```
procedure CONVENTIER (VALCHA : string ; var VALENT : ENTIER) ;
var     I           : integer ;
        CAR         : char ;
        NEGATIF : boolean ;
begin
   VALENT : = 0 ;
   NEGATIF : = FALSE ;
   for I: = 1 to LENGTH (VALCHA) do
     begin
         CAR : = VALCHA [I] ;
         if CAR in ['0'..'9']
            then
               VALENT : = 10 * VALENT + ORD (CAR) — ORD ('0')
            else
               if CAR in ['+','-']
                  then
                     NEGATIF : = CAR = '—'
     end ;
   if NEGATIF then VALENT : = — VALENT
end ;
```

recherche de l'indice de l'élément, le calcul de son adresse, et un test pour vérifier s'il ne se situe pas au-delà de la longueur de la chaîne.

Lorsque le caractère est un chiffre, il intervient dans l'évaluation du nombre. Celui-ci est multiplié par 10, ce qui a pour effet de décaler tous les chiffres qui le composent vers la gauche. La valeur numérique du caractère lui est ajoutée, et se situe donc à la position des unités.

Lorsque le caractère est un signe, l'expression logique « le signe est négatif » est mémorisée dans la variable NEGATIF.

A la fin du traitement, le nombre est pris égal à son opposé si le signe *moins* a été détecté dans la chaîne, c'est-à-dire si NEGATIF est vrai.

Thierry CHAMORET

BALISTIQUE SUR PB-700

QUESTION DE MOUVEMENT

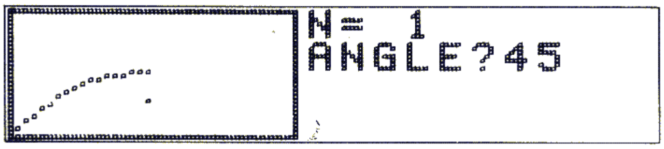
IL était une fois une équation. Devenue programme, elle transforma un phénomène physique peu connu, la balistique, en jeu... Ce n'est pas un conte de fées, mais une démarche intéressante – peut-être même un premier pas vers l'Enseignement Assisté par Ordinateur – suivie ici par le PB-700.

Posez à quelqu'un qui n'est pas très familiarisé avec les notions de mécanique la colle suivante : comment représenter la trajectoire d'un projectile lancé obliquement ou horizontalement, en négligeant la résistance de l'air.

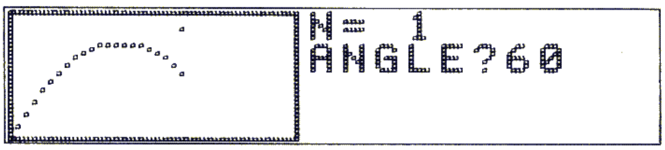
Votre interlocuteur vous tracera probablement une trajectoire oblique ou horizontale qui s'incurve progressivement pour devenir verticale... Ce petit test montre que l'intuition est souvent trompeuse et qu'en matière de balistique (puisque c'est de cela qu'il

Figure 1
Exécution du programme 1. Le petit point représente la cible

a. L'angle choisi (45 degrés) est trop important.



b. Quel que soit l'angle initial, la cible reste inaccessible. (Cas où l'équation de la trajectoire n'admet pas de racine correspondant aux coordonnées de la cible).



s'agit), bien des gens en sont restés à des concepts dépassés.

Les deux jeux proposés ici sont des variations sur ce thème de la balistique. Ils mettent à profit l'écran "haute résolution" du PB-700.

Dans le premier jeu, il s'agit d'atteindre une cible en ajustant correctement l'angle de tir d'un projectile ; celui-ci part du coin inférieur gauche de l'écran.

Trois niveaux de difficulté sont proposés (ligne 30 du programme 1) de 1 à 3. Selon le niveau choisi, la cible est soit immobile (niveau 1), soit en mouvement ; dans ce dernier cas, elle se

