

ION DIAMANDI



ION DIAMANDI

JOCURI PE CALCULATOR

Editura GETIC

BUCURESTI 1991

Cuvînt înainte

În orice societate jocurile îndeplinesc nenumărate roluri. Pentru a enumera doar cîteva dintre ele putem aminti formarea aptitudinilor, dezvoltarea abilității și a îndemînării, dezvoltarea gîndirii logice, algoritmice, anticipative și divergente. De asemenea este recunoscut și rolul social al jocurilor.

În universul jocurilor cele pe calculatoare ocupă un rol aparte reprezentînd, desigur, o tendință de mare viitor în care tehnologiile moderne se împletesc admirabil cu tradiția.

În practicarea jocurilor atît a celor clasice cît și a celor pe calculator rolul educatorului este esențial în organizarea propriu-zisă a jocului și în integrarea sa într-un context de învățare.

În prezent în țara noastră circulă multe jocuri pe calculatoare fiind practicate în special de către copii dar și de către adulți. Însă, marea majoritate a acestor jocuri nu sînt documentate, utilizatorul neștiind nici modul de conducere a jocului iar, cîteodată, nici măcar obiectivul jocului.

Lucrarea de față își propune tocmai acest scop, și anume, de a asigura un minim de informații pentru utilizarea jocurilor pe calculator.

Un alt aspect important este legat de învățarea realizării de către utilizatori a unor jocuri proprii. În acest scop se dau numeroase exemple de jocuri, de diverse tipuri: de îndemînare și reflexe, logice, de aventuri, pentru dezvoltarea unor aptitudini pentru școală. Se insistă în special asupra algoritmilor care se utilizează în multe din jocuri: algoritmi de căutare, algoritmi de sortare, algoritmul arborelui binar și multe altele. În paralel se învață tehnicile de bază pentru dezvoltarea jocurilor: tehnici pentru grafică, pentru animație, pentru

realizarea de caractere grafice, pentru realizarea de sunete pe calculator, etc.

Jocurile descrise sînt funcționale pe calculatoare compatibile cu tipul Sinclair Spectrum (HC, TIM, JET, CIP, COBRA) cele mai răspîndite calculatoare personale în țara noastră, fiind realizate în limbaj BASIC (acesta fiind cel mai inteligibil pentru majoritatea utilizatorilor calculatoarelor personale). S-a căutat să se utilizeze cît mai mult instrucțiunile BASIC standard astfel încît jocurile să poată fi transpuse cu efort minim și pe alte calculatoare. În unele listinguri ale programelor caracterele grafice definite de utilizator apar chiar sub forma lor grafică, fiind explicat în text modul lor de obținere (tastele), în timp ce, pentru altele se folosește în listing o reprezentare specială care diferă de linia de instrucțiune ce apare afișată pe ecran. De exemplu, forma din listing <CAPS A> se traduce prin acționarea tastei A în modul grafic.

În finalul lucrării sînt date cîteva sfaturi în legătură cu întreținerea jocurilor pe casete magnetice descriindu-se și cîteva programe utilitare de copiere a programelor.

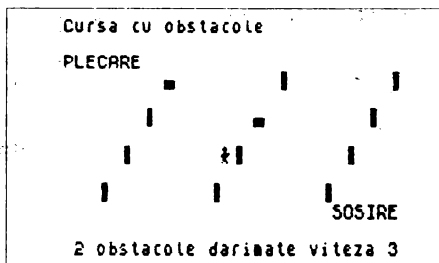
JOCURI DE INDEMINARE

Cursa cu obstacole

Un joc care se poate juca numai cu o singură tastă iar aceea poate fi oricare de pe tastatură (în afară de cele două taste de control CAPS SHIFT și SYMBOL SHIFT). La începutul jocului se solicită introducerea vitezei alergătorului (care reprezintă de fapt gradul de dificultate al jocului), un număr de la 1 la 10. Apoi pe ecran apare parcursul cu cele 12 obstacole, așezate câte 3 pe 4 linii drepte, distanța dintre obstacole fiind egală, indicîndu-se locul de plecare și cel de sosire.

În partea de jos a ecranului (sub parcurs) apare un text care indică permanent numărul de obstacole dărîmate și viteza alergătorului. Se dă un semnal sonor și alergătorul pornește din partea stîngă cu o viteză proporțională cu valoarea introdusă de jucător în prealabil. Acționînd o tastă la momentul potrivit, alergătorul va sări peste obstacol. Dacă însă tasta se va acționa prea devreme sau prea tîrziu, obstacolul va fi dărîmat. Cursa terminată, jocul se poate relua (eventual de un alt jucător cu comanda RUN). Pentru ca jocul să se reia automat se poate introduce linia:

235 RUN.



```
5 BORDER 6: PAPER 7: INK 9
10 PRINT "Cursa cu obstacole"
20 GO SUB 1200
30 LET n0=0: LET n=0: LET t=0
40 PRINT AT 12,1;0;" obstacol
darimat viteza ";v
50 GO SUB 1000
55 PRINT AT 2,0;"PLECARE"
57 PRINT AT 10,24;"SOSIRE"
65 BEEP 1,5
70 FOR l=3 TO 9 STEP 2
80 FOR c=0 TO 31 STEP 2
90 PRINT AT 1,c;"(q)"
100 PAUSE d: PRINT AT 1,c;" "
110 LET k=0: LET n=n+2
120 IF t=1 THEN GO TO 140
130 IF n=10 THEN IF INKEY$(("<"))""
THEN LET l=l-1: LET k=1
140 LET t=0: IF n("<")10 AND INKEY
$(("<"))"" THEN LET t=1
150 PAUSE d: PRINT AT 1,c+1;"(p
)"
160 IF n=10 THEN GO SUB 1300
170 IF k=0 AND n("<")0 THEN PAUSE
d: PRINT AT 1,c+1;" "
180 IF k=1 THEN PAUSE d: PRINT
AT 1,c+1;" "; LET l=l+1
200 NEXT c
210 NEXT l
230 BEEP .2,13: BEEP .2,16
240 STOP
1000 DATA 8,28,61,90,24,164,66,1
1010 DATA 16,56,16,58,84,48,40,6
8
1020 FOR s=80 TO 81
1030 FOR i=0 TO 7
1040 READ a: POKE USR CHR$ s+i,a
1050 NEXT i
1060 NEXT s
1100 FOR l=3 TO 9 STEP 2
1105 PRINT AT 1,12-l;
1110 FOR h=1 TO 3
1120 PRINT "(CAPS 5) ";
1130 NEXT h
1135 NEXT l
1150 RETURN
```

Modificare sugerată:

O cursă de obstacole cu doi alergători fiecare condus de către un jucător.

Descrierea programului

20 GO SUB 1200 - apelarea subrutinei de introducere a vitezei alergătorului și de validare a acesteia (un număr cuprins între 1 și 10). Pentru viteză se folosește variabila v. În linia 1230 a subrutinei se atribuie variabilei d (exprimă durata de parcurgere a distanței dintre doi "pași") valoarea 11-v. De fapt valoarea d reprezintă cât timp "stă" alergătorul pînă ce face următorul "pas", cînd va fi șters de pe vechea poziție și desenat pe următoarea.

Linia 30 - inițializarea variabilelor:

n0 reprezintă numărul de obstacole dărimate;

n reprezintă numărul de "pași" efectuați după trecerea ultimului obstacol,

t specifică dacă s-a acționat prea devreme o tastă (1 = da).

La începutul jocului toate aceste variabile au valoarea 0.

Liniile 40,55,57 - tablou de afișaj.

50 GO SUB 1000 - apelarea subrutinei de definire a caracterelor grafice corespunzătoare alergătorului (1000-1060) și a traseului cu obstacole (1100-1135).

```

1200 BEEP .2,20
1210 INPUT "Introduceți viteza(  
1 și 10) ";v
1220 IF v<1 OR v>10 THEN GO TO 1  
200
1230 LET d=11-v
1240 RETURN
1300 LET n=0
1310 IF k=1 THEN RETURN
1320 BEEP .1,30
1330 LET no=no+1
1340 PRINT AT 1,c+1;"<CAPS 3>"
1350 PRINT AT 12,1;no
1360 IF no=2 THEN PRINT AT 12,11  
;"e darimate"
1380 RETURN
    
```

Obstacolele sînt aranjate pe 4 linii drepte care corespund liniilor 3,5,7 și 9 ale ecranului grafic (vezi linia de program 1100 în care variabila l reprezintă linia). Liniile de program 1105 și 1120 servesc la desenarea obstacolelor. Se observă din linia 1105 că pe fiecare din cele 4 linii obstacolele sînt așezate în alt mod (și anume pe coloana ecranului grafic 12-l)

Din linia 1120 se poate remarca distanța egală dintre obstacole și anume de 9 celule caracter. Caracterul grafic pentru obstacol nedărîmat se obține acționînd tasta 5 (cu CAPS SHIFT) în modul grafic.

65 BEEP 1,5

- semnalul sonor de start.

70-210 - ciclul de parcurs al celor 4 linii drepte.

80-210 - ciclul de parcurs al unei linii (este o linie a ecranului grafic de 32 de celule - poziții caracter -, de la 0 la 31). Alergătorul se deplasează din poziție în poziție, dar pentru a se sugera mișcarea, pe pozițiile corespunzătoare coloanelor cu numere pare alergătorul va avea alt aspect față de alergătorul afișat pe pozițiile

corespunzătoare coloanelor de numere impare. Din această cauză în linia 80 apare STEP 2: alergătorul se afișează pe poziția corespunzătoare liniei l și coloanei c dar din două în două coloane (pe coloanele cu număr par).

90 - afișarea alergătorului pe poziția corespunzătoare liniei l și coloanei c (cu număr par). Caracterul "alergător" (de pe coloană cu număr par) se obține prin acționarea tastei Q în modul grafic.

100 - pauză care servește la reglarea vitezei. Dacă s-a introdus inițial o viteză $v = 1$, atunci pauza între doi "pași" va fi $d = 11 - v = 10$ adică circa o cincime de secundă (PAUSE 10). După pauză se va șterge caracterul alergătorului de pe poziție pentru afișarea lui pe coloana următoare.

110 - variabilă k, indică dacă alergătorul are de gând să "sară" sau nu ($k = 0$ nu sare - la începutul jocului -, $k = 1$ sare). Numărul de "pași" n efectuați după ultimul obstacol crește cu 2.

120 - dacă s-a apăsat prea devreme pe o tastă.

130 - alergătorul a ajuns la obstacol (dacă $n = 10$ înseamnă că a parcurs 10 "pași" de la ultimul obstacol). În acest caz, dacă se acționează o tastă saltul va avea loc. În acest scop numărul liniei l va scădea cu o unitate (caracterul alergătorului urmează să se afișeze pe o linie mai sus, deasupra obstacolului) iar k se va face 1, indicație că alergătorul va "sări". Dacă nu se acționează nici o tastă se va trece la linia următoare (140).

140 - Variabila t devine 0 și imediat este readusă la 1 dacă s-a acționat o tastă înainte de a se fi ajuns la un obstacol.

150 - pauză pentru a se trece la poziția următoare și se afișează caracterul alergător (de pe coloană cu număr impar) pe poziția următoare. Acest caracter se obține prin acționarea tastei P în modul grafic.

160 - apelarea subrutinei de dărîmarea a obstacolului (GO SUB 1300). La dărîmarea obstacolului numărul de "pași" după trecerea de

ultimul obstacol n devine 0 (linia 1300), iar numărul de obstacole dărîmate n0 crește cu 1 (linia 1310). Acest număr (scorul) se afișează imediat după căderea obstacolului (linia 1350). In locul obstacolului "în picioare" se afișează unul dărîmat care se obține acționînd tasta 3 (cu CAPS SHIFT) în modul grafic (linia 1340). Dărîmarea obstacolului este însoțită de un sunet (linia 1320).

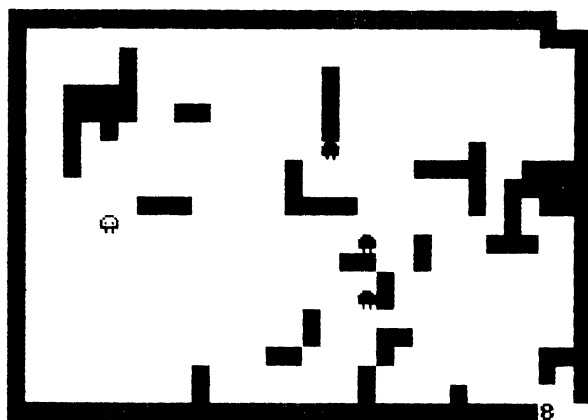
170 - dacă alergătorul nu sare ($k=0$) și nu se găsește deasupra unui obstacol atunci se șterge de pe poziția respectivă.

180 - dacă alergătorul sare atunci se șterge de pe poziția respectivă și se face pregătirea pentru "coborîrea" lui pe linia inferioară (numărul liniei grafice l crește cu o unitate).

230,240 - sfîrșitul jocului.

Labirint

Pe ecran apare un labirint, dar nu se pune problema de a găsi o ieșire din labirint, ci, de a reuși să se conducă personajul "bun" astfel



încît să scape de voracitatea unor creaturi "rele". Se stabilește inițial numărul "răilor" (adversarilor) prin introducerea răspunsurilor la întrebarea pusă de calculator. Pot fi unul, doi sau trei "răi". Apoi începe cursa, care este contra cronometru. Jucătorul poate deplasa personajul "bun" orizontal sau vertical cu tastele: 5 - pentru stînga, 6 - pentru jos, 7 - pentru sus și 8 - pentru dreapta. "Răii" cunosc poziția "bunului" dar se deplasează într-un mod puțin dezordonat, mai ales atunci cînd se lovesc de pereții labirintului. Ei pot avansa o poziție sau chiar două, pe orizontală, dar nu pe diagonală, putînd "sări" peste pereții subțiri. De aceea nu este indicat să se staționeze prea mult ascuns după pereți mai cu seamă că, numărul de pași făcuți deja contează, fiind afișat la sfîrșitul jocului, fie că ați fost "mîncat" de "răi", fie că ați reușit să scăpați pînă la sfîrșitul timpului regulamentar. În partea dreaptă jos a ecranului este afișat în permanență cronometrul care marchează timpul.

Modificări propuse:

Se poate încerca și varianta clasică a labirintului în care acesta va avea o ieșire, încercîndu-se ieșirea din labirint în timpul regulamentar. Versiunea propusă se poate complica, introducîndu-se în labirint și niște "obiecte" pe care personajul bun va trebui să le culeagă în număr cît mai mare. În acest caz "răii" vor fi de fapt niște paznici, iar scopul jocului va fi culegerea cît mai multor obiecte. De asemenea, pe măsura scurgerii timpului, labirintul își poate restrînge posibilitățile de parcurs, îngreunînd sarcina jucătorului.

În sfîrșit, altă variantă ar fi existența a două tipuri de "obiecte" de cules, pentru unele acordîndu-se mai multe puncte. În această situație, jucătorul va trebui să încerce să-și optimizeze parcursul prin labirint.

Descrierea jocului

10 GO SUB 3000 - apelarea subrutinei de definire a caracterelor grafice pentru "bun" și pentru "răi".

3000 - datele pentru caracterul grafic "bun".

3010 - datele pentru caracterul grafic "rău".

3020-3070 - ciclul de citire și memorare a datelor care definesc caracterele "bun" și "rău".

15 GO SUB 1000 - apelarea subrutinei de trasare a labirintului.

1000 - pereții vor fi verzi.

1005-1130 - ciclul pentru calculul locului și lungimii pereților. Acestea vor fi întâmplătoare.

