dire dans les instructions capables d'influer sur le déroulement du programme.

Une valeur logique peut être représentée par l'état d'un bit : 0 ou 1.

4°) Nombres entiers

Les ordinateurs peuvent manipuler des nombres entiers positifs (+1,+2), négatifs (-1,-2) et nuls (0). Contrairement aux mathématiques un entier est représenté par un nombre fini (limité) et en général fixe de bits.

Soit n la taille des mots de l'ordinateur (dans le cas de l'APPLE n=8 du fait des possibilités offertes par le microprocesseur 6502), les nombres entiers représentables sont compris entre 0 et 2 puissance 8 moins 1 ($0 < N < 256$).

Au niveau de l'APPLE un nombre entier est codé sur 2 octets soit 16 bits ce qui correspond à un nombre maximum de 2 puissance 16 moins 1 (65535 ou -32767 à +32768).

18/ SOUVENIRS, SOUVENIRS

5°) Nombres fractionnaires ou "réels"

Un nombre réel est composé d'une partie entière et d'une partie décimale ou fractionnaire :

- 35.42 lire 35 virgule 42
- partie entière : 35
- partie décimale : 0.42

Au niveau de l'ordinateur la représentation de ces nombres se fait par une notation scientifique de la forme :

- $x = c * B$ puissance n
- > B s'appelle base du système de numération (2=binaire)
- > c s'appelle mantisse du nombre x
- > n s'appelle exposant

- Ex: $14.5 = 0.145 * 10^{\text{puissance } +2}$ (système décimal)
- ou $14.5 = 0.145 * E +2$

La prise en compte des nombres réels par l'APPLE II ne peut se faire que pour des nombres compris entre $-1.7E+38$ à $1.7E+38$.

6°) Nombres complexes

Certains langages utilisés pour le calcul scientifique permettent de manipuler des "nombres complexes". Ces derniers sont représentés comme des couples de nombres fractionnaires:

$$- (3.29, 2.31) \hat{=} 3.29 + 2.31i$$

Objets de base : constantes, variables, tableaux, expressions

=====

Pour caractériser un objet dans un programme il doit avoir non seulement une valeur et un type mais aussi un nom.

1°) Constantes

Une constante a un nom, un type et une valeur fixes:

- 435 est le nom d'une constante de type entier et de valeur fixe.
- 2.37 est le nom d'une constante de type fractionnaire et de valeur fixe.
- BEBE est le nom d'une constante de type texte et de valeur fixe.

Mais il existe certaines constantes appelées "symboliques" qui associent une constante de type entier ou fractionnaire à une constante de type texte. Elles sont utilisées fréquemment en programmation:

$$- PI = 3.141592654$$

2°) Variables

Une variable a un type et un nom fixe. Seule la valeur varie suivant les spécifications du programmeur. Le nom d'une variable est appelé "identificateur".

En basic il existe 3 types de variables :

- B représente une variable numérique de type entier ou fractionnaire.
- B% représente une variable de type entier.
- B\$ représente une variable de type texte.

Dans ce langage de programmation il n'est pas besoin de déclarer le type de la variable qui devient "implicite" .

3°) Tableaux

Un tableau associe à un nom et un type fixe plusieurs valeurs variables. Par exemple en BASIC :

- A\$(20,20)
- > A\$ représente un tableau de nom A et de type texte.
- > (,) représente la dimension du tableau, ici il a 2 dimensions.
- > (i,j) représente les indices du tableau.
- > A\$(i,j) représente la variable indicée qui désigne un élément du tableau.
- > A\$(1,1) correspondra donc à la première case en haut à gauche du tableau A\$.
- > on pourra alors stocker 400 variables de type texte dans ce tableau.

Il est donc possible de créer un tableau en déclarant son type (texte, entier ou réel), son nom (au choix du programmeur dans les limites du langage utilisé) ainsi que ses dimensions (une, deux ou trois en général) et ses bornes (limitées par la capacité du système).

4°) Expressions

Une expression désigne le calcul d'une certaine valeur. Elle met en jeu d'autres valeurs et des opérations :

- 3+27 est une expression qui met en jeu les valeurs 3 et 27 avec une addition.
- X+32 est une expression où X est le nom d'une variable de type entier ou réel ajoutée à la valeur 32.
- A*B/C-4 ici les opérations utilisées sont la multiplication, la division et la soustraction. Les valeurs sont des variables du même type que X.

Dans les expressions où il y a plusieurs opérations successives l'ordre des opérations est déterminé par la priorité des opérateurs (opérateur = signe d'opération --> +,-,/,*,^,log,exp,<, >). On dit par exemple qu'en BASIC les ordres de priorité des opérateurs arithmétiques par ordre décroissant sont :

- priorité 1 --> () les parenthèses dont le contenu est toujours évalué en premier.
- priorité 2 --> ^ élévation à une puissance.
- priorité 3 --> -1 moins unaire.
- priorité 4 --> * multiplication,
--> / division.
- priorité 5 --> + addition,
--> - soustraction.