Programme 1

```
10 REM ---BALISTIQUE---
15 REM   *(JEU No 1)*
20 REM ================
25 CLS :N=1:G=10:V=25
30 INPUT "NIVEAU DE DIFFICULTE(1-2-3)
   ";D:CLS
40 GOSUB 200
45 IF N=6 THEN 100
50 DRAW(X0,Y0)
60 GOSUB 300
70 GOSUB 400
75 IF INT(Y+.5)=Y0 THEN 100
80 IF D=1 THEN 45
85 DRAWC(X0,Y0)
90 X0=X0-4:IF D=3 THEN Y0=Y0-1
95 IF X0*Y0>0 THEN 50
100 BEEP :BEEP :BEEP 1
110 GOTO 20
200 REM"CADRE ET CIBLE"
210 DRAW(0,0)-(70,0)-(70,31)-(0,31)-(0
   ,0)
220 Y0=ROUND(RND*25,-1)+3
230 X0=ROUND(RND*34,-1)+16
240 IF X0 MOD 2=1 THEN X0=X0+1
250 RETURN
300 REM"ANGLE DE TIR"
310 LOCATE 9,0:PRINT "N=";N
320 LOCATE 9,1:PRINT "ANGLE=    "
330 LOCATE 14,1:INPUT A
340 RETURN
400 REM"TRAJECTOIRE"
410 N=N+1:X1=0:Y1=0
420 B=G/2/V/V/COS(A)/COS(A):C=TAN(A):P
   =C/B
430 IF X0<P THEN P=X0
440 IF A>70 THEN I=1 ELSE I=2
450 FOR X=0 TO P STEP I
460 Y=B*X*X-C*X+31
470 DRAWC(X1,Y1):DRAW(X,Y):X1=X:Y1=Y
480 NEXT X
490 DRAWC(X1,Y1):BEEP
495 RETURN
```

Programme 2

```
10 REM ---BALISTIQUE---
11 REM   *(JEU No 2)*
12 REM   ============
15 G=10:K=.5:N=1:X1=0:Y1=0
20 GOSUB 100
30 GOSUB 200
40 GOSUB 300
50 IF ABS(X0-P0)<1 THEN 95
55 IF R*P<8 THEN 30
60 P1=P0+K*P:A=A/K
65 IF P1<80 THEN 80
70 P1=80:R=0
80 GOSUB 400
90 GOTO 50
95 BEEP :BEEP :BEEP 1:GOTO 10
100 REM"CADRE ET CIBLE"
110 CLS :DRAW(0,0)-(0,31)-(80,31)-(80,
   0)-(0,0)
120 X0=ROUND(RND*68,-1)+10
130 DRAWC(X0-1,31)-(X0+1,31)
140 DRAW(X0-2,30):DRAW(X0+2,30)
150 RETURN
200 REM "VITESSE"
210 LOCATE 11,0:PRINT "N=";N
220 LOCATE 11,1:PRINT "V=    "
230 LOCATE 13,1:INPUT V
240 R=1:N=N+1
250 RETURN
300 REM"PREMIERE CHUTE"
310 P0=V*SQR(62/G):P=2*P0:A=.5*G/V/V:B
   =SQR(62*G)/V
320 FOR X=0 TO P0 STEP 2
325 DRAWC(X1,Y1):Y=A*X*X
330 DRAW(X,Y):X1=X:Y1=Y
340 NEXT X
350 BEEP
355 DRAWC(X1,Y1)
360 RETURN
400 REM"REBOND"
410 FOR X=P0+1 TO P1
420 DRAWC(X1,Y1)
430 Z=X-P0:Y=A*Z*Z-B*Z+31
440 DRAW(X,Y):X1=X:Y1=Y
450 NEXT X
460 BEEP :P0=P1:P=K*P
480 RETURN
```

BALISTIQUE SUR PB-700

QUESTION DE MOUVEMENT

rapproche de vous en vol horizontal (niveau 2) ou en gagnant de l'altitude (niveau 3).

Dans la première hypothèse, le nombre d'essais est limité à cinq. Pour les deux autres niveaux, on a tout le loisir d'ajuster le tir jusqu'au moment où la cible sort de l'espace de contrôle, ou encore, éventualité plus dramatique, où elle se trouve à la verticale du poste de tir...

