

# SCIENCE & VIE

**UN SCÉNARIO FOU :  
TCHERNOBYL-SUR-SEINE**

MENSUEL

N° 840 SEPTEMBRE 1987

**LE SOLEIL**

**MACHINE-MYSTÈRE**

## GOLFE : LES ARMES EN PRÉSENCE

**AUTO :**

**LES ROUES**

**INTELLIGENTES**

M 2578 - 840 - 17,00 F



3792578017005 08400

SUISSE 5 FS - CANADA \$2.95 - BELGIQUE 120 FB - ESPAGNE 560 Ptas - MAROC 17 Db - TUNISIE 2,05 Dt



# Jeu de palet

INFORMATIQUE AMUSANTE

**N**ous vous proposons ce mois-ci de jouer au "Crazy Curling", un jeu imaginaire aux règles des plus simples. Le joueur dispose d'un palet vert qu'il déplace à son gré de manière à renvoyer un palet blanc. Le but du jeu est de faire passer le plus grand nombre de fois possible ce palet blanc sur une croix rouge présente sur la piste. Cette dernière est cependant encombrée de deux heurtoirs qui provoquent des rebonds "parasites" si le palet blanc les touche. Notons enfin qu'un effet supplémentaire pourra être communiqué au palet blanc en le frappant avec le bord de celui du joueur: de même pour les heurtoirs. Pour une partie complète, chaque participant dispose au total de cinq palets. Rappelons que le meilleur score obtenu sera mémorisé et affiché en rouge en permanence sur le tableau des résultats

placé à la droite de l'écran de jeu. Ces quelques règles fixées, passons à présent à l'écriture du programme.

En premier lieu nous demandons à l'Amstrad, à la ligne 40, de passer en mode 0, ceci afin de pouvoir profiter de l'ensemble de sa palette de couleurs. Ensuite nous mettons en place la fenêtre destinée au tableau d'affichage des résultats (ligne 120) en fixant sa couleur à la ligne 130. En 140 nous établissons, par la variable CR, le crédit de palet à cinq et remettons le score à 0, par la variable SC. Les enveloppes sonores nécessaires à l'illustration musicale de ce jeu sont déterminées dans les lignes 180 à 200. Vient ensuite l'animation de la page de présentation. Ici, "Crazy Curling" s'affiche sur l'écran lettre par lettre. Pour cela, l'ensemble des caractères est mémorisé dans une chaîne Data, ligne

210, puis lu successivement par la ligne 270. Enfin la boucle des lignes 280 à 300 assure leur déplacement et leur mise en place sur l'écran.

Passons ensuite à l'initialisation de l'écran de jeu. Sa couleur est d'abord fixée par la ligne 360. L'ensemble des variables du jeu est initialisé de la ligne 410 à 450. Nous trouvons respectivement les variables DP, mémorisation du début de jeu, B et XM, chargées de contrôler le mouvement des heurtoirs, XB et YB, la position du palet blanc, E, DH et DV, assurant respectivement l'effet, le déplacement horizontal et vertical du palet blanc, et PJ contrôlant la position du palet du joueur. L'ensemble des symboles et des indications est ensuite affiché sur l'écran de jeu et sur le tableau des résultats, par les lignes 490 à 580. La ligne 490 met en place la croix rouge à atteindre et la ligne 510 la mention "palets" sur le tableau des résultats. La boucle des lignes 510 à 540 affiche le nombre de palets encore disponibles et, enfin, sont affichés le meilleur score obtenu, la mention

```

10 REM *****
20 REM * DETERMINATION DES EMVELOPPES SONORES *
30 REM *****
40 MODE 0
50 REM *****
60 REM * MSC: VARIABLE MEMORISATION DU MEILLEUR SCORE *
70 REM *****
80 LET MSC=0
90 REM *****
100 REM * MISE EN PLACE FENETRE D'AFFICHAGE RESULTATS *
110 REM *****
120 WINDOW #1,14,20,1,25
130 PAPER 0:CLS
140 LET CR=5:LET SC=0
150 REM *****
160 REM * DETERMINATION DES EMVELOPPES SONORES *
170 REM *****
180 ENV 1,1,15,1,5,-2,1,5,-1,1
190 ENV 2,1,15,2,5,-3,2,5,0,2
200 ENV 3,1,15,1,5,-1,1,5,-1,3
210 DATA C,R,A,Z,Y, .C,U,R,L,I,N,G,!
220 REM *****
230 REM * ANIMATION DE LA PRESENTATION *
240 REM *****
250 RESTORE 210
260 FOR I=1 TO 14
270 READ L$
280 FOR P=18 TO 1+1 STEP -1
290 LOCATE P,10:PEN I:PRINT L$; " ":FOR T=0 TO 2*1:NEXT T
300 NEXT P
310 NEXT I
320 FOR T=1 TO 1000:NEXT T
330 REM *****
340 REM * DETERMINATION DES COULEURS DE L'ECRAN *
350 REM *****
360 PAPER 8:CLS
370 PAPER #1,9:CLS #1
380 REM *****
390 REM * INITIALISATION DES VARIABLES *
400 REM *****
410 LET DP=0
420 LET B=0:LET XM=1
430 LET XB=4+INT(RND*8):LET YB=20
440 LET E=1:LET DH=1:LET DV=-1
450 LET PJ=6
460 REM *****
470 REM * PREPARATION DE L'ECRAN *
480 REM *****
490 LOCATE 7,2:PFX 3:PRINT CHR$(203)
500 CLS #1
510 LOCATE #1,1,3:PEN #1,5:PRINT #1,"PALETS:"
520 FOR I=1 TO CR-1
530 LOCATE #1,I+1,5:PEN #1,4:PRINT #1,CHR$(231)
540 NEXT I
550 LOCATE #1,1,8:PEN #1,3:PRINT #1,MSC
560 LOCATE #1,1,10:PEN #1,5:PRINT #1,"SCORE:"
570 LOCATE #1,1,12:PEN #1,1:PRINT #1,SC
580 IF DP=1 THEN GOSUB 1470
590 REM *****
600 REM * DEBUT DE L'ANIMATION DU JEU *
610 REM *****
620 IF XB>12 THEN LET DH=-E:SOUND 1,0,0,0,1,0,2
630 IF XB<2 THEN LET DH=E:SOUND 1,0,0,0,1,0,2
640 IF YB=1 THEN LET DV=1:SOUND 1,0,0,0,1,0,2
650 REM *****
660 REM * DETERMINATION DES REBONDS PARASITES *
670 REM *****
680 IF YB<9 OR YB>13 THEN GOTO 760
690 IF DH=0 THEN LET DH=1:LET E=1
700 IF DV=1 AND YB=9 AND (XB=XM+1 OR XB=XM+2) THEN LET DV=-1:LET E=INT(RND*3):GOTO 750
710 IF DV=1 AND YB=11 AND (XB=12-XM OR XB=11-XM) THEN LET DV=-1:LET E=INT(RND*3):GOTO 750
720 IF DV=-1 AND YB=11 AND (XB=XM+1 OR XB=XM+2) THEN LET DV=1:LET E=INT(RND*3):GOTO 750
730 IF DV=-1 AND YB=13 AND (XB=12-XM OR XB=13-XM) THEN LET DV=1:LET E=INT(RND*3):GOTO 750
740 GOTO 760
750 SOUND 1,0,0,0,2,0,15
760 LOCATE XB,YB:PRINT " "
770 REM *****
780 REM * ANIMATION DU PALET *
790 REM *****
800 LET XB=XB+DH
810 IF XB<1 THEN LET XB=1
820 IF XB>13 THEN LET XB=13
830 LET YB=YB+DV
840 LOCATE XB,YB:PEN 4:PRINT CHR$(231)
850 GOSUB 1170:GOSUB 1370
860 IF DP=0 THEN GOSUB 1470
870 IF YB<25 THEN GOTO 990
880 REM *****
890 REM * DETERMINATION PALET MANQUE *
900 REM *****
910 LET CR=CR-1
920 IF CR=0 THEN GOTO 1090