Il existe aussi des opérateurs de relation (=,<,>,<=,>=,<>) et des opérateurs logiques (NOT,AND,OR) qui ont un niveau de priorité inférieur aux opérateurs arithmétiques .

Programme: instructions

=====

Le but d'un programme est, en général, de calculer une ou plusieurs valeurs. On aimerait pouvoir spécifier le programme par une expression simple :

- calcul de la circonférence --> $P=2*PI*R$

Les langages de programmation actuel ne le permettent pas et nous oblige à préciser les étapes du calcul par une liste d'actions appelées "instructions".

Les ensembles d'instructions forment des programmes.

On distingue 2 types d'instructions :

- Des instructions de traitement de l'information.
- Des instructions de commande de déroulement de programme.

Les instructions de second type seront vues ultérieurement.

Les instructions de traitement de l'information sont de trois sortes :

- AFFECTATION : $X \leftarrow 12$ = on met 12 dans la variable X
- LECTURE : Chercher des données sur des périphériques (disquettes, disque dur...)
- ECRITURE : Elle commande d'écrire sur un périphérique la valeur d'une expression, d'une variable ou d'une constante.

Eric RICHARD/MICROCAM SAINT-MALO

FIN DE LA PREMIERE PARTIE !



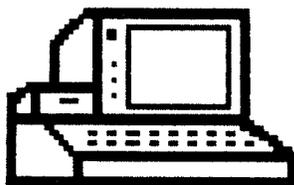
Depuis avril nous étions coincés au temps du crétacé inférieur n'ayant totalisé que 130 points sur 200. Avec la précieuse aide d'un ami, la solution est désormais trouvée.

Tout d'abord voyons pour la fronde. Lorsqu'on ramasse le baton à coté du STEGOSAURUS il faut taper BUILD SLINGSHOT, puis, une fois que l'on se trouve face à l'ALLOSAURE, USE SLINGSHOT; l'ordinateur demande quel objet sert de projectile : STONE. L'animal trépasse et on peut alors s'emparer d'un os.

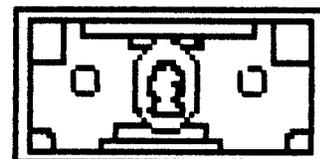
Trois dinosaures restaient sans nom. Pour les nommer il existe une manoeuvre simple que j'ignorais : au moment où l'ordinateur demande le nom, tapez RETURN, il nous fournit alors un détail pour identifier le dinosaure; tapez LIST et il apparait la liste des noms de tous les animaux du jeu, il suffit de choisir. Ainsi, le dinosaure de la clairière est un CAMPTOSAURUS et il accepte la pomme qu'il mange sans oublier de nous rendre le trognon ensuite. L'oiseau sur le tronc d'arbre est un ARCHAEOPTERYX, il mange l'insecte blessé. Enfin, celui qui mange les oeufs est un ORNITHOLESTES, le lézard fait parfaitement son affaire.

Notre score est de 195 points. Les 5 autres se gagnent en débroussaillant les herbes autour du car USE TOOTH. Voila, c'est gagné. Vous remontez dans le car et faite à nouveau un bon de 135 millions d'années pour vous retrouver au vingtième siècle.

Vincent Dutron.



Les P.A des Aventuriers



Ventes

Vends billard américain complet taille moyenne, état neuf. Prix : 800 Frs à débattre. Tel : (20) 92-38-80 après 17h30.

Vends velo Decathlon, 3 vitesses, état moyen. Prix modeste : 300 Frs. Tel : (20) 92-38-80 après 17h30.

Vends Sinclair QL complet avec moniteur ambre (Zenith) + 4 logiciels + 4 K7 vierges et 3 livres. Encore sous garantie (8 mois). Le tout : 5800 Frs. Tel : 64-53-55 (demander Olivier).

Vends cassettes Atari, état neuf, très bon état : Defender, Realsports Tennis, Othello... Prix sacrifiés... Tel : (99) 31-70-91 le soir sauf week-end.

Vends attache remorque R5 avec boule. Tel : (99) 31-70-91 le soir sauf week-end.

Contacts

Apple 2 cherche contacts pour échanges divers. Téléphonez à Antoine au (20) 92-38-80 après 17h30.

Vous qui aimez notre journal et qui desirez le faire connaître en le distribuant dans votre région, contactez David Cornil au (99) 68-94-87.

G.P.L.E.

GLOBAL PROGRAM LINE EDITOR

Enfin un programme qui vous aidera, programmeurs BASIC, pour modifier facilement vos chers programmes. Fini les touches ESC et I, J, K, M pour modifier vos programmes, voici un éditeur puissant qui vous fera gagner du temps.