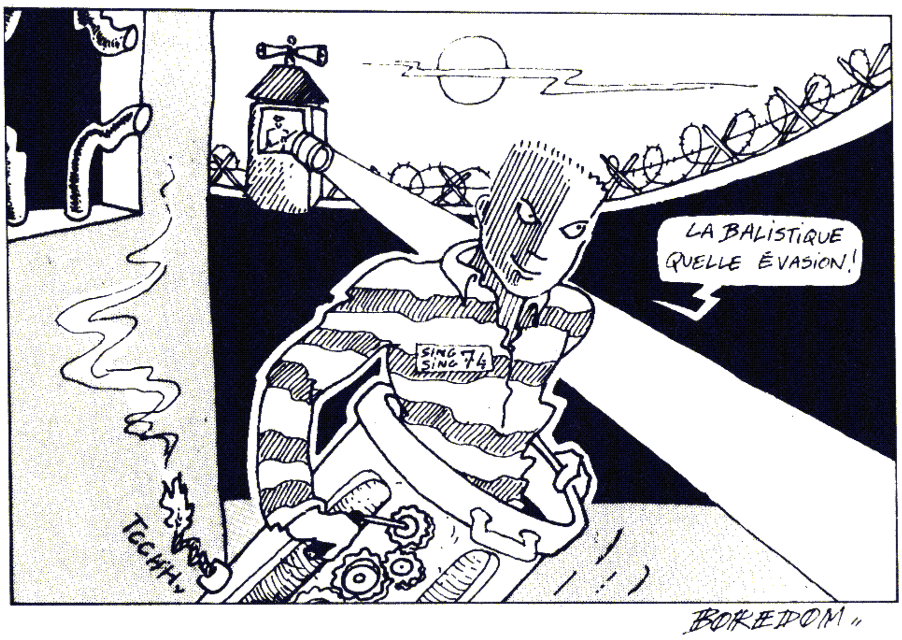
Le deuxième jeu est beaucoup moins stressant : il faut loger une bille dans un trou creusé dans le sol, en la lançant à la vitesse convenable. Et même si elle n'a pas été lancée suffisamment fort, il reste encore une chance de gagner ! En effet, dans ce cas, elle rebondit en arrivant au sol, et après plusieurs sauts, elle peut très bien tomber dans le but.

Quelques aspects théoriques doivent être abordés, à l'intention des curieux. Pour le premier jeu, l'équation de la trajectoire du projectile, qui sera appelée *équation 1*, s'écrit :
$z = (-g/2v^2\cos^2\alpha) + (\tan\alpha)x$
où g représente l'intensité de la pesanteur (10 m/s²), v la vitesse d'éjection du projectile (25 m/s), et α l'angle de tir.

La "portée" s'obtient en faisant $z=0$ dans l'*équation 1*, ce qui donne :
$P = (v^2\sin 2\alpha)/g$
On voit que P est maximum quand $\sin 2\alpha = 1$, c'est-à-dire quand $\alpha = 45$degrés.

Au cours des parties, on observe que certaines positions de la cible la mettent hors d'atteinte du projectile, quel que soit l'angle choisi (voir figure 1b). Ceci s'explique en faisant intervenir la notion de "parabole de sûreté". L'*équation 1*, en effet, s'écrit encore :
$\tan^2\alpha - 2(v^2/gx)\tan\alpha + ((2v^2/gx^2)z + 1) = 0$

Il s'agit alors d'une équation du second degré en "$\tan\alpha$", et trois cas peuvent se présenter selon le signe du discriminant :
- s'il est positif, la cible de coordonnées (x,z) est atteinte par deux angles de tir (les racines de cette équation) ;
- s'il est négatif, la cible est inaccessible, l'équation n'ayant pas de racines réelles ;
- s'il est nul, il n'existe qu'un seul angle de tir pour atteindre la cible.



Avant et après

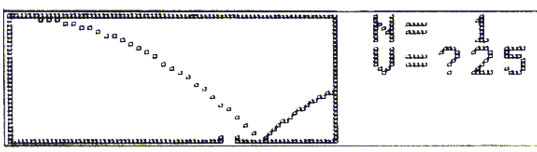
Le dernier cas (discriminant nul) permet d'écrire l'équation de la "parabole de sûreté", dans le plan. Pour le second jeu, à chaque rebond, la bille perd une partie de son énergie cinétique. Le coefficient K (ligne 15 du programme 2) représente le rapport des énergies cinétiques avant et après chaque contact de la bille avec le sol. Théoriquement, le nombre de rebonds est infini, mais dans la pratique, la bille s'arrête assez rapidement.

Un autre jeu consiste à calculer les distances totales parcourues par la bille dans les deux sens, vertical et horizontal !

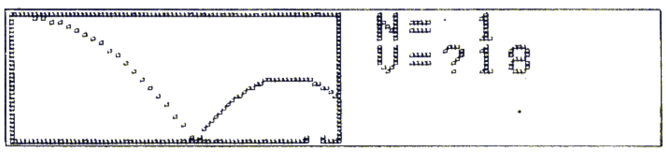
Trung HUA-NGOC

Figure 2
Exécution du programme 2.
La cible est le petit espace à la base du rectangle, qui représente un trou dans le sol

a. La vitesse initiale est trop forte



b. La vitesse initiale est trop faible



c. C'est gagné... grâce au rebond