1020-1030 - calculul extremității peretelui (întâmplătoare, dar între 1 și 30 pe orizontală și 1 și 20 pe verticală).

1040 - perete vertical: calculul celeilalte extremități a peretelui, cu limită la 20, lungimea cuprinsă între 1 și 3.

1050 - același lucru pentru perete orizontal.

1070-1110 - ciclul pentru trasarea peretelui.

1140 - trasare perete margine superioară. Se folosește tasta 8 împreună cu CAPS SHIFT în mod grafic.

1150 - trasare perete margine laterală.

1170 - trasare perete margine inferioară.

20 - rezervare de memorie pentru fișierul pozițiilor celor 3 "răi" (variabila c - pentru coloană, variabila l - pentru linie).

25 - introducerea numărului de răi, ng.

26 - refuzul răspunsurilor defectuase.

30 GO SUB 900 - apelarea subrutinei pentru afișarea "răilor" și "bunului". La începutul jocului aceștia pornesc din aceleași poziții fixe.

900 - poziția orizontală (z) și verticală (y) a bunului.

905 - numărul de pași. (pas). La început este 0 (nu s-a făcut nici o mișcare).

Labirint

```

5 BORDER 6: PAPER 7: INK 4
7 CLS
8 PRINT "LABIRINT"
10 GO SUB 3000
15 GO SUB 1000
20 DIM c(3): DIM l(3)
25 BEEP 0.2,20: INPUT "1,2 sau
3 rai ";ng
26 IF ng<1 OR ng>3 THEN GO TO
25
27 LET ng=INT ng
30 GO SUB 900
32 BEEP 3,5: BEEP 0.2,10
40 FOR s=0 TO 96 STEP ng+1
50 GO SUB 500
60 FOR g=1 TO ng
65 GO SUB 700
70 GO SUB 500
75 NEXT g
80 PRINT AT 21,29;s
85 NEXT s
95 BEEP 1,15
100 PRINT AT 21,0;"Salvat de go
ng! ";pas;" pasi"
110 FOR d=1 TO 200: NEXT d
120 INK 0
130 STOP
500 LET f$=INKEY$
510 IF f$="" THEN RETURN
530 LET dx=0: LET dy=0
535 IF f$("5" OR f$)"8" THEN RE
TURN
540 GO TO 540+CODE f$-48
545 LET dx=-1: GO TO 580
546 LET dy=1: GO TO 580
547 LET dy=-1: GO TO 580
548 LET dx=1: GO TO 580
580 LET xn=x+dx
590 LET yn=y+dy
610 IF xn<0 OR xn>31 THEN RETUR
N
620 IF yn<0 OR yn>21 THEN RETUR
N
630 IF POINT (xn*8,(21-yn)*8)=1
THEN RETURN
634 PRINT AT y,x;" "
635 LET pas=pas+1
640 LET x=xn
650 LET y=yn
660 INK 0: PRINT AT y,x;"(a)"
670 RETURN
700 LET dx=SGN (x-c(g))*INT (RN
D*3)
710 LET dy=SGN (y-l(g))*INT (RN
D*3)
730 LET xgn=c(g)+dx
740 LET ygn=l(g)+dy
750 IF xgn<0 OR xgn>31 THEN RET
URN
760 IF ygn<0 OR ygn>21 THEN RET
URN
770 IF POINT (xgn*8+4,(21-ygn)*
8+4)=1 THEN RETURN
775 PRINT AT 1(g),c(g);" "
780 LET c(g)=xgn
790 LET l(g)=ygn
800 INK g: PRINT AT 1(g),c(g);"
(b)"
810 IF x<>c(g) THEN RETURN
820 IF y<>l(g) THEN RETURN
830 BEEP 0.2,10: BEEP 0.3,5: PR
INT AT 21,0;"Ai fost mincat din
";pas;" pasi!";
840 FOR d=1 TO 200: NEXT d
860 INK 0
870 STOP
900 LET x=5: LET y=11
905 LET pas=0
910 INK 0: PRINT AT y,x;"(a)"
920 FOR g=1 TO ng
940 LET c(g)=20: LET l(g)=INT (
16*g/ng)
950 INK g: PRINT AT 1(g),c(g);"
(b)"
960 NEXT g
970 RETURN
1000 INK 4
1005 FOR t=1 TO 30
1010 LET k=INT (RND*2)
1020 LET x1=INT (RND*30)+1
1030 LET y1=INT (RND*20)+1
1040 IF k=0 THEN LET x2=x1: LET
y2=y1+INT (RND*3)+1: IF y2>20 TH
EN LET y2=20
1050 IF k=1 THEN LET y2=y1: LET
x2=x1+INT (RND*3)+1: IF x2>30 TH
EN LET x2=30
1070 FOR x=x1 TO x2
1080 FOR y=y1 TO y2
1090 PRINT AT y,x;"(CAPS 8)"
1100 NEXT y
1110 NEXT x
1130 NEXT t

```


530 - mărimea inițială a deplasării pe orizontală (dx) și pe verticală (dy).

540 - saltul la o linie de program în funcție de tasta acționată.

545 - pregătirea deplasării spre stînga.

546 - pregătirea deplasării în sus.

547 - pregătirea deplasării în jos.

548 - pregătirea deplasării spre dreapta.

580-590 - calculul viitoarei poziții orizontale și verticale.

610-620 - respingerea posibilității de ieșire din labirint (ecran).

630 - respingerea posibilității de traversare a unui perete.

634 - ștergerea bunului.

635 - a fost efectuat un pas; numărul pașilor (pas) crește cu o unitate.

640-650 - poziția viitoare devine poziția actuală.

660 - afișarea "bunului" în poziția actuală. Pentru "bun" se folosește tasta A în mod grafic.

700 - 870 - subrutină de apelare a unui "rău".

700-710 - determinarea deplasării orizontale și verticale în direcția "bunului", lăsînd o parte de hazard: 0, 1 sau 2 căsuțe (celule caracter) cu posibilitate de salt.

730-740 - viitoarea poziție a "răului" (xgn - coordonata pe orizontală, ygn - coordonata pe verticală).

750-760 - respingerea posibilității de ieșire din labirint (ecran).

770 - respingerea posibilității de aterizare pe un perete.

775 - ștergerea "răului" din poziția sa actuală.

780-790 - memorarea noii poziții.

800 - afișarea "răului" cu albastru, roșu sau purpuriu, în funcție de numărul său, la noua poziție. Pentru caracterul grafic "rău" se folosește tasta B în modul grafic.

810-820 - dacă "bunul" nu se găsește în același loc cu "răul", jocul continuă.

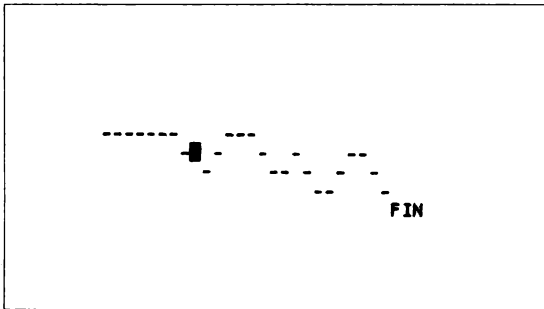
830 - "bunul" se găsea în acest loc; afișarea mesajului și a numărului de pași.

840 - ciclu pentru neutralizarea acțiunilor de taste care intervin după oprirea jocului.

860-870 - sfișit.

Firul Ariadnei

Iată și un scurt joc de îndemînare: pierdut într-un labirint va trebui să urmăriți un fir salvator care vă va conduce spre ieșire. Cu ajutorul tastelor 6 și 7 va trebui să urmăriți cu fidelitate drumul trasat pînă cînd veți ajunge la litera F (Final). Jocul are 10 niveluri de dificultate (nivel 1: deplasare lentă, nivel 10: deplasare foarte rapidă).



Descrierea programului

Programul principal: liniile 1-90:

- linia 10: apelarea subrutinei de prezentare;
 - linia 20: apelarea subrutinei de inițializare;
 - linia 30: apelarea subrutinei jocului;
 - linia 40: afișarea clipitoare a mobilului la sfișitul jocului; apelarea unei subrutine sonore;
-

```

1 REM *****
2 REM *Firul Ariadnei*
3 REM *****
10 GO SUB 9000
20 GO SUB 8000
30 GO SUB 5000
40 PRINT AT y,x; BRIGHT 1; FLASH 1; INK 2;");": GO SUB 9080
50 IF fin=1 THEN PRINT AT 16,5; FLASH 1;"Te-ai ratacit!!": GO TO 70
60 PRINT AT 16,3; FLASH 1;"Bra va esti salvat!!"
70 PAUSE 4E4: PAUSE 4E4
80 CLS
90 GO TO 20
5000 PRINT AT y,x;"--"
5010 LET x=x+1
5020 LET a$=INKEY$
5030 LET y=y+(a$="6")-(a$="7")
5040 LET k$=SCREEN$(y,x)
5050 PRINT AT y,x; INVERSE 1; INK 2;");"
5060 FOR j=1 TO 10*(10-n): NEXT j: BEEP 0.01,10
5070 IF k$="F" THEN LET fin=0: GO TO 5100
5080 IF k$="--" THEN GO TO 5000
5090 LET fin=1
5100 RETURN
8000 INPUT "Nivelul de joc (1-10)?:";n
8010 LET n=INT (ABS n)
8020 IF (n-1)*(n-10)>0 THEN GO TO 8000
8030 CLS
8040 PRINT AT 10,0;"-----"
8050 LET y=10
8060 FOR x=5 TO 25
8070 PRINT AT y,x;"--"
8080 LET y=y+INT (RND*3)-1
8090 IF y<5 THEN LET y=5
8100 IF y>15 THEN LET y=5
8110 NEXT x
8120 PRINT AT y,x;"FIN"
8130 LET x=0: LET y=10
8140 RETURN
9000 BORDER 0: PAPER 0: INK 4: CLS

```

- linia 50: cazul în care nu ați reușit;

- linia 60: cazul în care ați fost salvat;

- linia 70: așteptarea acționării unei taste pentru continuare;

- linia 90: reîntoarcerea la începutul jocului.

Subrutina de joc : liniile 5000-5100:

- linia 5000: desenarea drumului după trecerea mobilului;

- linia 5010: avans orizontal;

- linia 5020: citire tastatură;

- linia 5030: calculul deplasării pe verticală;

- linia 5040: salvarea caracterului situat pe viitoarea poziție de afișaj;

- linia 5050: afișarea mobilului;

- linia 5060: ciclul de temporizare a cărei durată depinde de jocul ales;

- linia 5070: ieșire victorioasă din labirint;

- linia 5080: verificarea dacă mobilul se află pe drum;

- linia 5090: dacă mobilul a pierdut firul variabila fin ia valoarea 1;

- linia 5100: întoarcerea în programul principal.

Subrutina de inițializare: liniile 8000-8140:

- liniile 8000-8020: alegerea nivelului de joc;

- liniile 8040-8120: afișarea drumului de parcurs avînd grijă ca la început acesta să fie în linie dreaptă (linia 8040);

- linia 8130: coordonatele poziției de plecare;

- linia 8140: întoarcerea la programul principal.

Subprogramul de prezentare: liniile 9000-9120:

- liniile 9030-9070: deplasarea unui mobil în lungul drumului desenat în linia 9020;

- liniile 9080-9110: rutină sonoră;

- linia 9120: întoarcerea la programul principal.

```

9010 PRINT AT 10,10;"FIRUL ARIAD
NEI"
9020 PRINT AT 20,0;"-----
-----FIN"
9030 FOR I=0 TO 25

9040 PRINT AT 20,I; BRIGHT 1; FL
ASH 1; INK 2;">"
9050 BEEP 0.01,20
9060 PRINT AT 20,I;"-"
9070 NEXT I
9075 PRINT AT 20,I; BRIGHT 1; FL
ASH 1; INK 2;">"
9080 FOR I=10 TO 30
9090 BEEP 0.01,2*I-20
9110 NEXT I
9120 RETURN
    
```

Robac

După apăsarea unei taste, pe ecran apare o rîmă de dimensiuni mai mari - ROBAC. Acesta va trebui condus pe suprafața de joc marcată cu o culoare mai deschisă cu scopul de a mîncă cît mai mulți păienjeni (gîndaci). Sînt 4 feluri de păienjeni care apar în mod întîmplător pe ecran în timpul jocului, iar pentru fiecare păianjen mîncat de ROBAC, jucătorul primește 10 puncte.

Robac

Deplasarea lui ROBAC se va realiza prin intermediul tastelor:

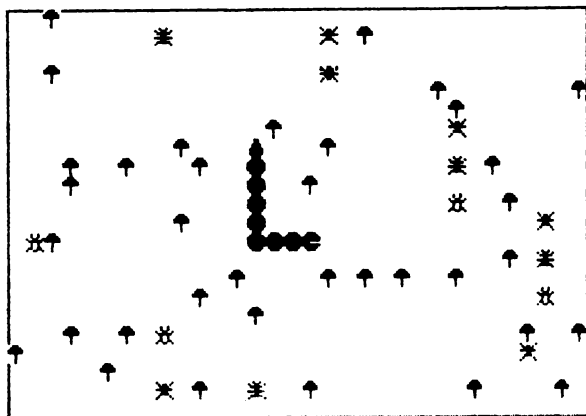
A - deplasare în jos;

Q - deplasare în sus;

O - la stînga;

P - la dreapta.

În timpul jocului (tot întîmplător) apar pe ecran și ciuperci otrăvitoare care îl omoară pe ROBAC, dacă acesta le mănîncă. În acest caz, ROBAC, va pierde o viață (are în total 5 vieți) și jocul se va relua de la început însă cu punctajul acumulat pînă în acel moment și cu viețile care i-au mai rămas la dispoziție. După terminarea jocului (pierderea tuturor celor 5 vieți) jocul se va putea relua (eventual de alt jucător) prin acționarea oricărei taste, de data aceasta, însă punctajul va porni de la 0. Atingerea unei margini a suprafeței de joc



va marca o pierdere de 10 puncte, iar în acest caz, dacă nu se va acționa o tastă pentru identificarea direcției de deplasare se vor pierde puncte în continuare și jocul se va pierde definitiv (cînd se ajunge la un punctaj negativ). El se poate relua de la început prin acționarea oricărei taste.

În partea de jos a ecranului apare afișat numărul de vieți a lui ROBAC: cu cât acesta va fi de dimensiuni mai reduse (mai scurt) cu atât numărul de vieți care mai rămân la dispoziție este mai mic. Tot în partea de jos a ecranului (dreapta) apare afișat și numărul de puncte realizat.

Modificări posibile:

Jocul se poate modifica prin următoarele adăugări sau modificări:

- introducerea nivelelor de dificultate. Un nivel de dificultate sporit va însemna apariția mai rapidă a ciupercilor față de păianjeni și/sau deplasarea mai rapidă a lui ROBAC;

- acordarea de puncte diferențiat în funcție de tipul păianjenului. În acest caz jucătorul va trebui să-și optimizeze traseul, încercînd să-l determine pe ROBAC să mănînce mai repede păianjenii mai "prețioși";

- la atingerea marginilor suprafeței de joc, se vor pierde mai multe puncte (toate sau o parte din ele);

- creșterea dificultății jocului pe măsură ce se pierd din vieți: apariția mai rapidă a ciupercilor față de păienjeni.

Pe aceeași idee se poate modifica și scenariul jocului ROBAC:

În loc de o rîmă poate fi un personaj, de exemplu Făt Frumos (caracter grafic pentru om), care culege mere fermecate sau omoară balauri care apar din loc în loc pe ecran. Pe Făt Frumos îl poate urmări un zmeu care, dacă îl prinde îl omoară. În acest caz, spre deosebire de ROBAC, în care ciupercile apar pe ecran sau stau pe loc, va trebui realizată și deplasarea zmeului care îl urmărește pe Făt Frumos.