```



"score" puis le score de la partie en cours.

Le jeu commence réellement à la ligne 680. Nous contrôlons la position du palet et, éventuellement son rebond horizontal par la modification de la variable DH. Nous passons ensuite à sa trajectoire verticale par modification de DV. Notons qu'ici plusieurs cas sont à prendre en compte. En effet, il faut, d'une part, que le palet rebondisse toujours en haut de l'écran,

fonction assurée par la ligne 640, mais également dans le cas où il touche l'un des heurtoirs ou le palet du joueur. Le rebond sur les heurtoirs est assuré par les lignes 680 à 760. Notons qu'ici la valeur donnée à E, et par voie de conséquence l'effet transmis au palet, est aléatoire. Ceci permet de contrôler l'animation du déplacement du palet. Celle-ci est obtenue par les lignes 800 à 870. Elle fait appel à deux sous-routines, celle de dépla-

cement du joueur et celle de déplacement des palets parasites (heurtoirs). Tant que la valeur prise par YB est différente de 25, cela signifie que le palet n'a pas été raté par le joueur. Le programme se reboucle donc entre les lignes 800 et 870. En cas contraire, cela signifie que le joueur a raté le palet et le programme passe donc à l'étape suivante: la modification du crédit de jeu.

En ligne 910 nous soustrayons 1 à la valeur de CR. Si cette nouvelle valeur est différente de 0, donc si le crédit de palets n'est pas épuisé, le jeu se reboucle de nouveau après réinitialisation de la position du joueur et de celle du palet. En cas contraire, le programme se rend en ligne 1090 et affiche sur le tableau des résultats "Fin de jeu". Reste le contrôle du passage du palet sur la croix rouge. Il est assuré par les lignes 990 à 1050. Si le palet se trouve effectivement sur la croix, nous ajouterons à SC les points obtenus. De même, nous contrôlerons si la valeur du meilleur score, MSC, est bien inférieure à celle de SC. Si tel n'est pas le cas

```

930 LOCATE XB,YB:PRINT " "
940 LOCATE PJ+1,24:PRINT " "
950 GOTO 420
960 REM *****
970 REM *          CONTROLE DU PALET SUR LA CROIX          *
980 REM *****
990 IF XB<>7 OR YB<>2 THEN GOTO 1050
1000 LET SC=SC+100+(10*E)
1010 IF MSC<SC THEN LET MSC=SC
1020 LOCATE 7,2:PRINT CHR$(231)
1030 LOCATE PJ+1,24:PRINT " "
1040 GOTO 420
1050 GOTO 620
1060 REM *****
1070 REM *          DETERMINATION DE LA FIN DE LA PARTIE          *
1080 REM *****
1090 LOCATE #1,3,20:PEN #1,1:PRINT #1,"FIN"
1100 LOCATE #1,4,22:PRINT #1,"DE"
1110 LOCATE #1,3,24:PRINT #1,"JEU!"
1120 GOSUB 1470
1130 GOTO 130
1140 REM *****
1150 REM *          SOUS ROUTINE DEPLACEMENT DU PALET DU JOUEUR          *
1160 REM *****
1170 IF INKEY(8)=0 THEN LET PJ=PJ-1
1180 IF PJ<1 THEN LET PJ=1
1190 IF INKEY(1)=0 THEN LET PJ=PJ+1
1200 IF PJ>10 THEN LET PJ=11
1210 LOCATE PJ,24:PEN 12:PRINT " ";CHR$(231);" "
1220 IF YB<>23 THEN GOTO 1330
1230 REM *****
1240 REM *          DETERMINATION DE "L'EFFET"          *
1250 REM *****
1260 IF PJ=XB-1 THEN LET DV=-1
1270 IF PJ=XB AND DH=-E THEN LET DV=-1:LET E=E+1:LET DH=-E
1280 IF PJ=XB AND DH=E THEN LET DV=-1:LET E=E-1:LET DH=-E
1290 IF PJ=XB-2 AND DH=E THEN LET DV=-1:LET E=E+1:LET DH=E
1300 IF PJ=XB-2 AND DH=-E THEN LET DV=-1:LET E=E-1:LET DH=E
1310 IF PJ>=XB-2 AND PJ<=XB THEN LET SC=SC+1:SOUND 1,0,0,0,2,0,15
1320 IF MSC<SC THEN LET MSC=SC
1330 RETURN
1340 REM *****
1350 REM *          SOUS ROUTINE D'ANIMATION PALETS PARASITES          *
1360 REM *****