GPLE travaille comme un traitement de texte pour APPLESOFT. Les lignes de programmes peuvent être éditées en positionnant le curseur à l'endroit où l'on veut insérer ou supprimer du texte et pas besoin d'aller à la fin de la ligne modifiée pour appuyer sur RETURN.

GPLE vous aide en recherchant et en remplaçant n'importe quel mot ou n'importe quelle variable dans vos programmes.

Par exemple retrouver toutes les lignes qui contiennent un GOSUB, ou imprimer ou supprimer toutes les lignes qui contiennent l'instruction REM. Remplacer n'importe quelle variable, mot ou caractère par un autre. Par exemple changer toutes les variables X en ABC ou remplacer "HELLO" par "BONJOUR".

GPLE peut travailler sur n'importe quel APPLE, en 40 ou 80 colonnes, sous DOS 3.3 ou ProDOS. En plus GPLE vous donne la possibilité de vous définir des touches de fonctions avec la touche ESC + une autre touche.

Par exemple ESC I vous donnera CATALOG Drive 1, ESC L peut exécuter "HOME:LIST". Vous pourrez définir vos propres fonctions et enrichir les fonctions fournies par GPLE.

GPLE est transparent pour vos programmes et restera actif tant que vous n'utiliserez pas les instructions FP, INT, PR&3, NEW.

Comment utiliser GPLE :

Bootez* avec la disquette GPLE et chargez la version DOS 3.3 ou Prodos, selon vos besoins.

Reportez vous au manuel GPLE (en américain) ou à l'adaptation française faite par votre serviteur.

MERCI BEAGLE BROTHERS !

YR CORNIL

*Excusez mon horrible français, je voulais dire : amorcez votre ordinateur.



NOUVELLES DU CLUB

NUIT DE LA MICRO

la première nuit de la Micro fut un réel succès. Plus de 50 branchés ont passé le premier jour de l'été (.sic), en compagnie de leurs micros préférés, ou ont pris contact pour la première fois avec ces petites merveilles de la technologie qui font quelquefois peur.

Au programme il y avait de la video avec WARGAME, des expositions de micros, des démonstrations, et surtout des appareils en libre service.

Du coté matériel il y avait:

- 6 APPLE 2e
- 1 APPLE Macintosh
- 1 T07/70 couleur
- 2 ERICSSON PC
- 1 VIDEO DISQUE PHILIPS

Le matériel venait du MICROCAM et de EXPOMATIQUE.

Devant ce franc succès, la NUIT DE LA MICRO sera reconduite en 1986.

LES REVUES

le club est abonné à de nombreuses revues, qui sont à votre disposition au local:

- revues générales:

- MICRO SYSTEME
- MICRO 7
- L'ORDINATEUR INDIVIDUEL
- SOFT ET MICRO
- S U M

- revues pour "applemaniaques":

- POM'S
- GOLDEN
- ECHO DES APPLE
- TREMLIN MICRO
- A+ (revue Américaine)

LA BIBLIOTHEQUE

pour emprunter ou pour consulter sur place, le club met à votre disposition une gamme variées de livres et de manuels.

FLASH

LE P.C. ENTRE AU MICROCAM

Le Club MICROCAM a décidé de donner une place au P.C.

Qui est ce représentant de la famille P.C.?

il est Suédois, a une bonne mémoire, une bonne vue, parle plusieurs langues, utilise plusieurs O.S.

les branchés l'auront sans doute reconnu . La seule politique du MICROCAM est de mettre à la disposition de ses Membres une gamme de matériels et de logiciels qui rendent service.

Des précisions:

il est Suédois: c'est un ERICSSON P.C. Grâce à un effort particulier de la Société ERICSSON, le club a saisi, avec le concours financier du COMITE D'ENTREPRISE, l'opportunité d'élargir la gamme de matériel. En effet, sans tomber entre les mains de "BIG BLUE", le club ne pouvait pas ignorer le phénomène du Personal Computer (en abrégé P.C.) lancé par I.B.M. avec la complicité de MICROSOFT qui a écrit MS-DOS.

il a une bonne mémoire : 256 k octets

il a bonne vue : 2 lecteurs de 360 K octets

il parle plusieurs langues: BASIC, PASCAL
FORTRAN ...

il utilise plusieurs O.S. : MS-DOS, PC-DOS ...

Ce nouvel arrivant agrandira la famille de vos petits esclaves (mais oui, c'est vous les Maitres !!):

3 APPLE 2+ , 2 APPLE 2e, 1 APPLE 2c
1 T07 , 1 T07/70
1 ERICSSON P.C.

BONNE RENTREE, ET N'oubliez PAS DE PAYER VOS COTISATIONS.

Y.R. CORNIL , trésorier (vous l'auriez deviné?)

INFOS CLUB