Descrierea programului

10 GO SUB 6000 - apelarea subrutinei pentru descrierea lui ROBAC.

6010 - citirea datelor pentru descrierea lui ROBAC (linia 9700).

```

7 PRINT AT 10,8;"Asteapta un
pic..."
10 GO SUB 6000
20 DIM p(3,10)
30 INK 6: BORDER 4: PAPER 6
40 CLS
50 PRINT #1;"Apasa o tasta"
60 PAUSE 0
80 CLS
90 LET a$="(sttu sttu sttu)"
100 LET puncte=0
110 CLS
120 PRINT #1;AT 1,0; PAPER 7;"
;#2
130 GO SUB 5000
140 LET paian=0
150 LET cap=9
160 LET coada=1
170 RESTORE 9500
180 FOR i=1 TO 9
190 READ p(2,i),p(1,i),p(3,i)
200 NEXT i
210 LET x=1
220 LET y=0
230 LET x1=(INKEY$="p")-(INKEY$
="o")
240 LET y1=(INKEY$="a")-(INKEY$
="q")
250 IF x1(>0) THEN LET x=x1: LET
y=y0
260 IF y1(>0) THEN LET y=y1: LET
x=x0
270 LET x2=p(1,cap)+x
280 LET y2=p(2,cap)+y
290 IF x2>31 OR x2<0 OR y2>21 O
R y2<0 THEN GO TO 295
293 GO TO 300
295 BEEP .1,-10: LET puncte=pun
cte-10: IF puncte<0 THEN GO TO 1
079
298 GO TO 390
300 LET cap=cap+1
310 IF cap>10 THEN LET cap=1
320 LET coada=coada+1
330 IF coada>10 THEN LET coada=
1
340 LET p(1,cap)=x2
350 LET p(2,cap)=y2
360 LET p(3,cap)=x+2*y+3
370 IF ATTR (y2,x2)=48 THEN LET
paian=1
380 IF ATTR (y2,x2)=50 THEN GO
TO 1000
390 GO SUB 3000
400 IF paian THEN GO SUB 4000
410 GO SUB 2000
420 PRINT AT p(2,cap),p(1,cap);
INK 1;q$(p(3,cap));
430 GO TO 230
1010 FLASH 1
1020 GO SUB 3000
1030 FLASH 0
1040 LET a$a$(5 TO )+"
1050 GO SUB 5000
1060 BEEP 2,-10
1070 IF a$(">)"
" THE
N GO TO 110
1080 CLS
1090 INK 2
1100 PRINT AT 5,11;"AI MURIT!";A
T 10,10-LEN (STR$ puncte)/2;"PUN
CTE : ";puncte
1110 RESTORE 9600
1120 FOR I=1 TO 2
1130 FOR J=0 TO 9
1140 READ timp,ton
1150 BEEP timp,ton
1160 NEXT J
1170 PAUSE 20
1180 NEXT I
1190 PRINT AT 20,9;"tasteaza ENT
ER"
1200 PAUSE 0
1210 GO TO 30
2010 RANDOMIZE
2020 LET ik=2
2030 LET s$="(r)"
2040 IF RND(.3) THEN LET s$=bs(IN
T (4*RND)+1): LET ik=0
2050 LET tx=INT (21*RND)
2060 LET ty=INT (32*RND)
2070 IF ATTR (tx,ty)=48 THEN RET
URN
2080 IF ATTR (tx,ty)<>54 THEN LE
T ik=7: LET s$=" "
2090 PRINT AT tx,ty; INK ik;s$;
2100 RETURN

```

6020-6050 - definirea caracterelor grafice pentru corpul lui ROBAC (guri, cozi) și a păienjenilor.

6060 - variabila r\$ este o variabilă șir de caractere care conține toate caracterele grafice pentru gura deschisă. Aceste caractere se obțin astfel: gură deschisă în sus - tasta A în mod grafic; gură deschisă spre stînga - tasta B în modul grafic; gură deschisă spre dreapta - tasta C în modul grafic; gură deschisă în jos - tasta D în modul grafic.

6070 - variabila g\$ este o variabilă șir de caractere care conține toate caracterele grafice pentru gură închisă. Aceste caractere se obțin astfel: gură închisă în sus - E în modul grafic; gură închisă spre stînga - F în modul grafic; gură închisă spre dreapta - G în modul grafic; gură închisă în jos - H în modul grafic. 6080 - variabila o\$ este o variabilă șir de caractere care conține toate caracterele grafice pentru coadă. Aceste caractere se obțin astfel: coada, cînd deplasarea se face spre stînga - J în modul grafic; coada, cînd deplasarea se face spre dreapta - K în modul grafic; coada cînd deplasarea se face în jos - L în modul grafic. 6090 - b\$ este o variabilă șir de caractere care conține toate caracterele grafice pentru păienjeni. Cele 4 feluri de păienjeni se obțin cu tastele N, P, O și Q în modul grafic.

20 - alocarea spațiului de memorie pentru descrierea lui ROBAC.

El poate avea o dimensiune maximă de 10 caractere grafice (2 cozi, 7 bucăți de corp și o gură sau, o coadă, 8 bucăți de corp și o gură).

90 - a\$ este o variabilă șir de caractere cu ajutorul căreia se va afișa ROBAC în partea de jos a ecranului. Aici el este de dimensiuni mai mici decît ROBAC-ul care se mișcă pe suprafața de joc. Caracterele grafice pentru obținerea ROBAC-ului mai mic se vor obține cu tastele S pentru coadă, T pentru o bucată de corp și U pentru cap,

```

3010 PRINT AT p(2, cap), p(1, cap);
      INK 1; r*(p(3, cap));
3020 LET a=cap
3030 FOR i=1 TO 7
3040 LET a=a-1
3050 IF a(1) THEN LET a=10
3060 PRINT AT p(2, a), p(1, a); INK
      1; "(m)";
3070 NEXT i
3080 PRINT AT p(2, coada), p(1, coa
      da); INK 1; o*(p(3, coada));
3090 LET a=coada-1
3100 IF a(1) THEN LET a=10
3110 PRINT AT p(2, a), p(1, a); " ";
3120 RETURN
4010 LET puncte=puncte+10
4020 BEEP .2,5; BEEP .1,10; BEEP
      .05,2
4030 GO SUB 5000
4040 LET paian=0
4050 RETURN
5010 PRINT #0; AT 1,0; INK 0; PAP
      ER 7; " ROBAC "; a$; " " ; AT 1,2
      1; " PUNCTE "; puncte; #2
5020 RETURN
6010 RESTORE 9700
6020 FOR i=USR "a" TO USR "u"+7
6030 READ a
6040 POKE i, a
6050 NEXT i
6060 LET r$="(ab cd)"
6070 LET q$="(ef gh)"
6080 LET o$="(ij kl)"
6090 LET b$="(nopq)"
6100 RETURN
9500 DATA 5,5,4,5,6,4,5,7,4,5,8,
      4,5,9,4,5,10,4,5,11,4,5,12,4,5,1
      3,4
9600 DATA .4,12,.3,14,.1,12,.4,9
      ,.4,9,.3,9,.1,7,.3,9,.1,10,.6,9
9610 DATA .4,0,.3,2,.1,0,.4,-3,.
      4,-3,.3,-3,.1,-5,.3,-3,.1,-2,.6,
      -3
9700 DATA 66,195,195,231,255,255
      ,126,60
9710 DATA 60,126,31,15,15,31,126
      ,60
9720 DATA 60,126,248,240,240,248
      ,126,60
9730 DATA 60,126,255,255,231,195
      ,195,66
9740 DATA 44,110,239,239,239,255
      ,126,60
9750 DATA 60,126,255,255,7,255,1
      26,60
9760 DATA 60,126,255,255,224,255
      ,126,60
9770 DATA 60,126,255,239,239,239
      ,110,44
9780 DATA 60,126,126,126,60,60,2
      4,24
9790 DATA 0,112,124,255,255,124,
      112,0
9800 DATA 0,14,63,255,255,63,14,
      0
9810 DATA 24,24,60,60,126,126,12
      6,60
9820 DATA 60,126,255,255,255,255
      ,126,60
9830 DATA 129,90,60,255,24,60,66
      ,129
9840 DATA 0,153,221,60,255,60,90
      ,129
9850 DATA 36,153,126,24,255,24,1
      26,129
9860 DATA 36,36,153,126,36,36,10
      2,153
9870 DATA 56,124,254,254,16,16,1
      6,16
9880 DATA 0,28,62,126,255,126,62
      ,28
9890 DATA 0,115,255,255,255,255,
      255,115
9900 DATA 0,144,254,252,248,252,
      254,144

```


toate în modul grafic. ROBAC-ul din partea de jos a ecranului va reprezenta numărul de vieți pe care le are ROBAC la un anumit moment dat. Inițial el este format din 3 bucăți; după pierderea unei vieți, o bucată va dispărea, apoi încă una, apoi o jumătate din ultima bucată etc.

100 - inițializarea cu 0 a variabilei pentru punctaj (puncte).

110 - începe pregătirea suprafeței de joc.

130 GO SUB 5000 - apelarea subrutinei pentru afișarea în partea de jos a ecranului a punctajului și a vieților.

140 - inițializarea variabilei paian. Aceasta este un indicator care arată dacă ROBAC a mâncat un păianjen (0 = nu, 1 = da).

150,160 - inițializarea variabilelor pentru mărimea lui ROBAC. Corpul (gura și corpul propriu-zis) este format din 9 părți (caractere grafice) - variabila cap, plus o parte pentru coadă (variabila coadă).

170 - ROBAC începe să circule. Mișcarea lui se realizează astfel: pornește dintr-un punct fix spre dreapta având 2 cozi, 7 bucăți de corp și gura deschisă; se șterge o coadă; se închide gura; se înlocuiește gura cu o bucată de corp; se pune o gură deschisă pe celula caracter din față; se șterge coada și ciclul se repetă.

210,220 - stabilirea coordonatelor.

230 - stabilirea direcțiilor în care este condus ROBAC. Dacă variabila x_1 este diferită de 0, mișcarea se va face pe orizontală (s-a acționat tasta 0 - pentru deplasare stînga - sau P - pentru deplasare dreapta), iar dacă variabila y_1 este diferită de 0, mișcarea se va face pe verticală (s-a acționat tasta Q - pentru deplasarea în sus - sau A - pentru deplasarea în jos).

270 - stabilirea coordonatelor lui ROBAC pentru deplasarea pe orizontală.

280 - stabilirea coordonatelor lui ROBAC pentru deplasarea pe verticală.

290 - testarea atingerii marginilor suprafeței de joc.

295 - se indică sonor atingerea marginii și se diminuează punctajul cu 10 puncte. Dacă acesta a devenit negativ jocul ia sfârșit (1080).

298 - dacă nu s-a atins marginea, se continuă înaintarea.

300-320 - avansarea capului și cozii.

340,350 - stabilirea direcției de deplasare.

370 - ce s-a aflat în fața lui ROBAC?

380 - în fața lui ROBAC a fost o ciupercă; se pierde o viață.

390 GO SUB 3000) - apelarea subrutinei pentru deplasarea lui ROBAC.

3010,3020 - se arată mai întâi capul.

3030,3070 - corpul. Caracterul pentru corp din linia 3060 se obține cu tasta M în modul grafic.

3080,3090 - coada.

400 GO SUB 4000 - în fața lui ROBAC a fost un păianjen.

4010 - punctajul crește cu 10 puncte.

4020 - note muzicale.

4030 - se apelează subrutina pentru afișarea din partea de jos a ecranului (punctaj și vieți).

4040 - variabila paian redevine 0.

410 GO SUB 2000 - apelarea subrutinei pentru afișarea pe ecran a ciupercilor sau păienjenilor.

2020 - apare o ciupercă.

2030 - caracterul grafic pentru ciupercă se obține cu tasta R în modul grafic.

2040 - apare un păianjen.

2080 - nu apare nimic.

430 - de la început pentru a se testa ce tastă s-a acționat pentru deplasarea lui ROBAC.

1010 - după ce a mâncat ciuperca, ROBAC va deveni pentru o secundă clipitor și va pierde o viață.

1040 - diminuarea lungimii lui ROBAC afișat în partea de jos a ecranului.

1050 - apelarea subrutinei pentru afișarea în partea de jos a ecranului.

1070 - se testează dacă mai sînt vieți. Dacă da, jocul se reia de la început cu punctajul actual (linia 110), dacă nu, se începe secvența de final de joc.

1100 - mesajul de sfîrșit de joc.

1120-1180 - melodia de sfîrșit de joc.

9500-9900 - date.

9500 - date ce reprezintă începutul drumului pentru ROBAC.

9600,9610 - date pentru notele muzicale.

9700-9790 - date pentru definirea caracterelor pentru gurile lui ROBAC.

9800-9890 - date pentru definirea caracterelor pentru cozi.

9900 - date pentru definirea caracterului cozii.

JOCURI LOGICE

Cuvinte încrucișate

Cu acest program veți introduce cuvinte pe care calculatorul le va aranja într-o grilă caroiată ale cărei dimensiuni depind de mărimea celui mai lung dintre cuvintele indicate. Apoi, calculatorul va umple restul pătrățelelor grilei cu litere alese la întâmplare. Rămîne ca un jucător să identifice cuvintele din grilă. Dacă acest lucru este dificil se poate introduce o opțiune prin care calculatorul va face ca pe grilă să apară cuvintele scrise cu culoarea fondului, acesta căpătînd culoarea cuvintelor (video invers).

Atenție! Cel mai lung cuvînt nu trebuie să depășească 18 litere (în acest caz grila ar depăși limitele ecranului). La introducerea cuvintelor trebuie ținut seama de faptul că 20 de cuvinte de lungime variabilă pot fi introduse în grilă într-un interval de timp cuprins între 5 și 10 minute. O listă mai lungă poate provoca un timp de așteptare prea mare. O bună idee constă în introducerea cuvintelor în ordinea descrescătoare a lungimii lor, viteza de execuție devenind astfel mai mare.