1370 IF B=0 THEN LET XM=XM+1
1380 IF B=1 THEN LET XM=XM-1
1390 IF XM=10 THEN LET B=1
1400 IF XM=1 THEN LET B=0
1410 PEN 14:LOCATE XM,10:PRINT " ";CHR$(231);CHR$(231);" "
1420 PEN 15:LOCATE 11-XM,12:PRINT " ";CHR$(231);CHR$(231);" "
1430 RETURN
1440 REM *****
1450 REM *          SOUS ROUTINE DE L'ILLUSTRATION MUSICALE          *
1460 REM *****
1470 FOR I=1 TO 4
1480 RESTORE 1490
1490 DATA 0,1,2,0,1,2,0,2,15,0,1,2,956,1,0,0,2,15
1500 FOR M=1 TO 6
1510 BORDER 15
1520 READ N:READ F:READ BR
1530 SOUND 1,N,0,0,F,0,BR
1540 BORDER 18
1550 GOSUB 1370
1560 NEXT M
1570 NEXT I
1580 FOR I=1 TO 3
1590 RESTORE 1600
1600 DATA 1,0,1,2,2,956,3,0,1,0,1,2
1610 DATA 2,676,3,0,1,0,2,15,2,638,3,0
1620 DATA 1,0,1,2,0,0,0,0,1,1276,1,0
1630 DATA 2,758,3,0,1,0,2,15
1640 FOR M=1 TO 11
1650 BORDER M+15
1660 READ C:READ N:READ F:READ BR
1670 SOUND C,N,0,0,F,0,BR
1680 NEXT M
1690 NEXT I
1700 FOR I=1 TO 3
1710 GOSUB 1370
1720 RESTORE 1730
1730 DATA 1,0,1,2,2,30,1,0,1,0,1,2
1740 DATA 2,60,1,0,1,0,2,15,2,60,2,0
1750 DATA 1,0,1,2,2,30,1,0,1,956,1,0
1760 DATA 2,30,1,0,1,0,2,15,2,60,2,0
1770 FOR M=1 TO 12
1780 BORDER M
1790 READ C:READ N:READ F:READ BR
1800 SOUND C,N,0,0,F,0,BR
1810 NEXT M
1820 NEXT I
1830 IF INKEY$="" THEN GOTO 1470
1840 LET DP=1
1850 BORDER 9
1860 RETURN

```

MSC prendra alors la valeur de SC. Ce contrôle effectué, le jeu reboucle encore une fois.

Restent nos trois sous-routines. La première, comprise entre les lignes 1170 et 1220, est consacrée aux déplacements du palet du joueur. Les lignes 1170 et 1190 permettent de modifier la position de ce dernier en fonction des touches tapées au clavier. Le palet du joueur est donc affiché en vert sur l'écran en fonction de la valeur prise par PJ grâce à la ligne 1210. Toujours dans cette sous-routine vient la détermination de l'effet en fonction de la position respective du palet blanc et de celui du joueur lors du rebond.

Cette fonction est assurée par les lignes 1260 à 1320. Notons que cette partie de sous-routine n'est utilisée que si le joueur a réussi à renvoyer le palet. La seconde sous-routine prend en charge le déplacement des heurtoirs. Celle-ci, très simple, est comprise entre les lignes 1370 et 1430. La dernière sous-routine assure l'illustration musicale accompagnant le jeu. Localisée de la ligne 1470 à 1860, elle utilise essentiellement des instructions "Sound". Notons que l'ensemble des sons et notes est mémorisé dans 3 groupes de chaînes Data. Le premier à la ligne 1490, le second des lignes 1600 à 1630 et le dernier des lignes 1730 à 1760. En effet, cette illustration musicale comporte trois phases et chaque phase possède donc son propre groupe de données.

L'utilisation de ce programme ne doit pas poser de problème particulier. Après l'avoir entièrement frappé, l'instruction "run" est demandée. "Crazy Curling" doit alors s'inscrire sur l'écran puis le tableau de jeu, accompagné de l'illustration sonore (rythme de batterie très syncopé). Pour commencer une partie il suffit d'appuyer sur une touche quelconque du clavier. Le jeu débutera à la fin de la musique. Le joueur doit alors placer son palet à l'aide des flèches de déplacement du curseur (flèche à droite et flèche à gauche) de manière à renvoyer le palet blanc, essayant de le faire passer sur la croix rouge. Rappelons qu'un effet supplémentaire est obtenu en frappant le palet sur l'un de ses bords au lieu de le frapper de face.

Henri-Pierre Penel

# Un joystick proportionnel

INFORMATIQUE PRATIQUE

Sur un joystick conventionnel la position de la manette ne commande que la direction de déplacement du curseur, ou du joueur, sa vitesse restant liée au programme. Dans notre cas, au contraire, nous utiliserons deux potentiomètres pour commander directement la position du curseur. L'un agira sur l'axe des X et l'autre sur celui des Y. Ainsi sa position sera directement fonction de celle des deux potentiomètres et sa vitesse de déplacement ne sera donc plus liée au programme. Comme illustration de cette propriété nous vous proposons de transformer, grâce à un petit programme très simple, votre ordinateur en "Télécran". Ce jeu bien connu des enfants permet de tracer à l'aide de deux boutons des figures sur un écran effaçable. Cependant notre joystick proportionnel vous sera certainement beaucoup plus utile si vous l'utilisez pour commander les déplacements du joueur dans un programme de jeu de votre cru.

Le principe de fonctionnement de ce montage est des plus simples. Nous nous contenterons en effet de transformer la tension variable issue de chaque potentiomètre en une valeur numérique comprise sur un octet. Pour effectuer cette conversion, nous utiliserons deux convertisseurs analogiques-digital du type ADC 0804. L'un d'entre eux sera également utilisé comme générateur d'horloge et pilotera directement le second ; ceci simplifie le câblage de ce joystick. Enfin notons que nous avons deux valeurs à fournir à l'ordinateur : l'une concernant l'axe des X et l'autre celui des Y. Notre interface principale ne pouvant transmettre qu'une seule valeur à la fois, nous enverrons ces deux données alternativement. Pour cela, l'ordinateur demandera à l'interface de sélectionner l'octet à lui transmettre, et ce grâce à une instruction "OUT". En effet, si l'ordinateur envoie 254 (OUT 255, 254), il recevra l'octet concernant l'axe des X et s'il envoie 253 (OUT 255, 253) il recevra celui concer-

nant l'axe des Y : tout ceci dans le cas où nous travaillons avec une machine équipée d'un Z 80 (Sinclair par exemple).

La réalisation de ce montage est également très simple : seuls deux circuits intégrés sont nécessaires, les deux convertisseurs ADC 0804. Le câblage de ces derniers sera, comme toujours, effectué en wrapping ainsi que leur connection au support 24 points nous servant de

```
5 BORDER 1:PAPER 0:INK 5:CLS
10 LET XP=0:LET YP=0
15 OUT 255,254
20 LET X=IN 255
25 OUT 255,253
30 LET Y=IN 255
100 IF XP=0 AND YP=0 THEN GOTO 120
110 DRAW X-XP,(Y-YP)/1.5
120 PLOT X,Y/1.5
130 LET XP=X:LET YP=Y
200 GOTO 15
```

connecteur vers l'interface principale. Les potentiomètres seront reliés aux circuits à l'aide de fils de câblage classique. Précisons que si vous avez la fâcheuse tendance de malmener vos joysticks, nous vous conseillons vivement de prévoir des fils assez longs de manière à ce que tous les éléments électroniques soient en sécurité et que seuls les potentiomètres supportent vos humeurs.

Simplicité toujours avec le programme que nous vous proposons. Il permet, comme nous l'avons dit, d'effectuer des tracés sur l'écran à l'aide de notre joystick proportionnel. Voyons rapidement ses grandes lignes. Nous l'avons écrit ici sur Sinclair spectrum +2. La ligne 5 fixe simplement la couleur de l'écran, de son pourtour et du tracé. Nous prévoyons ensuite deux variables XP et YP que nous utiliserons par la suite pour le tracé. A la ligne 15 nous demanderons la valeur de X grâce à OUT 255, 254 puis celle-ci sera attribuée à X en ligne 20. Nous pratiquerons de même pour Y aux lignes 25 à 30. Enfin, les lignes 110 et 120 effectueront notre tracé. Notons que pour le MO 5 la ligne 15 deviendra POKE 43006,254 et la ligne 20 : X = PEEK (43007). De même, les lignes

25 et 30 devront être remplacées par :