Descrierea programului:

5-60 - stabilirea atributelor de culoare folosite în timpul jocului și afișarea mesajelor de început de joc;

80 - introducerea numărului de cuvinte care se vor folosi (variabila a);

90 - introducerea celui mai lung cuvînt (b\$);

100 - dimensionarea unui vector bidimensional (c\$) care atașează fiecărui cuvînt lungimea maximă posibilă;

```

1 REM *****
  * CUVINTE *
  *****
5 BORDER RND*6: PAPER RND*6:
INK 9: CLS
10 PRINT "Daca nu doriti sa vedeti"
20 PRINT "raspunsurile apasati tasta "; FLASH 1;"n"; FLASH 0;
"
30 PRINT "Altfel apasati orice tasta."
50 LET r$=INKEY$: IF r$="" THE
N GO TO 50
60 CLS
65 POKE 23692,-1
70 PRINT AT 0,5;"CUVINTE INCRU
CISATE";AT 0,5; OVER 1;"-----
-----"
80 INPUT "Cite cuvinte?";a
90 INPUT "Introduceti cel mai lung cuvint ";b$
100 DIM c$(a,LEN b$)
110 LET c$(1)=b$
120 FOR c=2 TO a
130 INPUT ("Introduceti cuvint ul
numarul ";c);d$
140 IF LEN d$>LEN b$ THEN PRINT
AT 20,0; FLASH 1;"LUNGIME DE CU
VINT NEVALIDA"; GO TO 130
160 LET c$(c)=d$
190 NEXT c
199 REM Urmatoarea linie contin
e 22 de spatii
200 PRINT AT 19,0;"
"
210 LET d=LEN b$+2
220 FOR e=1 TO d: FOR f=1 TO d
230 PRINT AT e,f;"*"
240 NEXT f: NEXT e
250 DIM h$(d,d)
260 FOR q=1 TO a
270 LET j$=c$(q)
280 PRINT AT 19,0;j$
310 LET x=INT (RND*d)+1: LET y=
INT (RND*d)+1
320 LET z=INT (RND*3): LET w=IN
T (RND*3)
330 IF z=0 AND w=0 THEN GO TO 3
20
360 IF z=2 THEN LET z=-1
370 IF w=2 THEN LET w=-1
380 DIM k(LEN j$,2)
390 FOR l=1 TO LEN j$
400 IF j$(l)="" THEN GO TO 460
410 LET x=x+z: LET y=y+w
430 IF x<1 OR x>d OR y<1 OR y>d
THEN GO TO 270
440 IF (NOT h$(x,y)="" ) AND (N
OT (h$(x,y)=j$(l))) THEN GO TO 2
70
445 LET k(1,1)=x: LET k(1,2)=y
450 PRINT AT 19,1-1; INVERSE 1;
INK 9;j$(l)
460 NEXT l
470 FOR m=1 TO LEN j$
480 IF j$(m)="" THEN GO TO 500
485 LET h$(k(m,1),k(m,2))=j$(m)
490 IF r$="n" THEN GO TO 500
495 BEEP .01,RND*10-RND*20: PRI
NT AT k(m,1),k(m,2);j$(m)
500 NEXT m: NEXT q
510 REM Urmatoarea linie contin
e 15 spatii
520 PRINT AT 19,0;"
"
530 FOR n=1 TO d: FOR p=1 TO d
535 BEEP .01,RND*10-RND*20
540 IF NOT h$(n,p)="" THEN GO
TO 565
550 LET p$=CHR$ (INT (RND*26)+9
7)
560 PRINT AT n,p;p$: GO TO 570
565 PRINT AT n,p;h$(n,p)
570 NEXT p: NEXT n
580 PRINT AT 19,10; FLASH 1;"Te
rminat"
590 PRINT AT 20,0; FLASH 1;"Oric
e tasta pentru raspunsuri"
600 LET q$=INKEY$: IF q$="" THE
N GO TO 600
610 FOR n=1 TO d: FOR p=1 TO d
620 IF h$(n,p)="" THEN GO TO 6
40
630 PRINT AT n,p; INVERSE 1; IN
K 9;h$(n,p)
640 NEXT p: NEXT n
650 PRINT AT 19,10;"

```

Dame (variantă)

110 - primul cuvînt (c\$(1)) va fi cel mai lung cuvînt;

120-190 - introducerea celorlalte cuvinte (în afara celui mai lung);

140 - dacă se introduce un cuvînt mai lung decît cel maxim, introducerea nu se validează.

210 - dimensiunea grilei de joc (pătrat) va fi cea corespunzătoare dimensiunii celui mai lung cuvînt + 2;

220-240 - pe grila de joc se pun semne "*";

250 - rezervarea unui spațiu de memorie pentru toate căsuțele din matricea grilei de joc (h\$ de dimensiune dxd);

260-500 - încercarea de a pune în mod convenabil cuvintele introduse pe grila de joc;

310 - identificarea coordonatelor literelor cuvintelor pe grila de joc;

530 - 570 - umplerea grilei cu litere. Dacă o căsuță a grilei este rezervată pentru un anumit cuvînt atunci în ea se va pune litera corespunzătoare din cuvînt (h\$), iar dacă nu, adică dacă este goală, atunci în ea se va pune o literă aleasă la întîmplare din alfabet (p\$);

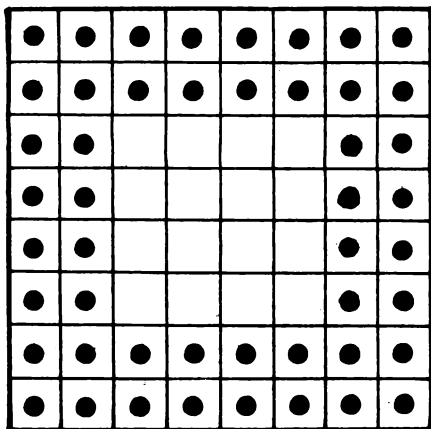
550 - stabilirea literei întîmplătoare din alfabet care se va pune într-o căsuță goală a grilei;

580 - terminarea completării grilei cu litere;

610-640 - dacă jucătorul dorește, calculatorul identifică pe ecran cuvintele introduse afișîndu-le în video invers (linia 630).

Dame (variantă)

Jocul se desfășoară pe o tablă caroiată de 8x8 poziții din care unele sînt ocupate cu piese.



Iată dispunerea pieselor în poziția inițială.

Scopul jocului este de a rămîne cu o singură piesă pe tabla de joc prin eliminarea celorlalte.

Eliminarea unei piese se efectuează ca în jocul de dame, adică prin săritura peste o piesă în diagonală.

Locația de pe tablă în care se ajunge trebuie să fie goală.

Calculatorul va cere mutarea care urmează să se efectueze.

Se introduce un număr de 4 cifre, primele două reprezentînd linia și coloana locației de plecare iar următoarele linia și coloana locației în care se dorește să se ajungă. Se acționează CR (ENTER) pentru validarea răspunsului.

Exemplu de mutare care se poate efectua chiar la începutul jocului: 1133 va elimina piesa situată în locația 22 și va plasa piesa care era inițial în locația 11 în locația 33.

Calculatorul depistează orice eroare de mutare cerînd o altă mutare.

Aveți posibilitatea să întrerupeți partida în orice moment acționînd tasta CR (ENTER) în locul unui răspuns atunci cînd calculatorul solicită o mutare. Acest lucru este deosebit de util în finaluri de partide, atunci cînd nu mai

este posibilă nici o eliminare de piese. Calculatorul va propune în acest caz o altă partidă. Pentru aceasta veți acționa tasta D și apoi CR (ENTER). Această variantă de dame nu este un joc ușor. Eliminarea a 42-45 de piese este o realizare satisfăcătoare (vezi figura), iar eliminarea a 46 sau 47 de piese este o adevărată performanță.

	1	2	3	4	5	6	7	8
1		●						●
2								
3					●			
4								
5								
6								●
7								
8								

Dame (variantă)

```

5 POKE 23609,100
7 BORDER 0: PAPER 0: INK 4: C
LS
8 GO SUB 8000
10 DIM a(4): DIM t(8,8): LET h
=48
20 FOR i=3 TO 6
30 FOR j=3 TO 6
40 LET t(i,j)=1
50 NEXT j
55 NEXT i
57 GO SUB 9500
60 GO SUB 9000
65 PRINT AT 0,28;"48";AT 2,27;
"pioni"
70 PRINT AT 0,0;"
"
75 INPUT "mutarea ?";m$
80 IF m$="" THEN GO TO 2000
85 IF LEN m$(<>)4 THEN GO TO 75
90 FOR i=1 TO 4
95 LET a(i)=VAL m$<i)
100 IF a(i)=0 THEN GO TO 75
110 NEXT i
140 FOR l=1 TO 4
150 IF (a(i)-1)*(a(i)-8)>0 THEN
GO TO 500
160 NEXT j
170 IF t(a(1),a(2))<>0 OR t(a(3)
,a(4))=0 THEN GO TO 500
180 IF ABS (a(1)-a(3))<>2 OR ABS
(a(2)-a(4))<>2 THEN GO TO 500
190 IF t(0.5*(a(1)+a(3)),0.5*(a
(2)+a(4)))<>0 THEN GO TO 500
200 LET t(0.5*(a(1)+a(3)),0.5*(
a(2)+a(4)))=1: LET t(a(1),a(2))=
1: LET t(a(3),a(4))=0
210 LET h=h-1: GO SUB 1000
220 IF h=1 THEN GO TO 2000
230 GO TO 70
500 PRINT AT 0,0;"mutare incoro
cta"
510 FOR k=1 TO 10
520 BEEP 0.01,10
530 BEEP 0.01,30
540 BEEP 0.01,20
550 NEXT k
560 GO TO 70
1000 PRINT AT a(1)*2+3,a(2)*2+7;
" ";AT a(3)*2+3,a(4)*2+7; INK b;
"●";AT a(1)+a(3)+3,a(2)+a(4)+7;"
"
1010 PRINT AT 0,28;" ";AT 0,28;
h
1020 RETURN
2000 INPUT "doriti rejucare ?";r
$: IF r$="" THEN GO TO 2000
2010 IF r$<1><"d" THEN STOP
2020 RUN
8000 FOR l=0 TO 7
8010 READ a
8020 POKE USR "<"<i,a
8030 NEXT l
8040 DATA 60,126,255,255,255,255
,126,60
8050 RETURN
9000 FOR l=0 TO 8
9010 PLOT 67+16*i,140: DRAW INK
a,0,-128
9040 PLOT 67,140-16*i: DRAW INK
a,128,0
9050 NEXT i
9055 LET q=0
9060 FOR i=1 TO 8
9070 PRINT AT 3,9+q;i;AT 21,9+q;
i
9080 PRINT AT 5+q,7;i;AT 5+q,25;
i
9090 LET q=q+2
9100 NEXT i
9130 FOR l=1 TO 8
9140 FOR j=1 TO 8
9150 IF t(i,j)=0 THEN PRINT AT 2
*i+3,2*j+7; INK b;"●": BEEP 0.01
,10
9160 NEXT j
9170 NEXT i
9180 RETURN
9500 LET a=2: LET b=5
9510 CLS : GO SUB 9000
9520 PRINT AT 0,9; FLASH 1; BRIG
HT 1;"DAME"
9530 FOR i=1 TO 8
9531 FOR j=1 TO 8
9532 PRINT AT 2*i+3,2*j+7;" ": B
EEP 0.01,30
9533 NEXT j
9534 NEXT i
9535 LET i=INT (RND*8+1)
9536 LET j=INT (RND*8+1)
9537 PRINT AT 2*i+3,2*j+7; INK 5
; FLASH 1; BRIGHT 1;"●"

```



```

9540 FOR i=1 TO 20
9542 BEEP 0.01,20
9544 BEEP 0.01,10
9545 NEXT i
9550 CLS
9560 LET a=2: LET b=5
9570 RETURN
    
```

Descrierea programului

- programul principal se află între liniile 5 și 230;
- în linia 5 se modifică durata sunetului emis la acționarea unei

taste;

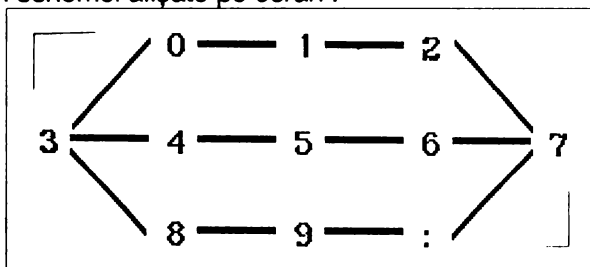
- linia 8: apelarea subrutinei de definire a caracterului care va reprezenta o piesă (tasta A în modul grafic);
- linia 10: declararea tablourilor de lucru : variabila a va reprezenta mutarea în curs iar variabila t va reprezenta locațiile tablei de joc;
- liniile 20-50: locațiile care nu conțin piese sînt identificate în tabloul t prin cifre 1;
- linia 57: apelarea subrutinei de prezentare;
- linia 60: apelarea subrutinei de afișare a tablei de joc și a pieselor;
- linia 80: dacă la cererea calculatorului pentru mutare jucătorul acționează CR (ENTER) partida se încheie;
- linia 85: dacă jucătorul introduce o mutare care conține mai puțin de 4 cifre, propunerea sa este refuzată;
- liniile 90-110: analiza mutării; a(1) și a(2) conțin coordonatele piesei care se va muta; a(3) și a(4) conțin coordonatele locației goale în care se va muta piesa;
- liniile 140-160: verificarea faptului că locația de plecare și cea de sosire aparțin tablei de joc;
- linia 170: testarea faptului că locația de plecare conține o piesă, iar cea de sosire este goală;
- linia 180: testarea faptului că mutarea se face în diagonală;
- linia 190: testarea faptului că locația peste care se sare conține o piesă;
- linia 200: eliminarea piesei peste care s-a sărit, eliberarea locației de plecare și afișarea unei piese în locația de sosire;

Vînătoare engleză

- linia 210: numărul de piese care rămîn pe tabla de joc este diminuat cu 1; apelarea subrutinei de afișare a pieselor;
- linia 220: testul unei eventuale victorii;
- linia 230: se reia ciclul;
- liniile 500-560: cazul unei mutări interzise;
- liniile 1000-1020: subrutina de afișare a pieselor; pentru obținerea caracterului pentru pisă se va acționa tasta A în modul grafic;
- liniile 8000-8050: subrutina de definire a caracterului pentru piesă (A în modul grafic);
- liniile 1000-9180: subrutina de afișare a tablei de joc și a pieselor;
- liniile 9500-9570: subrutina de prezentare a jocului.

Vînătoare engleză

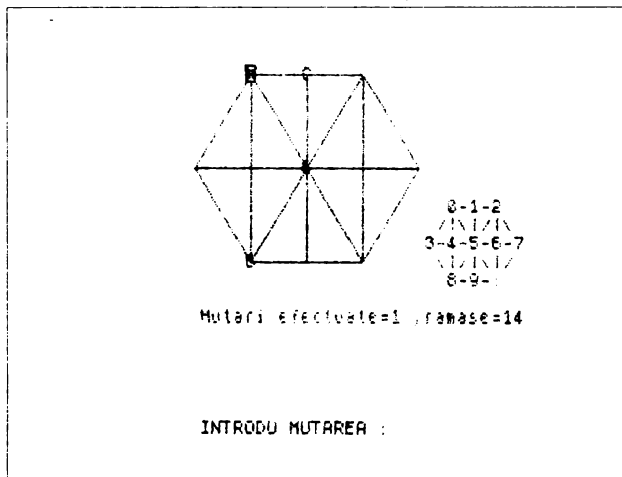
Pe ecran apare o rețea tablă de joc cu 11 noduri. Fiecare nod este simbolizat prin cifre de la 0 la 9, iar al 11-lea nod prin simbolul ":" conform schemei afișate pe ecran .



Pe nodurile rețelei sînt poziționați 3 copoi (reprezentați grafic prin litera C) și un iepure (reprezentat grafic prin litera I). Un copoi se poate deplasa în alt nod, dacă există traseu între nodurile respective, scopul jocului fiind de a prinde iepurele într-un nod de unde nu se mai poate deplasa.

Jocul este asemănător cu cel numit "moara", joc ce se juca la moară cu un bob de grâu și trei de porumb, un iepure și trei câini.

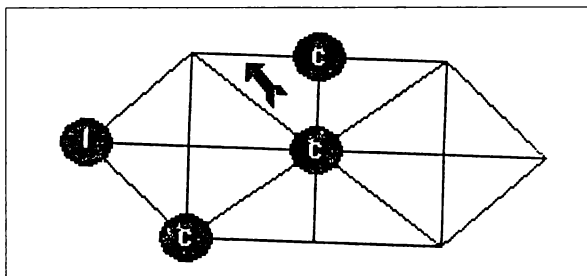
Jocul de desfășoară astfel: jucătorul va indica o mutare pentru deplasarea unui copoi, după care calculatorul va efectua o mutare a iepurelui. Indicarea unei mutări de către jucător se va face astfel: se



tastează simbolul nodului unde se află copoiul care se dorește a se deplasa și simbolul nodului unde se dorește să ajungă. Apoi se va acționa tasta CR (ENTER). Pentru indicarea nodului al 11-lea se va acționa tasta Z împreună cu SYMBOL SHIFT. De exemplu, pentru deplasarea unui copoi care se găsește în nodul 5, în nodul :, se va tasta 5 și apoi simbolul SYMBOL SHIFT + Z.