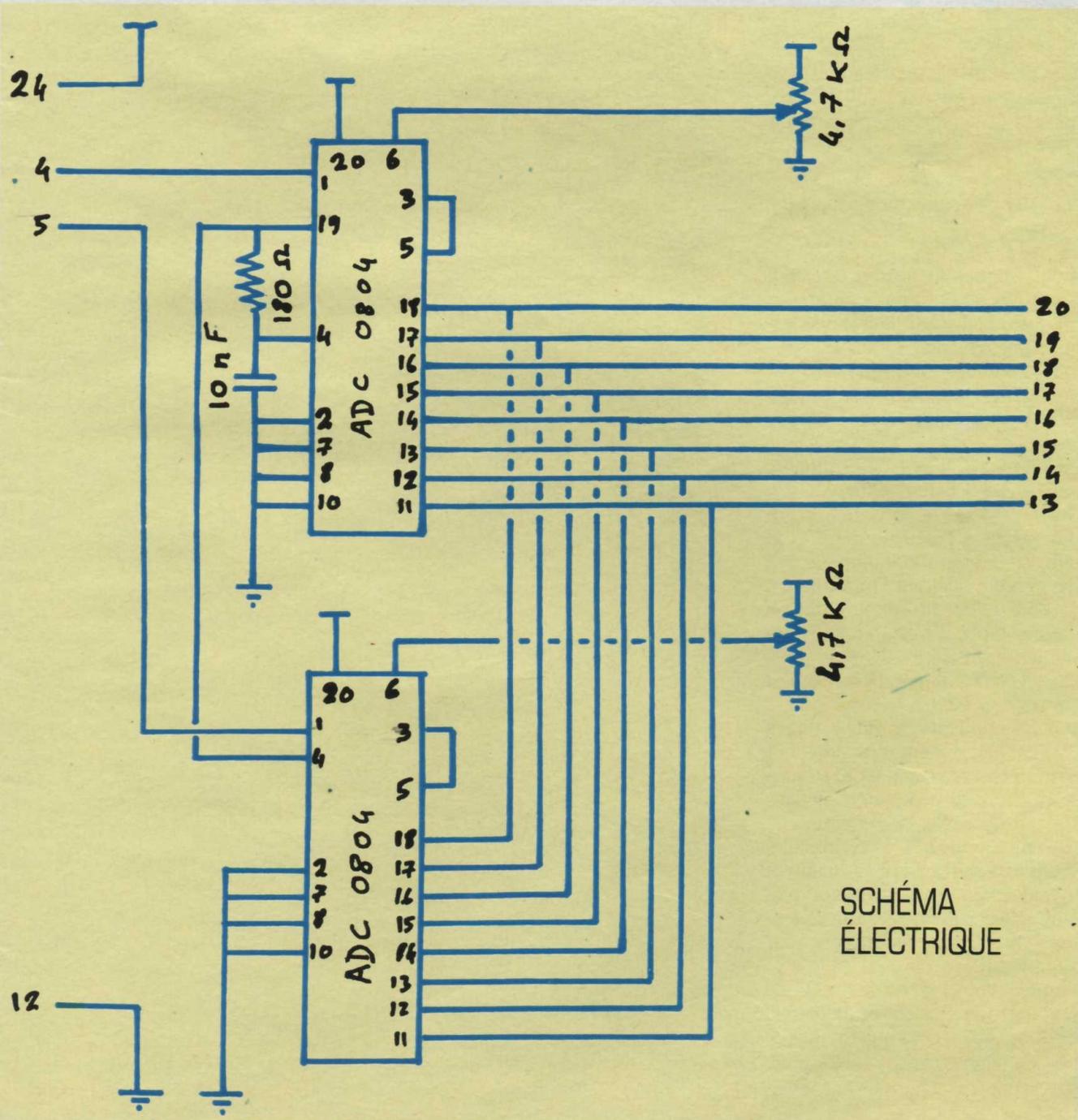
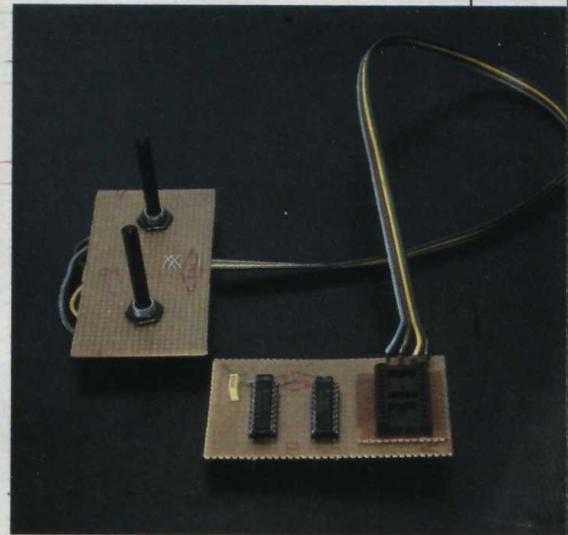
- 25 POKE 43006,253
- 30 Y = PEEK (43007)

Précisons que les valeurs fournies pour X et Y par notre réalisation seront comprises entre 0 et 255. Ceci ne correspond pas forcément au format de l'écran de l'ordinateur en mode graphique. Pour le Spectrum, par exemple, si cette gamme de valeur convient pour X il n'en est pas de même pour Y. C'est pourquoi, dans ce cas, nous appliquerons à Y un facteur correctif, 1,5 en l'occurrence, lors de l'instruction PLOT de la ligne 120.

Il nous reste à mettre en place

notre joystick. En premier lieu, avant de mettre l'ordinateur sous tension, l'interface principal lui sera connectée. Puis ce nouveau montage sera raccordé à l'interface grâce à notre connecteur. L'ordinateur sera alors mis sous tension et le programme frappé. Une fois RUN demandé, les tracés sur l'écran seront obtenus en agissant sur les potentiomètres. Rappelons que les lignes 15 à 30 de notre programme sont les seules indispensables au fonctionnement de ce joystick proportionnel. Vous pourrez donc les réutiliser pour les insérer dans un programme de votre choix.

Henri-Pierre Penel



## La méthode de Gauss - Seidel

LE "MICRO" DE L'INGÉNIEUR

Tout bachelier a sinon retenu du moins appris la résolution des systèmes d'équations à 2 ou 3 inconnues, par la méthode dite des déterminants. Ainsi le système :

$$\begin{cases} 2x + 3y = 1 \\ 3x + 2y = -1 \end{cases}$$

peut-il s'écrire :

$$\begin{matrix} 2 & 3 & x & = & 1 \\ 3 & 2 & y & = & -1 \end{matrix}$$

et aboutit à la solution unique  $x = \Delta x / \Delta$ ,  $y = \Delta y / \Delta$ , où  $\Delta$  est le déterminant de la matrice

$$\begin{vmatrix} 2 & 3 \\ 3 & 2 \end{vmatrix}, \text{ et } \Delta x, \Delta y \text{ les déterminants de la même matrice où l'on a remplacé respectivement la première et la deuxième colonne par}$$

$$\begin{matrix} 1 \\ -1 \end{matrix}$$

Le déterminant d'une matrice  $\begin{matrix} a & c \\ b & d \end{matrix}$  valant  $ad - bc$ , la formule

vue plus haut pour le calcul de  $x$  et de  $y$  aboutit au résultat  $x = -1$  et  $y = 1$ . Cette méthode est applicable à tout système dans lequel le nombre  $n$  d'équations est le même que celui d'inconnues (la matrice est alors dite carrée d'ordre  $n$ ).