Există 3 grade de dificultate diferențiate prin poziția inițială. Stabilirea nivelului se face de jucător la începutul jocului (în urma întrebării adresată de calculator). Pentru a prinde iepurele, jucătorul are la dispoziție 15 mutări, după care se poate începe, la dorința jucătorului, un joc nou. Pe ecran apar afișate permanent numărul de mutări efectuate precum și numărul de mutări rămase.

Poziția importantă a jocului se poate observa în figură, bineînțeles existînd și poziția simetrică (cu iepurele în nodul 7). În ambele poziții trebuie găsită o mutare, astfel încît iepurele să nu aibă altă alternativă decît nodul 4 (de unde va avea doar posibilitatea reîntoarcerii în nodul 3) și, în același timp, să existe și pentru copoi posibilitatea de mutare. Astfel mutarea copoiului din nodul 1 în nodul 0 nu este bună deoarece după forțarea iepurelui la nodul 4, copoiul nu va mai avea o replică



fără a lăsa alternativă de scăpare a iepurelui. Mutarea corectă este de la nodul 5 la nodul 0 după care urmarea este foarte simplă. Din orice poziție de joc (inclusiv, cele cu grade de dificultate sporită) se poate ajunge în poziția descrisă.

Descriere joc

20 - date pentru citirea variabilelor tip șir de caractere folosite pentru strategia de joc a iepurelui.

30 - 60 - citirea variabilelor tip șir de caractere (A\$ -K\$) folosite pentru strategia de joc a iepurelui. A\$ reprezintă strategia iepurelui atunci cînd acesta se află în nodul 0 (din nodul 0 se poate deplasa într-unul din nodurile 5, 1, 4 sau 3). Similar se vor forma perechi B\$ - iepurele în nod, 1, C\$ - iepurele în nod 2, etc. Cînd iepurele se află într-un nod și trebuie să efectueze o mutare se vor inspecta pe rînd nodurile conform informațiilor conținute în variabila tip șir de caractere asociată, iepurele mutîndu-se în primul nod liber găsit astfel.

```

2 BORDER 7: PAPER 7: INK 0: F
LASH 0: BRIGHT 0: OVER 0: CLS
18 CLS
20 DATA "5134", "502", "5176", "0
84", "5083", "08:21946", "52:7", "2:
6", "5934", "58:", "5976"
30 RESTORE 20: READ A$, B$, C$
40 READ D$, E$, F$
50 READ G$, H$, I$
60 READ J$, K$
70 DATA 56, 159, 96, 159, 136, 159,
16, 119, 56, 119, 96, 119, 136, 119, 176
, 119, 56, 79, 96, 79, 136, 79
80 DIM K(11,2)
90 FOR I=1 TO 11
100 FOR J=1 TO 2
110 READ K(I,J)
120 NEXT J
130 NEXT I
140 CLS
145 LET MU=15
150 PRINT AT 9,25;"0-1-2";AT 10
,24;"\|:\|:\|";AT 11,23;"3-4-5-6-
7";AT 12,24;"\|:\|:\|/";AT 13,25;"
8-9-:"
165 PRINT AT 15,3;"Mutari efect
uate= ;ramase=";
177 GO SUB 180: GO TO 330
180 PLOT K(4,1)+4,K(4,2)-4
190 DRAW K(8,1)-K(4,1),K(8,2)-K
(4,2)
200 DRAW K(3,1)-K(8,1),K(3,2)-K
(8,2)
210 DRAW K(9,1)-K(3,1),K(9,2)-K
(3,2)
220 DRAW K(11,1)-K(9,1),K(11,2)
-K(9,2)
230 DRAW K(3,1)-K(11,1),K(3,2)-
K(11,2)
240 DRAW K(1,1)-K(3,1),K(1,2)-K
(3,2)
250 DRAW K(9,1)-K(1,1),K(9,2)-K
(1,2)
260 DRAW K(4,1)-K(9,1),K(4,2)-K
(9,2)
270 DRAW K(1,1)-K(4,1),K(1,2)-K
(4,2)
280 DRAW K(11,1)-K(1,1),K(11,2)
-K(1,2)
290 DRAW K(8,1)-K(11,1),K(8,2)-
K(11,2)
300 PLOT K(2,1)+4,K(2,2)-4
310 DRAW K(10,1)-K(2,1),K(10,2)
-K(2,2)
320 RETURN
330 DIM P(11)
350 PRINT #1;"GRADUL DE DIFICUL
TATE (1,2,3) ?";
360 PAUSE 0: LET GD=CODE INKEY$
-48: IF GD<1 OR GD>3 THEN GO TO
350
365 INPUT 1
366 IF GD<>2 THEN GO TO 425
370 FOR Q=1 TO 3
380 LET X=INT (RND*11)
390 IF P(X+1)=1 THEN GO TO 380
400 GO SUB 1000
410 NEXT Q
413 LET X=INT (RND*11)
416 IF P(X+1)=1 THEN GO TO 413
419 GO SUB 3000
422 GO TO 430
425 LET X=1: GO SUB 1000
426 LET X=5: GO SUB 1000
427 LET X=9: GO SUB 1000
428 IF GD=1 THEN LET X=0: GO SU
B 3000
429 IF GD=3 THEN LET X=3+4*(INT
(RND*2)): GO SUB 3000
430 PRINT AT 21,3;"INTRODU MUTA
REA : "; INPUT M$: GO SUB 250
0
440 LET X=CODE (M$(1))-48
450 LET Y=CODE (M$(2))-48
460 LET MU=MU-1
461 IF MU=-1 THEN GO TO 465
462 PRINT AT 18,0;" Incearca
cu un lepure ma
i obositi!"
463 GO TO 6080
465 PRINT AT 15,20;" ";AT 15,2
0;15-MU
466 PRINT AT 15,30;" ";AT 15,3
0;MU
470 IF X<0 OR X>10 OR Y<0 OR Y>
10 THEN GO TO 430
480 IF P(X+1)=0 OR P(X+1)=2 THE
N GO TO 1500
490 IF P(Y+1)=1 OR P(Y+1)=2 THE
N GO TO 1500
500 GO SUB 5000

```

Vînătoare engleză

```

510 IF X=Y THEN GO TO 430
520 GO SUB 2000
530 LET X=Y
540 GO SUB 1000
550 LET X=-1
560 LET X=X+1
570 IF NOT P(X+1)=2 THEN GO TO
560
580 GO SUB 6000
590 GO SUB 4000
600 LET X=Y
610 GO SUB 3000
620 GO TO 430
630 STOP
1010 LET A=K(X+1,1): LET AA=A/B
1020 LET B=K(X+1,2): LET BB=175-
B: LET BB=BB/B
1030 BEEP .1,10: PRINT AT BB,AA;
FLASH 1;"C"
1050 LET P(X+1)=1
1060 RETURN
1500 PRINT AT 20,0;"MUTARE GRESI
TA I": PAUSE 0: PRINT AT 20,0;TA
B 20: GO TO 430
2010 LET A=K(X+1,1): LET AA=A/B
2020 LET B=K(X+1,2): LET BB=175-
B: LET BB=BB/B
2030 PRINT AT BB,AA;" "
2040 GO SUB 180
2050 LET P(X+1)=0
2060 RETURN
2500 LET M$=M$+" ": PRINT AT 21
,21: INK 2; PAPER 6;M$(1 TO 2):
PAPER 7: INK 0: RETURN
3010 LET A=K(X+1,1): LET AA=A/B
3020 LET B=K(X+1,2): LET BB=175-
B: LET BB=BB/B
3030 BEEP .1,10: PRINT AT BB,AA;
INVERSE 1;"I"
3050 LET P(X+1)=2
3060 RETURN
4010 LET A=K(X+1,1): LET AA=A/B
4020 LET B=K(X+1,2): LET BB=175-
B: LET BB=BB/B
4030 PRINT AT BB,AA;" "
4040 GO SUB 180
4050 LET P(X+1)=0
4060 RETURN
5000 GO SUB 7000: LET Z$=V$
5010 LET L=LEN (Z$)
5020 LET I=0
5030 LET I=I+1
5040 LET Z=CODE (Z$(I))-48
5050 IF Y=Z THEN RETURN
5060 IF I=L THEN LET Y=X: RETURN

5061 GO TO 5030
6000 GO SUB 7000: LET Z$=V$
6010 LET L=LEN (Z$)
6020 LET I=0
6030 LET I=I+1
6040 LET Z=CODE (Z$(I))-48
6050 IF P(Z+1)=0 THEN GO TO 6140
6060 IF I<L THEN GO TO 6030
6070 PRINT AT 18,5;"Iepurele est
e prins.": FOR G=1 TO 20: BEEP .
05,G: BEEP .05,G-3: NEXT G
6080 PRINT AT 20,6; FLASH 1;"JOC
UL S-A TERMINAT"
6090 INPUT 2: PRINT #2;" MAI
DORESTI ?(DA/NU)"
6100 PAUSE 0
6105 LET R$=INKEY$
6110 IF R$="D" OR INKEY$="d" THE
N GO TO 140
6130 CLS : PRINT AT 10,5;"CRED C
A TI-A PLACUT.": STOP
6140 IF NOT (Z=1 OR Z=9) THEN LE
T Y=Z: RETURN
6150 IF NOT (P(1)+P(6)+P(3)=4 OR
P(9)+P(6)+P(11)=4) THEN LET Y=Z
: RETURN
6160 IF P(1)=2 OR P(9)=2 THEN GO
TO 6190
6170 IF P(8)=0 THEN LET Y=7
6175 LET Y=6
6180 RETURN
6190 IF P(4)=0 THEN LET Y=3
6195 LET Y=4
6200 RETURN
6210 IF NOT (P(9) AND P(6) AND P
(3)=1 OR P(1) AND P(6) AND P(11)
=1) THEN LET Y=Z: RETURN
6220 IF X=4 OR X=6 THEN LET Y=Z:
RETURN
6230 IF X=0 OR X=8 THEN LET Y=4:
RETURN
6240 IF X=2 OR X=10 THEN LET Y=6
: RETURN
6250 LET Y=Z: RETURN
7000 LET V=X+1
7010 IF V=1 THEN LET V$=A$
7020 IF V=2 THEN LET V$=B$
7030 IF V=3 THEN LET V$=C$

```

```

7040 IF V=4 THEN LET V$=D$
7050 IF V=5 THEN LET V$=E$
7060 IF V=6 THEN LET V$=F$
7070 IF V=7 THEN LET V$=G$
7080 IF V=8 THEN LET V$=H$.
7090 IF V=9 THEN LET V$=I$
7100 IF V=10 THEN LET V$=J$
7110 IF V=11 THEN LET V$=K$
7120 RETURN
9999 SAVE "vinatoare": VERIFY ""
    
```

70 - date reprezentînd coordonatele punctelor - noduri ale tablei de joc de pe ecranul grafic.

80 - rezervarea de spații de memorie pentru matricea K în elementele căreia sînt memorate coordonatele de puncte ale tablei de joc.

90 - 130 - citirea coordonatelor punctelor pentru tabla de joc. K(I,1) reprezintă coordonatele sau modificările coordonatelor de pe orizontală; K(I,2) reprezintă coordonatele sau modificările coordonatelor de pe verticală.

145 - inițializarea variabilei MU care reprezintă numărul de mutări rămase. La începutul jocului numărul de mutări (rămase) la dispoziția jucătorului este 15.

177 GO SUB 180 - apelarea subrutinei de desenare a tablei de joc.

180 - 310 - desenarea rețelei prin intermediul coordonatelor și modificărilor de coordonate K(I,3).

330 - rezervarea unui spațiu de memorie variabilei - indicator P. Această variabilă indică starea fiecărui din cele 11 noduri ale rețelei la un moment dat, astfel:

$P(X) = 0$ nodul X este liber;

$P(X) = 1$ în nodul X se află un copoi;

$P(X) = 2$ în nodul X se află iepurele.

350 -360 - introducerea gradului de dificultate (variabila GD) și respingerea valorilor neinteresante pentru gradul de dificultate.

365 - ștergerea mesajului apărut pe linia de jos a ecranului.

366 - stabilirea locului de pornire a jocului (numărul liniei de program) pentru diferite grade de dificultate introduse.

370 - începerea jocului pentru gradul de dificultate 2.

370 - 410 - desenarea copoilor pe tabla de joc.

380 - alegerea întâmplătoare a unui număr care va reprezenta un nod de rețea în care se va pune un copoi.

390 - dacă în nodul ales se află deja un copoi, atunci se va alege altul.

400 GO SUB 1000 - apelarea subrutinei pentru stabilirea coordonatelor pentru desenarea copoilor.

A - variabila pentru stabilirea coordonatelor pe orizontală.

B - variabila pentru stabilirea coordonatelor pe verticală.

AA - variabila pentru stabilirea coloanei pe care va fi desenat caracterul corespunzător pentru copoi (C).

BB - variabila pentru stabilirea liniei pe care va fi desenat caracterul corespunzător pentru un copoi.

Calculul se va face conform următorului algoritm: dacă nodul rețelei (punctul) în jurul căruia se va înscrie caracterul corespunzător pentru copoi este de coordonate (A,B) atunci acel caracter se va înscrie în celula-caracter a cărei linie va fi $(175-B)/8$ și a cărei coloană va fi A/8.

1030 - desenare copoi.

1050 - marcarea nodului de rețea cu indicatorul de copoi.

413 - alegere nod pentru iepure.

416 - dacă în nodul ales se găsește un copoi atunci se alege alt nod.

419 GO SUB 3000 - apelarea subrutinei pentru stabilirea coordonatelor pentru desenarea iepurelui.

Calculul se va face conform aceluiași algoritm ca și cel descris pentru înscrierea caracterului copoi.

3030 - desenarea iepure.

3050 - marcarea nodului de rețea cu indicatorul de iepure.

425 - început joc pentru grad de dificultate 1 sau 3.

425-427 - stabilire noduri pentru copoi (pentru gradul de dificultate 1 sau 3, nodurile pentru copoi vor fi 1, 5 și 9).

428 - dacă gradul de dificultate este 1 atunci iepurele se va desena în nodul 0. Apoi urmează același algoritm pentru înscrierea iepurelui în rețea și marcarea nodului cu indicatorul iepure.

429 - dacă gradul de dificultate este 3 atunci iepurele se va desena într-unul din nodurile 3 sau 7.

430 - introducere mutare (subrutina 2500). Sirul de două caractere introduse se va păstra în variabila M\$ care va reprezenta deci mutarea efectuată de jucător.

440 - X reprezintă primul caracter introdus pentru mutare.

450 - Y reprezintă al 2-lea caracter introdus pentru mutare.

462 - dacă numărul de mutări rămase este mai mic ca 0 jocul se termină.

465 - 466 - afișarea numărului de mutări efectuate (15-MU) și de mutări rămase (MU).

470 - dacă codul oricărui caracter (dintre cele două introduse la mutare) nu reprezintă un nod, atunci introducerea mutării se va repeta (numărul de mutări rămase scăzând însă cu o unitate).

480 - dacă la efectuarea mutării, primul caracter introdus reprezintă un nod liber sau un nod ocupat de un iepure, înseamnă că mutarea este greșită (numai copoii se pot muta de către jucător, deci primul caracter introdus trebuie să reprezinte un nod în care se află un copoi). Mutarea se va repeta (numărul de mutări rămase scăzând însă cu o unitate).

490 - dacă al doilea caracter introdus reprezintă un nod ocupat.

500 GO SUB 5000 - apelarea subrutinei pentru mutarea iepurelui.

GO SUB 7000 - apelarea subrutinei pentru salvarea (memorarea) nodului în care se află iepurele (în variabila V) și a posibilităților de mutare a iepurelui (în variabila V\$) conform nodului în care se află.

510 - dacă al doilea caracter introdus pentru mutare este identic cu primul (X-Y) atunci introducerea mutării se va repeta (numărul de mutări scăzînd cu o unitate).

520 GO SUB 2000 - se începe mutarea copoiului conform mutării (validate) introduse de jucător. Mai întîi se șterge copoiul (2030), se redesenează tabla de joc (GO SUB 180) și se descarcă nodul de indicatorul copoi devenind un nod liber (2050).

540 GOSUB 1000 - mutarea copoiului se termină prin desenarea copoiului pe nodul indicat de mutarea jucătorului.

580 GO SUB 6000 - apelarea subrutinei pentru găsirea unei mutări pentru iepure.

6010 - 6060 - se inspectează fiecare posibilitate de mutare a iepurelui indicată de variabila tip șir de caractere Z\$.

6050 - dacă s-a găsit un nod liber, se începe efectuarea mutării iepurelui în acel nod.

6060 - dacă s-a inspectat toată lista de posibilități de mutare a iepurelui și nu s-a găsit nici un nod liber înseamnă că iepurele este prins.

60-70 - mesaj și efecte sonore care indică prinderea iepurelui.

6090-6131 - selectare opțiune joc nou.

590 GO SUB 4000 - apelarea subrutinei pentru iepure. Mai întîi iepurele este șters din vechea poziție (4030) se redesenează tabla de joc (GO SUB 180) și se descarcă nodul de indicatorul iepure, devenind nod liber (4050).

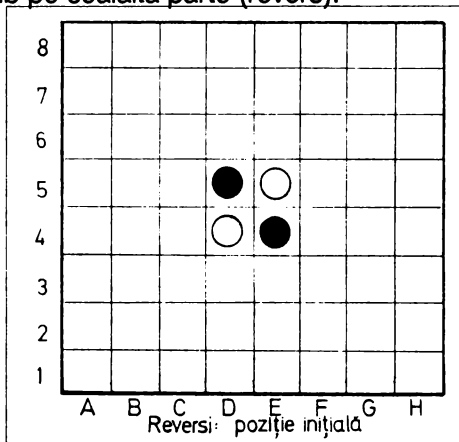
610 GO SUB 3000 - mutarea iepurelui se termină prin desenarea iepurelui în nodul liber găsit.

620 - se reia algoritmul prin cererea de introducere a unei noi mutări.

Reversi

Se joacă pe o tablă de 8x8 poziții (căsuțe), fiecare aflându-se la intersecția unei coloane cu o linie. În versiunea prezentată, jocul se poate desfășura între calculator și jucător sau între doi jucători, existând și posibilitatea unei partide demonstrative simulată de calculator (0 jucători).

În joc există 64 piese, fiecare putând fi albă sau neagră, de unde și denumirea jocului, care inițial se juca cu piese colorate în negru pe o parte și în alb pe cealaltă parte (revers).

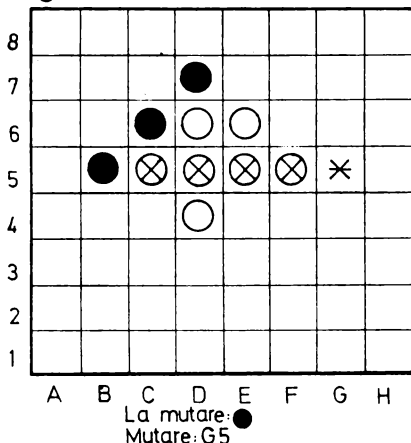


La începutul jocului, din poziția inițială jucătorii își aleg culorile, piesele negre considerându-se a fi ale unuia, iar cele albe ale celuilalt. Jucătorul la mutare va pune o piesă de culoare aleasă inițial pe o poziție care formează o linie orizontală, verticală sau diagonală cu o poziție pe care este situată o piesă de aceeași culoare, astfel încât toate piesele dintre acestea vor căpăta și ele culoarea celor din

Reversi

extremități. De exemplu: în situația din figură , dacă jucătorul care are piesele negre va pune o piesă pe poziția marcată cu *, atunci toate piesele albe marcate cu X vor deveni negre.

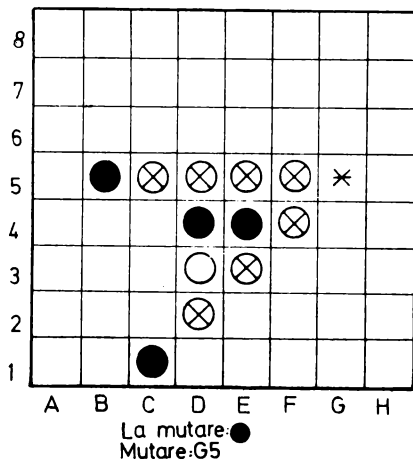
Dacă la punerea piesei se formează mai multe linii cu poziții pe care sînt situate piese de aceeași culoare, astfel încît, între acestea să existe numai poziții ocupate cu piese de culoare inversă, atunci toate piesele de pe aceste linii vor căpăta culoarea celor din extremități. Este cazul descris în situația din figura de mai jos , în care dacă se pune o piesă neagră pe poziția marcată cu * atunci toate piesele albe marcate cu X vor deveni negre, deoarece se formează două linii (una diagonală și alta orizontală) care au la



extremități piese negre.

Cîștigă jucătorul care are mai multe piese pe tablă atunci cînd nu mai este posibilă punerea vreunei piese de nici unul dintre parteneri.

Jocul prezentat începe cu întrebarea dacă se dorește sunet sau nu în timpul desfășurării jocului. Se va răspunde acționîndu-se una din tastele D sau N, după cum se dorește prezența sunetelor. Apoi va apare pe ecran poziția inițială a celor 4 piese și tabla de joc. Va



urma întrebarea referitoare la numărul de jucători. Se va răspunde cu 0, 1 sau 2 după cum se dorește o demonstrație de partidă (0 jucători se va juca calculator-calculator), o partidă calculator-jucător (1 jucător) sau, respectiv, o partidă între 2 jucători.

a) Dacă se răspunde cu 0 se va afișa scorul inițial: **NEGRU 2 - ALB 2** și calculatorul va începe simularea unei partide, mutînd singur pentru ambele piese. După fiecare punere de piesă se indică scorul și jucătorul care este la mutare. La sfîrșitul demonstrației, jocul se va putea relua introducîndu-se din nou opțiunea de 0, 1 sau 2 jucători.

b) Dacă se răspunde cu 1, jucătorul va avea posibilitatea de a-și alege culoarea pieselor cu care va juca, răspunzîndu-se la întrebarea "joci ALB sau NEGRU?", prin tastarea primei litere a culorii alese. Pentru mutarea sa calculatorul va muta singur. Jucătorul va indica poziția în care dorește să pună piesa prin deplasarea cursorului (un cerculeț clipitor) pe tabla de joc. Deplasarea cursorului se realizează cu următoarele taste: A - sus, Z - jos, X - stînga, C - dreapta. Dacă prin aceste deplasări s-a ajuns într-o poziție pe care dorește să se pună o piesă, atunci se face confirmarea mutării prin acționarea tastei V. Dacă se face o confirmare de mutare (se acționează tasta V) atunci cînd cursorul nu este pe o poziție validă (nu se formează o linie care are la extremități piese de culoarea aleasă de jucător), atunci mutarea (punerea piesei) nu va fi luată în considerație. După fiecare punere de piesă se indică scorul și jucătorul (NEGRU sau ALB) care este la mutare. La sfîrșitul partidei jocul se va putea relua introducîndu-se din nou opțiunea de 0, 1 sau 2 jucători.

c) Dacă se răspunde cu 2 partidă între cei doi jucători se va desfășura în mod similar cu b) cu deosebirea că nu va mai pune piese calculatorul ci, un alt jucător. Primul la mutare va fi jucătorul care și-a ales piesele negre.

Descrierea jocului

Pentru înțelegerea modului de realizare a programului este necesară o descriere a algoritmului (strategiei) de punere a piesei de către calculator, aceasta fiind problema cea mai interesantă în joc.

Toate piesele de pe tabla de joc formează o mulțime conexă. În vederea punerii piesei de către calculator, programul pornește dinspre partea de nord a formației de piese și inspectează toate pozițiile în care se poate pune o piesă înconjurând formația prin vest (face o tură). Pentru fiecare poziție posibilă, evaluează situația (ce s-ar întâmpla dacă ar pune acolo o piesă) obținând un anumit punctaj astfel: pentru pozițiile ocupate se adună punctele celui care efectuează mutarea și se scad cele ale adversarului. Fiecare poziție de pe tablă are asociată în acest algoritm, o anumită valoare, formându-se un câmp de valori al tuturor pozițiilor. Astfel, pozițiile din colțurile tablei au valoarea cea mai mare (100), valori mari avînd și pozițiile de pe marginea tablei. Unele poziții au asociate valori foarte mici sau chiar negative: acestea sînt pozițiile în care nu este indicată punerea piesei și sînt situate pe a doua linie și/sau a doua coloană a tablei de joc. Calculatorul va pune piesa pe poziția al cărei punctaj calculat în urma evaluării este maxim. Evaluările pe poziții se vor putea viziona pe ecran în partea stîngă a tablei de joc sub forma de mici puncte. Aceste puncte apar pe ecran de la stînga la dreapta, urmărind ordinea de parcurs a zonelor de poziții de pe lîngă formația de piese. Locul în care punctele formează o grămadă (linie) mai înaltă îi va corespunde o zonă de poziții a tablei în care se află poziția cu punctaj maxim.

10 - rezervarea de spațiu de memorie pentru variabilele b - ține minte situația de pe tabela de joc - și t - ține minte punctajul.

30 - variabila legal este un indicator al legalității mutării: dacă mutarea (punerea de piesă) se poate efectua atunci legal are valoarea 1, iar dacă nu, are valoarea 0. Variabilele șir de caractere i

```

10 DIM b(8,8): DIM t(8,8): DIM
s$(26)
20 PRINT
30 LET legal=1: LET i$="": LET
j$=""
40 PAPER 0: BORDER 0: INK 6: 0
VER 0: FLASH 0: BRIGHT 1: BEEP 0
.01,10: CLS
50 PRINT " R E V E R S I "
60 PRINT : GO SUB 600
70 PRINT "Doriti sunet ?"
80 PAUSE 0: LET a$= INKEY$
90 IF a$="d" OR a$="D" OR a$="
n" OR a$="N" THEN GO TO 110
100 GO TO 70
110 GO SUB 750
120 GO SUB 1000
130 LET turn=3: LET enemy=2
140 LET m$="Scor: NEGRU "+ STR$
bp+"- ALB "+ STR$ wp: GO SUB 18
80
150 GO SUB 770: GO SUB 1530: IF
legal=0 THEN LET m$="NEGRUL nu
are mutare": GO SUB 1880: GO TO
190
160 LET m$="mutarea NEGRULUI":
GO SUB 1880
170 IF black=1 THEN GO SUB 115
0: GO TO 130
175 IF black=2 THEN GO SUB 134
0
180 IF gamov=1 THEN GO TO 240
190 LET turn=3: LET enemy=3
200 GO SUB 770: LET m$="Scor: N
EGRU "+ STR$ bp+"- ALB "+ STR$ w
p: GO SUB 1880
210 GO SUB 1530: IF legal=0 THE
N LET m$="ALBUL nu are mutare":
GO SUB 1880: GO TO 240
220 LET m$="mutarea ALBULUI": G
O SUB 1880
230 IF white=1 THEN GO SUB 115
0: GO TO 240
235 IF white=2 THEN GO SUB 134
0
240 IF gamov=0 THEN GO TO 130
250 LET n$="Final: ALB ="+ STR$
wp+"-NEGRU="+ STR$ bp
260 LET m$="Inca un joc ?": GO
SUB 1880
270 PAUSE 0: LET a$= INKEY$
280 IF a$="D" OR a$="d" THEN G
O TO 110
290 IF a$ <> "N" AND a$ <> "n"
THEN GO TO 260
300 BORDER 7: PAPER 7: INK 0: C
LS
310 STOP
330 LET take=0
340 FOR v=-1 TO 1: LET dx=v
350 FOR w=-1 TO 1: LET dy=w
360 LET dist=0
370 LET dist=dist+1
380 LET ax=nx+dx*dist: LET ay=n
v+dy*dist
390 IF ax<1 OR ax>8 OR ay<1 OR
ay>8 THEN GO TO 560
400 IF b(ax,ay)=enemy THEN GO
TO 370
410 IF dist=1 THEN GO TO 560
420 IF b(nx+dx*dist,ny+dy*dist)
<> turn THEN GO TO 560
425 LET take=take+dist-1
430 IF moving=0 THEN GO TO 560
440 IF take <> dist-1 THEN GO
TO 520
450 LET x=nx: LET y=ny
460 IF x=1x THEN LET lx=x-1: I
F lx=0 THEN LET lx=1
470 IF x=hx THEN LET hx=x+1: I
F hx=9 THEN LET hx=8
480 IF y=ly THEN LET ly=y-1: I
F ly=0 THEN LET ly=1
490 IF y=ny THEN LET hy=y+1: I
F hy=9 THEN LET hy=8
500 IF turn=2 THEN GO SUB 1650
: GO TO 510
505 IF turn=3 THEN GO SUB 1760
510 IF turn=3 THEN LET wp=wp+1
: GO TO 520
515 LET bp=bp+1
520 FOR z=1 TO dist-1
530 LET x=nx+dx*z: LET y=ny+dy*
z
540 IF turn=2 THEN GO SUB 1650
: GO TO 550
545 IF turn=3 THEN GO SUB 1760
550 NEXT z
560 NEXT w
570 NEXT v
580 BEEP 0.01,1+take: RETURN
600 RESTORE 690: FOR x=1 TO 4
610 FOR y=1 TO 4
620 READ tactic
630 LET t(x,y)=tactic
640 LET t(x,y-v)=tactic