Seulement, si les déterminants sont faciles à calculer lorsque  $n=2$  ou  $n=3$ , ils le sont déjà beaucoup moins à partir de  $n=4$ . A la main, la résolution commence à être pénible dès que l'on atteint  $n=5$  ou  $6$ , relève de l'exploit pour  $n$  compris entre  $10$  et  $15$ , et décourage les ordinateurs les plus puissants, CRAY 2 y compris, pour  $n$  supérieur à  $50$ .

Or les applications scientifiques et techniques actuelles jonglent couramment avec des systèmes comportant des milliers ou des dizaines de milliers d'équations.

La méthode des déterminants est ainsi à rejeter au profit d'autres techniques même si, théoriquement, elle garde une grande valeur, pour des démonstrations par exemple.

Nous avons décrit la méthode de Gauss dans notre numéro 809 de février 1985, et avons dit qu'elle était applicable pour  $n$  valant quel-

ques dizaines ou, à la rigueur, quelques centaines.

De fait, la précision de ses résultats est d'autant meilleure que la matrice de départ est "creuse" (avec beaucoup de zéros) et que

les valeurs qu'elle contient sont comparables en regard des valeurs extrêmes que peut stocker l'ordinateur.

La méthode de Gauss est dite directe car les  $n$  solutions s'obtiennent directement à partir d'un algorithme décomposable en multiplications, divisions, additions. Or on conçoit que plus le nombre de ces opérations élémentaires est grand, et plus les erreurs

```

10 REM PROGRAMME DE RESOLUTION D'UN SYSTEME LINEAIRE DE CRAMER PAR LA
    METHODE ITERATIVE DE GAUSS - SEIDEL
20 REM
30 HOME : VTAB 10: PRINT " RESOLUTION D'UN SYSTEME DE N EQUATIONS LINEA
    IRES A N INCONNUES DU TYPE": PRINT : PRINT " (A)(X) = (B)"
50 PRINT : INPUT "COMBIEN VAUT N ? ";N
60 DIM A(N,N),B(N),X(N),ODR(N),A1(N,N),B1(N)
70 ER = 0.001:MI = 1E - 9:NMAX = 60
80 PRINT : PRINT "INTRODUCTION DES DONNEES": PRINT
90 FOR K = 1 TO N
100 PRINT : PRINT "INTRODUCTION DE LA ";K;"ième EQUATION": PRINT
110 FOR L = 1 TO N
120 PRINT "COEFFICIENT DE X";L;: INPUT " = ? ";A1(K,L)
130 NEXT L
140 PRINT : INPUT "SECOND MEMBRE = ? ";B1(K)
150 NEXT K
160 PRINT : PRINT "VOULEZ-VOUS ";: PRINT
170 PRINT "1. VERIFIER LES MATRICES (A) ET (B)": PRINT
180 PRINT "2. MODIFIER UN ELEMENT DE (A)": PRINT
190 PRINT "3. MODIFIER UN ELEMENT DE (B)": PRINT
200 PRINT "4. RESOUDRE LE SYSTEME": PRINT
210 PRINT "5. ARRETER LE PROGRAMME": PRINT
220 INPUT R
230 ON R GOTO 240,320,370,500,410
240 PRINT : FOR K = 1 TO N
250 PRINT : PRINT "LIGNE n" ";K: PRINT
260 FOR L = 1 TO N
270 PRINT : PRINT "COEFFICIENT DE X";L;" = ";A1(K,L),
280 NEXT L
290 PRINT : PRINT "SECOND MEMBRE = ";B1(K)
300 NEXT K
310 GOTO 160
320 PRINT : INPUT "n" DE LIGNE= ? ";K
330 INPUT "n" DE COLONNE = ? ";L
340 PRINT "COEFFICIENT ACTUEL = ";A1(K,L)
350 PRINT : INPUT "NOUVELLE VALEUR = ? ";A1(K,L)
360 GOTO 160
370 PRINT : INPUT "n" DE LIGNE = ? ";K
380 PRINT : PRINT "VALEUR ACTUELLE = ";B1(K)
390 PRINT : INPUT "NOUVELLE VALEUR = ? ";B1(K)
400 GOTO 160
410 END
500 REM
510 REM RESOLUTION DU SYSTEME
520 REM
530 FOR K = 1 TO N
540 FOR L = 1 TO N
550 A(K,L) = A1(K,L)
560 NEXT L
570 B(K) = B1(K):ODR(K) = K
580 NEXT K
590 GOSUB 2000: REM DEMANDE DE PIVOTAGE
600 FOR K = 1 TO N
610 W = A(K,K): IF ABS(W) > MI GOTO 630
620 HOME : PRINT : PRINT "REDUCTION DE LA MATRICE IMPOSSIBLE": GOTO 160

630 B(K) = B(K) / W:X(K) = B(K)
640 FOR L = 1 TO N
650 IF K < L THEN A(K,L) = - A(K,L) / W
660 NEXT L
670 A(K,K) = 0
680 NEXT K
690 REM
700 REM CALCUL DU NOMBRE D'ITERATIONS NN
710 SA = 0:SB = 0:S = 0
    
```

d'arrondi provenant de la machine s'accumulent, fournissant des résultats à la précision décroissante.

Ainsi, dès que  $n$  dépasse quelques centaines on préfère utiliser des méthodes dites itératives, c'est-à-dire qui approchent peu à peu la solution par rebouclage. Un exemple d'algorithme itératif est la méthode de Newton utilisée pour la résolution des fonctions, et que nos lecteurs fidèles connaissent

maintenant bien.

Il existe plusieurs méthodes de résolution de systèmes par itérations, mais nous allons en décrire une seule, celle appelée "Gauss-Seidel". Pour être tout à fait honnête, son emploi n'est pas pleinement justifié sur notre Apple II car la capacité mémoire de cette machine limite  $n$  à 55 et autorise encore l'emploi de la méthode de Gauss, plus rapide. De plus, elle

n'est pas véritablement universelle, car elle n'est efficace que pour une certaine catégorie de matrices. Seulement, outre le fait de donner un (petit) aperçu de ce qui se fait de mieux en matière d'analyse numérique, cet algorithme sera utilisé dans une prochaine rubrique, pour le calcul des fonctions Spline, lesquelles, nous le verrons, ont un intérêt graphique et esthétique indéniable.

```

720 FOR K = 1 TO N
730 FOR L = 1 TO N
740 S = S + ABS (A(K,L))
750 NEXT L
760 IF S < 1 GOTO 780
770 HOME : PRINT : PRINT "LA CONVERGENCE EST IMPOSSIBLE": PRINT "LA MAT
RICE N'EST PAS ASSEZ CREUSE": PRINT : GOTO 160
780 IF SA < S THEN SA = S
790 S = 0: IF ABS (B(K)) > SB THEN SB = ABS (B(K))
800 NEXT K
810 NN = INT ( LOG (ER * (1 - SA) / SB) / LOG (SA))
820 IF NN > NMAX THEN NN = NMAX
830 REM
840 REM CALCUL DES NN ITERATIONS
850 FOR T = 1 TO NN
860 FOR K = 1 TO N
870 X(K) = B(K)
880 FOR L = 1 TO N
890 X(K) = X(K) + A(K,L) * X(L)
900 NEXT L
910 NEXT K
920 NEXT T
930 REM
940 REM MISE EN ORDRE DE LA SOLUTION
950 FOR K = 1 TO N - 1
960 FOR L = K + 1 TO N
970 IF ODR(K) < ODR(L) GOTO 1000
980 R = ODR(K):ODR(K) = ODR(L):ODR(L) = R
990 R = X(K):X(K) = X(L):X(L) = R
1000 NEXT L
1010 NEXT K
1020 HOME : REM AFFICHAGE DE LA SOLUTION
1030 PRINT : PRINT : PRINT "*** SOLUTION: ***": PRINT : PRINT
1040 FOR K = 1 TO N
1050 PRINT "X";K;" = ";X(K),
1060 NEXT K
1070 PRINT : IF NN = NMAX THEN PRINT "LA PRECISION DEMANDEE N'EST PAS
ASSUREE DU FAIT DE LA CONVERGENCE TROP LENTE DU PROCESSUS": PRINT
1080 GOTO 160
2000 REM
2010 REM PIVOTAGE DES LIGNES ET DES COLONNES
2020 REM
2030 N1 = N - 1
2040 FOR KU = 1 TO N1
2050 PIVOT = A(KU,KU)
2060 KX = KU:LX = KU
2070 FOR K = KU TO N
2080 FOR L = KU TO N
2090 W = A(K,L): IF ABS (PIVOT) > ABS (W) GOTO 2110
2100 PIVOT = W:KX = K:LX = L: REM NOUVEAU PIVOT
2110 NEXT L
2120 NEXT K
2130 IF KX = KU GOTO 2180
2140 FOR L = 1 TO N: REM ECHANGE DES LIGNES KU ET KX
2150 R = A(KX,L):A(KX,L) = A(KU,L):A(KU,L) = R
2160 NEXT L
2170 R = B(KX):B(KX) = B(KU):B(KU) = R
2180 IF LX = KU GOTO 2230
2190 FOR K = 1 TO N: REM ECHANGE DES COLONNES KU ET LX
2200 R = A(K,KU):A(K,KU) = A(K,LX):A(K,LX) = R
2210 NEXT K
2220 R = ODR(LX):ODR(LX) = ODR(KU):ODR(KU) = R: REM ENREGISTREMENT DE C
ETTE DERNIERE PERMUTATION DANS LE VECTEUR ODR
2230 NEXT KU
2240 RETURN

```

```

LIGNE n° 4
COEFFICIENT DE X1 = 0
COEFFICIENT DE X2 = 1
COEFFICIENT DE X3 = 1
COEFFICIENT DE X4 = 0
COEFFICIENT DE X5 = 6
COEFFICIENT DE X6 = 0
SECOND MEMBRE = 2
LIGNE n° 5
COEFFICIENT DE X1 = 0
COEFFICIENT DE X2 = 1
COEFFICIENT DE X3 = 0
COEFFICIENT DE X4 = 0
COEFFICIENT DE X5 = 0
COEFFICIENT DE X6 = 3
SECOND MEMBRE = 2

```

Une des options du programme permet de vérifier, ligne à ligne, les coefficients introduits dans la matrice.

**Formulation.** Le système initial s'écrit classiquement :

$$\begin{cases} a_{11} x_1 + \dots + a_{1n} x_n = b_1 \\ \dots \\ a_{n1} x_1 + \dots + a_{nn} x_n = b_n \end{cases}$$

Si les  $a_{11}, a_{22}, \dots, a_{nn}$  sont non nuls, l'ensemble peut s'écrire :

$$\begin{cases} x_1 = \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2 - \dots - \frac{a_{1n}}{a_{11}} x_n \\ x_2 = \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1 - \frac{a_{23}}{a_{22}} x_3 - \dots - \frac{a_{2n}}{a_{22}} x_n \\ \dots \\ x_n = \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1 - \dots - \frac{a_{nn-1}}{a_{nn}} x_{n-1} \end{cases}$$

Le principe de la résolution est alors le suivant. Partant d'un ensemble de valeurs  $x_1, \dots, x_n$  a priori quelconques (le terme consacré est n-uplet  $(x_1, \dots, x_n)$ ), la formule plus haut permet de recalculer  $x_1, \dots, x_n$ . Ce deuxième ensemble est différent du premier et, si on recommence le processus plusieurs fois, les n-uplets  $(x_1, \dots, x_n)$  vont converger vers la solution finale dans la mesure où certaines conditions sont remplies. Quelles sont



# Championnat de France des jeux mathématiques

CONCOURS

vérifications ou corrections de A et de B, afin d'être utilisées pour un nouveau calcul, par exemple. Lors de la vérification complète de la matrice A, le programme affichera tous ses éléments et, comme le défilement est un peu trop rapide, il faudra de temps en temps appuyer sur CTRL/S pour interrompre l'affichage et avoir le temps de lire les coefficients. En appuyant à nouveau sur S, le défilement reprendra.