```

Reversi

```

650 LET t(9-x,9-y)=tactic
660 LET t(9-x,y)=tactic
670 NEXT y
680 NEXT x
690 DATA 10,1,7,5
700 DATA 1,0,2,2
710 DATA 7,2,4,3
720 DATA 5,2,3,3
730 RETURN
750 LET xp=0: LET yp=0: CLS : L
ET bp=0: LET wp=0
760 LET vib=1500: LET var=1.06:
LET n$="": GO TO 850
770 FOR x=64 TO 192 STEP 16
780 PLOT x,175
790 DRAW 0,-128
800 NEXT x
810 FOR y=175 TO 47 STEP -16
820 PLOT 64,y
830 DRAW 128,0
840 NEXT y: FOR y=0 TO 17: PRIN
T AT y,0: OVER 0:"      ": NEXT
y: RETURN
850 FOR x=1 TO 8
860 FOR y=1 TO 8
870 LET b(x,y)=1
880 NEXT y
890 NEXT x
900 FOR z=4 TO 5
910 LET x=z
920 LET y=z: GO SUB 1650
930 LET y=9-x: GO SUB 1760
940 NEXT z: GO SUB 770
950 LET bp=2: LET wp=2
960 LET lx=3: LET ly=3: LET hx=
6: LET hy=6
970 LET gamov=0
980 RETURN
1000 LET m$="Citi jucatori ?": G
O SUB 1880
1010 PAUSE 0: LET p$=INKEY$
1020 IF p$("0" OR p$)"2" THEN LE
T m$="introduceti 0,1 sau 2": GO
SUB 1880: GO TO 1000
1030 LET n$=n$+p$
1040 IF VAL p$=0 THEN GO TO 1050
1043 IF VAL p$=1 THEN GO TO 1090
1047 IF VAL p$=2 THEN GO TO 1070
1050 LET black=1: LET white=1
1060 RETURN
1070 LET black=2: LET white=2
1080 RETURN
1090 LET m$="joci ALB sau NEGRU
?": GO SUB 1880
1100 PAUSE 0: LET c$=INKEY$
1110 IF c$="a" OR c$="A" THEN LE
T black=1: LET white=2: LET n$=n
$+"ALB": RETURN
1120 IF c$="n" OR c$="N" THEN LE
T white=1: LET black=2: LET n$=n
$+"NEGRU": RETURN
1130 GO TO 1090
1150 LET moving=0
1160 LET pts=-1000
1170 LET scraux=(hx-lx)*(hy-ly)+
1: FOR c=lx TO hx: LET nx=c
1180 FOR d=ly TO hy: LET ny=d
1190 IF b(nx,ny)<>1 THEN GO TO 1
260
1200 GO SUB 330
1210 IF take=0 THEN GO TO 1260
1220 LET scrc=(nx,ny)*20+RND*10
+take*(xp+yp-32)
1230 IF scrc<pts THEN GO TO 1255
1240 LET pts=scrc
1250 LET xbest=nx: LET ybest=ny
1255 LET xsc=3+37/scraux*((nx-lx
)*(hy-ly)+ny-ly+1): LET ysc=scrc
/30: PLOT OVER 0;xsc,40: DRAW 0,
ysc
1260 NEXT d
1270 NEXT c
1280 LET nx=xbest: LET ny=ybest
1290 LET moving=1
1300 GO SUB 330
1310 IF bp+wp=64 THEN LET gamov=
1.
1320 RETURN
1340 LET ax=xc*16+56: LET ay=183
-yc*16
1350 IF b(xc,yc)=2 THEN OVER 1:
CIRCLE ax,ay,2: OVER 0
1355 IF b(xc,yc)<>2 THEN CIRCLE
ax,ay,2
1370 OVER 1: CIRCLE ax,ay,2: OVE
R 0
1380 IF b(xc,yc)=2 THEN CIRCLE a
x,ay,2
1390 IF INKEY$="x" OR INKEY$="X"
THEN LET xc=xc-1: IF xc<1 THEN
LET xc=1
1400 IF INKEY$="c" OR INKEY$="C"
THEN LET xc=xc+1: IF xc>8 THEN
LET xc=8

```


și j\$ servesc la memorarea mesajelor care apar în partea de jos a ecranului (referitor la scor și cine este la mutare).

60 GO SUB 600 - apelarea subrutinei pentru tactici.

610-670 - citirea datelor de descriere a tablei de joc (care vor fi memorate în variabila tactic. În funcție de valorile pe care le ia variabila tactic se vor calcula și valorile pentru variabilele t, care exprimă valorile asociate pozițiilor tablei de joc.

690-720 - datele asociate pozițiilor tablei de joc. Fiecare linie de program conține datele pentru o linie a tablei de joc. Se observă că este descris doar un sfert din tabla de joc (16 date), existând o simetrie a valorilor pozițiilor (tablă simetrică). De aceea se și calculează valorile variabilelor t pentru 4 zone ale tablei de joc. Se mai observă cum valoarea cea mai mare (bună) este atașată poziției din colțul tablei, iar cea mai mică (slabă), poziției

```

1410 IF INKEY$ ="z" OR INKEY$
="Z" THEN LET yc=yc+1: IF yc>8
THEN LET yc=0
1420 IF INKEY$ ="a" OR INKEY$
="A" THEN LET yc=yc-1: IF yc<1
THEN LET yc=1
1430 IF INKEY$ ="v" OR INKEY$
="V" AND b(xc,yc)=1 THEN GO TO
1450
1440 GO TO 1340
1450 LET nx=xc: LET ny=yc
1460 GO SUB 330
1470 IF take=0 THEN GO TO 1340
1480 LET moving=1
1490 GO SUB 330
1500 IF bp+wp=64 THEN LET gamov
=1
1510 RETURN
1530 LET moving=0: LET take=0
1540 FOR i=1x TO hx: LET nx=i
1550 FOR j=1y TO hy: LET ny=j
1560 IF b(nx,ny) (<) 1 THEN GO T
O 1590
1570 GO SUB 330
1580 IF take (<) 0 THEN LET xc=n
x: LET yc=ny: LET nx=8: LET ny=8
: LET i=8: LET j=8
1590 NEXT j
1600 NEXT i
1610 IF take (<) 0 THEN LET lega
l=1: RETURN
1620 IF legal=0 THEN LET gamov=
1
1625 LET legal=0
1630 RETURN
1660 LET vib= INT (vib/var)
1670 IF qs="n" OR qs="N" THEN G
O TO 1690
1680 BEEP .2,vib/100
1690 FOR n=1 TO 6
1700 CIRCLE x*16+56.183-y*16,n
1710 NEXT n
1720 LET b(x,y)=2
1730 LET wp=wp+1: LET bp=bp-1
1740 RETURN
1770 LET vib= INT (vib/var)
1780 IF qs="n" OR qs="N" THEN G
O TO 1800
1790 BEEP .2,vib/100
1800 LET rx= INT (x*16+48)/8: LE
T ry=22- INT (191-y*16)/8
1810 PRINT AT ry,rx: " "; AT ry
+1,rx: " ";

```

Reversi

```
1830 CIRCLE x*16+56,183-y*16,6
1840 LET b(x,y)=3
1850 LET bp=bp+1: LET wp=wp-1
1860 RETURN
1880 PRINT AT 18,6;:s$:AT 18,6;i$
;
1885 PRINT AT 19,6;:s$:AT 19,6;j$
;
1890 PRINT AT 20,6;:s$:AT 20,6;n$
;
1895 PRINT AT 21,5;:>":s$:AT 21,
6;m1;
1900 LET i$=j$: LET j$=n$: LET n
$=m1
1910 RETURN
1920 CLS
```

următoare pe diagonală față de aceasta (intersecția liniei și coloanei 2 a tablei de joc). Dacă se modifică aceste date se va schimba și strategia de joc a programului (calculatorului).

90 - introducerea comenzii referitoare la sunet.

100 - dacă nu a fost o comandă validă se va repeta

întrebarea solicitându-se din nou introducerea comenzii.

110 GO SUB 750 - apelarea subrutinei pentru desenarea poziției inițiale.

700 - inițializarea variabilelor pentru punerea pieselor.

760 - inițializarea valorii pentru înălțimea sunetelor.

770 - 840 - desenarea tablei de joc.

850 - 930 - desenarea pieselor (în poziția inițială).

120 GO SUB 1000 - apelarea subrutinei pentru stabilirea numărului de jucători.

1000 GO SUB 1880 - apelarea subrutinei pentru afișarea mesajelor în partea de jos a ecranului.

1880-1900 - afișarea mesajelor în partea de jos a ecranului (jucătorul la mutare și scorul).

1010-1020 - introducerea numărului de jucători; respingerea introducerii unui număr invalid de jucători.

130 - variabila turn ține minte cine este la mutare. Dacă turn are valoarea 3 atunci calculatorul va fi la mutare. În acest caz, variabila enemy (inamicul) va avea automat valoarea 2 și, invers, dacă turn are valoarea 2, atunci enemy va avea valoarea 3. Valorile turn și enemy lucrează în pereche.

140 - formarea mesajelor.

150 - apelarea subrutinei de desenare a tablei de joc și a celei de control a legalității mutării (GO SUB 1530).

180 - variabila gamov indică existența unei mutări posibile.

330 - 580 - subrutină de calcul a evaluării unei mutări.

330 - valoarea 0 a variabilei take indică inexistența unei mutări dacă legal = 0. La începutul evaluării se face take = 0 și legal = 1, iar după evaluarea lor ori amîndouă sînt 0 ori amîndouă sînt 1. (Variabilele legal și take lucrează în pereche).

Variabila dist reprezintă distanța în poziții (căsuțe) între poziția pe care se pune piesa și cea cu care se formează o linie. Valoarea variabilei dist intervine în evaluare.

430 - variabila moving semnalează cine este la mutare. Calculatorul generează o mutare la nx și riv și o evaluează în variabila scre (linia 1220) care ține minte punctajul și a cărei valoare este calculată în funcție de t. Dintre toate mutările posibile, calculatorul o alege pe aceea care are valoarea pentru scre cea mai mare. Iar acest lucru se face prin intermediul variabilelor xbest și ybest.

410 - dacă dist = 1 nu se ia în această direcție.

920 - desenează o piesă albă (GO SUB 1650).

930 - desenează o piesă neagră (GO SUB 1760).

1050 - calculatorul joacă cu amîndouă piesele.

1070 - joacă doi jucători.

1150 - începe efectuarea mutării de către calculator.

Variabila pts reprezintă numărul de puncte. Se pornește inițial de la cea mai slabă situație (1000).

Variabila nts memorează valoarea cea mai mare a punctajului pentru fiecare evaluare (care se face cu variabila scraux).

1220 - evaluarea a fost terminată.

134-1510 - subrutină de introducere a mutării jucătorului.

1390-1430 - posibilitatea de deplasare a cursorului de către jucător cu tastele A, Z, X și C. Validarea mutării (cu tasta V).

JOCURI DE AVENTURI

Pierdut în junglă

Jucătorul se află în mijlocul junglei avînd la dispoziție cîteva resurse pentru supraviețuire: o busolă (pentru a se putea orienta), un pistol cu 6 gloanțe, un cuțit și o praștie (pentru a se putea apăra). Prin deciziile care le ia va trebui să străbată jungla și să iasă cu bine din ea, ajungînd la o așezare omenească. Jucătorul își poate alege un joc ușor, potrivit sau greu. În funcție de nivelul de dificultate ales, probabilitatea de apariție a unei situații speciale (leu, păianjen uriaș, mlaștină, nisipuri mișcătoare, crocodil, etc.) crește. Jungla are o formă pătrată (100 x 100 km), punctul din care pornește inițial jucătorul fiind ales întîmplător. Jocul se desfășoară sub formă de dialog prin care calculatorul descrie situația în care se află jucătorul și opțiunile pe care acesta le are la dispoziție și cere luarea unei decizii de către jucător. Deciziile jucătorului pot fi de deplasare (taste folosite: A - pentru nord, B - pentru sud, C - pentru est și D - pentru vest), jucătorul parcurgînd la fiecare opțiune o distanță de 1 km, sau, de ieșire dintr-o situație specială descrisă în prealabil de calculator (leu, păianjen, etc.).

Schema de desfășurare a jocului este următoarea:

deciziile inițiale sînt cele de deplasare, putînd apoi apare (în mod aleator, în funcție de gradul de dificultate ales), situații speciale (pe nivelul 1) care, la rîndul lor, pot genera alte situații speciale (pe nivelul 2). După fiecare decizie a jucătorului calculatorul evaluează locul în care se găsește jucătorul, dacă a fost o decizie de deplasare, sau starea (puterea) jucătorului, dacă a fost o decizie de rezolvare a unei

situații speciale. Jocul poate lua sfârșit după evaluarea locului în care se află jucătorul (dacă acesta a ieșit din junglă) sau după evaluarea stării sale (dacă "puterea" sa a ajuns la 0). Jocul se poate relua, eventual de către alt jucător.

Descrierea programului

Linia 1 - modificări ale variabilelor de sistem pentru realizarea unui dialog fără mesajul "scroll?" și numai cu litere mari (cursor C).

30-80 - prezentare sumară a jocului.

110-120 - introducerea gradului de dificultate. Variabila J\$ reprezintă gradul de dificultate introdus de către jucător (U, M sau C). Această valoare (șir de caractere) se transformă într-o valoare numerică (prin intermediul variabilei J cu care se va lua în program și care va reprezenta gradul de dificultate.

160 - comenzile disponibile, reprezentate prin variabilele: A\$ (pentru nord), B\$ (pentru sud), C\$ (pentru est) și D\$ (pentru vest). Variabilele F\$ și G\$ reprezintă alte șiruri de caractere folosite pentru dialog (în situații speciale numărul comenzilor disponibile poate fi mai mare).

170 - K reprezintă puterea jucătorului; aceasta crește când jucătorul bea apă, mănâncă zmeură, etc. și scade când așteaptă, se rănește etc. Inițial K este 100, iar când ajunge la 0, aceasta implică sfârșitul jocului.

H, F, G, T - coordonatele jucătorului în junglă. Jucătorul începe jocul dintr-un punct al junglei ales la întâmplare. Coordonatele pe orizontală și pe verticală vor fi de minim 21 km și de maxim 70 km.

180 - M reprezintă numărul de gloarițe (inițial 6).

200 - în funcție de nivelul jocului, fiecare situație specială va apărea cu o anumită probabilitate. Variabila Y va reprezenta valoarea cu ajutorul căreia se va selecta numărul de linie la care se generează o situație.