Les lignes 530 à 580 initialisent les tableaux A et B à partir des données introduites et/ou corrigées par l'utilisateur, avant la résolution proprement dite, laquelle débute par un appel au sous-programme 2000 qui effectue le pivotage des lignes et des colonnes. A chaque permutation de colonne le vecteur ODR enregistre le nouvel ordonnancement des inconnues qui en résultent.

La ligne 610 détecte si un élément de la nouvelle matrice est inférieur à  $10^{-9}$  en valeur absolue, auquel cas il sera considéré comme étant nul et la résolution ne pourra continuer (affichage d'un message). La réduction de la matrice (passage des  $a_{ki}$  aux  $c_{ki}$ ) débute à la ligne 630 et la norme SA de la matrice A est calculée lignes 700 à 800. Si  $SA \geq 1$ , un message indiquera que la convergence est impossible. Autrement, après le calcul du nombre NN d'itérations à effectuer (limité à 60), le programme procédera à ces itérations, jusqu'à la ligne 920.

La remise en ordre des inconnues tiendra compte du vecteur ODR, avant affichage de la solution définitive. Si n est grand, il sera encore utile d'utiliser la commande CTRL/S pour interrompre l'affichage. Après résolution, le menu ligne 160 proposera de modifier A ou B pour un nouveau calcul.

**Exemple.** Soit à résoudre le système d'équations (écrit sous forme matricielle) :

$$\begin{array}{cccccc} 1 & 0 & 3 & 0 & 0 & 1 & x_1 & 2 \\ 4 & 0 & 0 & 2 & 1 & 0 & x_2 & 2 \\ 0 & 3 & 0 & 0 & 0 & 1 & x_3 & 2 \\ 0 & 1 & 1 & 0 & 6 & 0 & x_4 & 2 \\ 0 & 1 & 0 & 0 & 0 & 3 & x_5 & 2 \\ 0 & 1 & 0 & 4 & 1 & 0 & x_6 & 2 \end{array}$$

Après 27 itérations, le "micro" affichera (photo ci-contre) la solution.

Daniel Ferro

**P** arthenay, 9 juillet au soir. Dans les quelques restaurants de la ville, on parlait... mathématiques.

Le médecin strasbourgeois, le lycéen lillois et le collégien toulousain, à la même table, évoquaient entre deux spaghettis les jeux-problèmes de la demi-finale. « Que nous réservent les organisateurs pour demain ? » « J'espère qu'il ne faudra pas encore utiliser le théorème de Pythagore, qui n'est pas à mon programme », s'inquiète l'élève de quatrième.

Il allait être déçu. Car le championnat est en dehors des programmes. Mais, Pythagore, c'est de la culture. Ainsi, pendant quarante-huit heures, Parthenay a vécu à l'heure des jeux mathématiques. Et voir ces dizaines de mordus parler avec passion d'une compétition pour laquelle ils avaient parfois fait mille kilomètres, écourté des vacances, ou pris avec le sourire un logement quelque peu spartiate, cela vous récompensait de l'énergie dépensée une année durant, à préparer et gérer ce championnat.

Les concurrents ont fait preuve de qualités exceptionnelles. Dans la salle de sport, transformée en temple studieux, ils ont réalisé des prodiges avec les 12 problèmes qui leur étaient soumis (6 par jour, qu'ils avaient trois heures pour résoudre).

Le vainqueur, Dominique Ceugniet, est venu à bout de 10 d'entre eux, ce qui n'est pas un mince exploit. Le mois dernier, vous avez eu l'énoncé de trois problèmes. En voici trois autres :

## ENTRAÎNEZ-VOUS

### • Les douze points

Parmi les douze points du dessin, quel est le nombre maximum N de points qu'on peut choisir sans que 4 quelconques d'entre eux puissent former un rectangle à côtés parallèles aux axes ?

$$N = \square \square$$



### • Après la virgule

On effectue la somme :

$$1/10 + 2/100 + 3/1\,000 + 4/10\,000 + \dots + N/10^N$$

### • Les douze points

La réponse est 7. En effet, si on place trois points sur une colonne, aucune autre ne peut en contenir plus d'un, et on ne pouvait dépasser 6 points sans former un

autrement dit  $0,1 + 0,02 + 0,003 + \dots$  et ainsi de suite à l'infini, et on écrit cette somme dans le système décimal. On obtient ainsi un certain nombre. Dans ce nombre, quels sont les 1987<sup>e</sup>, 1988<sup>e</sup>, 1989<sup>e</sup> et 1990<sup>e</sup> chiffres après la virgule ?

$$\square \square \square \square$$

1987<sup>e</sup>    1988<sup>e</sup>    1989<sup>e</sup>    1990<sup>e</sup>

### • Tricubes

Trouver les nombres entiers positifs N de trois chiffres, égaux à la somme des cubes de leurs chiffres.

$$\square \square \square \quad \square \square \square \quad \square \square \square$$

$$\square \square \square \quad \square \square \square \quad \square \square \square$$

$$\square \square \square \quad \square \square \square \quad \square \square \square$$

## SOLUTIONS

Voici les solutions (brèves) des trois problèmes du jour.

rectangle.

En plaçant 2 points par colonne, on remarque que 2 des 4 colonnes seront identiques, et conduisent à un rectangle. Il est donc impossible de placer 8 points sans former de rectangle.

Or, il existe une solution à 7 points :