Pierdut în junglă

```

1 POKÉ 23692,255: POKÉ 23658,
B
3 FOR G=1 TO 25: PRINT : NEXT
G
20 RANDOMIZE
30 BORDER 5: PAPER 6: INN 1: C
LS: PRINT "" Pierdut in
jungla"
40 PRINT FLASH 1;"
": GO SUB 21
10
50 PRINT "" Te-ai ratacit
in jungla"
60 PRINT "Ai un pistol cu sase
gloante,un cutit,o prastie si
o busola ."
70 PRINT "" TREBUIE SA IESI
CU VIATA DIN JUNGL
A"
80 PRINT "" Jungla are 100x
100 Km."
100 PRINT "" POKÉ 23692,255
110 PRINT "Vrei un joc:""U-uso
r;"M-mediu;"G-greu?"
120 PRINT : INPUT J$
130 LET J%=J$(1): LET J=CODE J$
: LET J=(J-50)*J/80
140 PRINT "" APASA O TASTA CA
SA PORNESTI!": PAUSE 0
160 LET A$="A": LET B$="B": LET
C$="C": LET D$="D": LET E$="E":
LET F$="F": LET G$="G"
170 LET K=100: LET H=INT (RND*5
0+21): LET F=INT (RND*50+21)
180 LET M=6: LET G=100-F: LET I
=100-H: LET Z%=C$
200 LET S=J-10: LET Y=INT (RND*
5)
210 GO SUB 1190
220 IF Y=1 THEN GO SUB 470
230 IF Y=2 OR Y=3 THEN GO SUB 7
50
240 IF Y=4 THEN GO SUB 1220
250 IF Y=5 THEN GO SUB 1430
260 IF Y=6 THEN GO SUB 380
270 IF Y=7 THEN GO TO 350
280 IF Y=8 THEN GO TO 350
290 IF Y=9 THEN GO TO 350
300 IF Y=10 THEN GO TO 350
310 IF Y=11 THEN GO TO 350
320 IF Y=12 THEN GO TO 350
330 IF Y=13 THEN GO TO 350
340 IF Y=14 THEN GO SUB 1690
350 LET K=K-1: IF K)0 THEN GO S
UB 1750
355 IF K=0 THEN GO TO 2050
360 GO TO 190
380 PRINT "Ai gasit un tufis.c
u zmeura"
390 PRINT "Le maninci?" "A-da;"
"B-nu": INPUT Q$
400 IF Q$=A$ THEN GO TO 450
410 RETURN
420 IF RND(.75) THEN GO TO 450
430 PRINT "ERAU OTRAVITOARE !!!
""Te-ai imbolnavit"
440 LET I=INT (1-RND): RETURN
450 PRINT "Ti-ai mai potolit fo
amea"
460 LET K=INT (K+J/10): RETURN
470 PRINT "Nisipuri miscatoare
!!!": GO SUB 1190
480 PRINT "Ce faci?"
490 PRINT "A-te zbati sa iesi;"
500 PRINT "B-stai nemiscat;"
510 PRINT "C-incerci sa te apuc
i de o craca"
520 PRINT "D-strigi dupa ajutor
": INPUT Q$
530 IF Q$=B$ THEN GO TO 690
540 IF Q$=C$ THEN GO TO 650
550 IF Q$=D$ THEN GO TO 610
560 PRINT "" PRINT AT 20,4;
" B ^ L D ^ B ^ C !!!"
570 GO SUB 1190
580 PRINT ""TE SCUFUNZI !!!"
: GO SUB 1190
590 PRINT ""
""TE-AI DUS LA FUND!!!": PAUSE 1
00: PRINT ""
600 LET K=0: RETURN
610 PRINT " A J U T
O R !!!": GO SUB 1190
620 PRINT " A J U T O
R !!!": GO SUB 1190
630 PRINT " A J U T O
R !": GO SUB 1190: IF Y=4 THEN R
ETURN
640 PRINT TAB 12;"AJU.....": GO
SUB 1190: PRINT TAB 14;"A...":
GO SUB 1190: PRINT TAB 14;"Ah.."
: GO SUB 1190: GO TO 590

```

```

650 IF RND*J>6 THEN GO TO 670
660 GO SUB 1190: PRINT "NU AI R
EUSIT !!!": GO SUB 1190: GO TO 5
80
670 GO SUB 1190: PRINT "AI REUS
IT !!!"
480 LET K=INT (K*(1-RND*.67)):
RETURN
690 PRINT : GO SUB 710: PRINT "
Au trecut ";Q;" ore.": IF Q(24 A
ND K)>0 THEN PRINT "Acum "; GO T
O 480
700 GO TO 1120
720 LET Q=INT (RND*40+1): LET K
=INT (K-Q)
730 PAUSE Q*10: RETURN
750 IF Y=2 THEN PRINT TAB 10;"u
n LEU !!!"
760 IF Y=3 THEN PRINT TAB 10;"u
n SARPE !!!"
770 PRINT "Ce faci?"
780 PRINT "A- o iei la sanatoas
a;"
790 PRINT "B- tragi cu pistolul
;"
800 PRINT "C- tragi cu prastia;"
810 PRINT "D- folosesti cutitul
;"
820 PRINT "E- te urci in copac?
": INPUT U$
830 IF Q$=A$ THEN GO TO 890
840 IF Q$=B$ THEN GO TO 970
850 IF Q$=C$ THEN GO TO 1020
860 IF Q$=D$ THEN GO TO 1030
870 IF Q$=E$ THEN GO TO 1060
880 LET K=INT (K-1): PRINT "Nu
s-a miscat din loc.": GO TO 770
890 GO SUB 1990: GO SUB 2040
900 IF Z$=Q$ THEN GO SUB 1690:
GO TO 770
910 IF RND*K<6 THEN GO TO 940
920 PRINT "Uff !!!: GO SUB 1190
930 PRINT "Esti in siguranta.":
RETURN
940 PRINT "TE-A AJUNS DIN URMA
!!!"
950 LET K=INT (RND*K): IF K<5 T
HEN GO TO 1060
960 LET J=J-1: PRINT "Acum ": G
O TO 770
970 IF M<1 THEN PRINT " CLIC
!...": PRINT "NU MAI AI GLOANTE
!!!": LET Q=17: GO TO 1050
980 GO SUB 1190: PRINT "
B A N G !!!": GO SUB 1190
990 LET L=J*2.5: LET M=M-1
1000 IF RND*M>17 THEN GO TO 1150
1010 PRINT "NU AI NIMERIT !!!":
GO TO 940
1020 LET L=J/1.5: GO TO 1040
1030 LET L=J/1.7: IF Y=J THEN LE
T L=L*2
1040 LET Q=INT (RND*L+1)
1050 IF Q>18 THEN GO TO 1150
1060 PRINT " Te-a muscat
!": LET K=K-(2*Q)
1070 IF K<15 THEN PRINT " T
E-A MINCAT !!!": LET K=0: RETURN
1075 GO TO 960
1080 IF Y=3 THEN GO TO 1130
1090 PRINT "Leul sta si asteapta
.": GO SUB 710: IF K<30 THEN GO
TO 1180
1100 PRINT "Au trecut ";Q;" ore.
": IF Q>24 THEN GO TO 1120
1110 PRINT "De acum poti cobori.
": RETURN
1120 GO SUB 1190: GO SUB 1190: G
O SUB 1190: PRINT "TI-E FOAME SI
TI-E SETE !!!": LET K=0: RETURN
1130 PRINT "Serpii se urca in co
paci mai usor ca oamenii
!"
1140 PRINT "S-A INCOLACIT IN JUR
UL TAU !!!": GO SUB 710: PRINT "
TE-A INGHITIT !!!": LET K=
0: RETURN
1150 PRINT "L-ai omorit !": GO S
UB 1190: GO SUB 1190
1160 PRINT "Il maninci :""A-da;
""B-nu ?": INPUT Q$: IF Q$=A$ T
HEN LET K=K+J
1170 RETURN
1180 PRINT ""AI CAZUT DIN COPAC
DE EPUIZARE !": GO TO 950
1190 LET W=INT (RND*88)+1: POKE
23692,255
1200 PAUSE W: RETURN
1220 PRINT "Un palanjen URIAS ti

```

Pierdut în junglă

```

s-a asezat pe ceafa !"
1230 PRINT "Ce faci :""A-ramii
nemiscat;"
1240 PRINT "B-te scuturi;"
1250 PRINT "C-il impusti;"
1260 PRINT "D-strigi ?": INPUT Q
$
1270 IF Q$=A$ THEN GO TO 1390
1280 IF Q$=B$ THEN GO TO 1350
1290 IF Q$=C$ THEN GO TO 1320
1300 IF Q$=D$ THEN GO TO 1340
1310 GO TO 1360
1320 IF M<1 THEN PRINT " CL
IC !...": GO TO 1350
1330 PRINT "" B A N G !!!"
: GO SUB 1190: PRINT "TE-AI RANI
T LA CEAFA !!!": GO TO 824
1340 GO SUB 610
1350 IF RND<.6 THEN GO TO 1380
1360 PRINT " Te-a muscat
!": GO SUB 1190
1370 PRINT " ERA VENINOS !!
!": GO SUB 1190: LET N=0: RETURN

1380 PRINT " Uff ! A plecat .":
RETURN
1390 GO SUB 710: IF K<5 AND K>0
THEN PRINT " Lesini de spaim
a !": GO TO 1350
1400 PRINT "Au trecut ";Q;" ore
.": IF Q>24 THEN GO TO 1120
1410 IF Q<5 THEN PRINT "Acum ":
GO TO 1230
1420 PRINT ""Ai ajuns la o mlast
ina."
1440 PRINT "Ce faci :""A-iti po
tolesti setea;"
1450 PRINT "B-treci inot mai dep
arte;"
1460 PRINT "C-te intorci ?": INP
UT Q$: GO SUB 1190
1470 IF Q$=A$ THEN GO TO 1510
1480 IF Q$=B$ THEN GO TO 1560
1490 IF Q$=C$ THEN GO TO 1670
1500 PRINT "Ai cazut in apa !":
GO TO 560
1510 IF RND<.3 THEN GO TO 1540
1520 LET K=K+RND*K/2
1530 PRINT "Te-ai mai racorit ."
: RETURN

1540 PRINT "ERA INFECTATA !!!!"
Ai febra si esti foarte slabit."
1550 LET K=K-INT (RND*K): RETURN

1560 LET Q=INT (RND*4)+1: GO SUB
1190: LET K=K-2
1570 IF Q=1 THEN GO TO 1600
1580 IF Q=2 THEN GO TO 1610
1590 GO SUB 1190: PRINT "Ai reus
it sa treci cu bine .": RETURN
1600 PRINT " TE INNECI !!!": G
O TO 610
1610 PRINT " Un crocodil !": LET
K=K-1: GO SUB 1190
1620 PRINT " INCA UNUL !!!": GO
SUB 1190: PRINT "Inoata mai repe
de !": GO SUB 1190
1630 LET Q=INT (RND*3)+1
1640 IF Q<20 THEN PRINT " Te-a
ajuns !!!": GO SUB 1190: PRINT
" TE-A MNCAT !!!": LET
K=0: RETURN
1650 PRINT " Uff ! Ai ajuns la
mal .": PRINT "Ai scapat doar c
u o sperietura !"
1660 LET K=K-INT (RND*K): RETURN

1670 GO SUB 1990: LET K=K-1: RET
URN
1690 PRINT "" BUF !!!"
1700 PRINT "Te-ai ciocnit de un
";
1710 IF Y=2 THEN PRINT "... LEU
!": RETURN
1720 IF Y=14 THEN PRINT "... COP
AC !": GO SUB 1190: LET N=INT (K
*.8): RETURN
1730 IF Y=3 THEN PRINT "... SARP
E !": RETURN
1740 RETURN
1760 PRINT TAB 10:"Ai mers 1 km.
""INCOTRO O IEI ACUM : "
1770 PRINT "A-est;""B-vest;""C
-nord;""D-sud ?": INPUT X$: LET
Z$=X$
1780 IF X$="A" THEN GO TO 1830
1790 IF X$="B" THEN GO TO 1840
1800 IF X$="C" THEN GO TO 1850
1810 IF X$="D" THEN GO TO 1860
1820 GO TO 1770

```


220-340 - stabilirea punctului de pornire al jocului (linia de program) și generarea diferitelor situații în funcție de gradul de dificultate introdus.

350 - situație obișnuită (jucătorul a parcurs 1 km în direcția indicată); puterea scade cu o unitate. Dacă K ajunge la 0 jocul ia sfârșit (linia 2050) nema mai apelându-se rutina (GO SUB 1760) prin care se generează dialogul pentru o situație obișnuită.

1760 - mesajul și cererea de indicare a comenzii.

1770 - tipărirea comenzilor disponibile.

1780-1810 - selectarea unei linii (în funcție de comanda introdusă) pentru modificarea coordonatelor locului din junglă în care se află jucătorul.

1820 - dacă se acționează o tastă care nu reprezintă o comandă validă se cere reintroducerea comenzii.

```

1830 LET F=F-1: LET G=G+1: GO TO
1870
1840 LET F=F+1: LET G=G-1: GO TO
1870
1850 LET H=H-1: LET I=I+1: GO TO
1870
1860 LET H=H+1: LET I=I-1
1870 IF F=0 OR G=0 OR H=0 OR I=0
THEN GO TO 1910
1880 IF F<5 OR G<5 OR H<5 OR I<5
THEN GO TO 1960
1890 IF F<10 OR G<10 OR H<10 OR
I<10 THEN GO TO 1950
1900 RETURN
1910 PRINT " AI REUSIT !
!!"
1920 PRINT " ESTI SALVAT
!!!"
1930 PRINT "AI AJUNS LA O ASEZAR
E OMENEASCA."
1940 GO SUB 1190: PAUSE 0: PRINT
"*****"MAI DORE
STE CINEVA SA INCERCE :""A-DA""
"B-NU": INPUT O$: IF O$(<)A$ THEN
GO TO 2060
1945 GO TO 100
1950 PRINT " SE AUD DAMENI IN DE
PARTARE !!!": RETURN
1960 PRINT " Esti aproape !"
1970 PRINT " SE ZARESC NISTE CAS
E !!!"
1980 RETURN
1990 IF Z$=A$ THEN LET X$=B$
2000 IF Z$=B$ THEN LET X$=A$
2010 IF Z$=C$ THEN LET X$=D$
2020 IF Z$=D$ THEN LET X$=C$
2040 RETURN
2040 PAUSE INT (RND*30/J): PRINT
" si fugi !": LET K=INT (K*.9):
GO SUB 1780: RETURN
2050 PRINT " MORI .""
"IMI PARE RAU,JOCUL S-A SFIRSIT.
"": GO TO 1940
2060 PRINT "APASA ORICE TASTA SA
CONTINUI .": PAUSE 0: CLS
2070 POKE 23692,255: PRINT AT 20
,10;"LA REVEDERE !""*****"
2100 BORDER 7: GO TO 9999
2110 RETURN
    
```

Pierdut în junglă

1830-1860 - modificarea coordonatelor în funcție de comanda introdusă.

1870-1890 - testarea coordonatelor locului în care se află jucătorul.

380 - încep situațiile speciale pe nivelul 1: "un tufiș cu zmeură"

390 - opțiuni pentru situația "tufiș cu zmeură"; introducere decizie.

420-450 - consecințe decizie: în 3/4 din cazuri, consecințe pozitive (crește puterea jucătorului), în 1/4 din cazuri, consecințe negative (zmeură otrăvitoare).

460 - calculul situației jucătorului (K).

470 - situație specială: "nisipuri mișcătoare".

480-520 - descriere opțiuni pentru situația specială "nisipuri mișcătoare"; introducere decizie (520).

530-550 - selectare consecință în funcție de opțiune.

560-730 - consecințe.

750 - situație specială: "leu" (sau "șarpe").

760-820 - descriere opțiuni pentru situația specială; introducere decizie.

830-870 - selectare consecințe în funcție de opțiune.

880-1200 - consecințe.

1220 - situație specială: "păianjen uriaș pe ceafă".

1230-1260 - descriere opțiuni pentru situație specială; introducere decizie (Q\$).

1270-1300 - selectarea unei ramuri de program în funcție de decizia luată.

1310-1380 - consecințe posibile ale deciziei luate.

1350 - dacă s-a luat decizia "strigi", în 1/3 din cazuri păianjenul va mușca, iar în 2/3 din cazuri va pleca.

1430 - situație specială: "mlaștină".

1440-1460 - descriere opțiuni pentru situația specială; introducere decizie (Q\$).

1470-1490 - selectarea unei ramuri de program în funcție de decizia luată.

1510-1540 - consecințe posibile ale deciziei luate în care aceasta a fost "bei apă".

1510 - în 1/3 din cazuri (RND 0.3) va rezulta o consecință negativă, iar în 2/3 din cazuri una-pozitivă.

1610 - situație specială de nivel 2: "crocodil" (în anumite cazuri dacă în prealabil s-a luat decizia de trecut mlaștina înot).

1630-1650 - descriere opțiuni (consecințe) ale apariției crocodilului (întîmplătoare).

1690 - situație specială: ciocnire de un copac, leu sau șarpe. Este în funcție de Y (1710, 1720, 1730).

1910-1930 - jucătorul a îndeplinit scopul jocului și a ieșit cu viață din junglă.

1940 - opțiune de joc nou.

1950 - jucătorul s-a apropiat de o așezare omenească.

1960-1970 - jucătorul se apropie de marginea junglei.

Comoara din peșteră

Un joc clasic de aventură în care jucătorul se plimbă într-un labirint (peșteră) încercînd să găsească o comoară ascunsă aici și să iasă cu ea afară din peșteră. Dar, pe lângă faptul că aceasta este păzită cu strășnicie de un uriaș, jucătorul are de înfruntat și o serie întregă de alte primejdii: pirați, lilieci, un personaj ciudat (Bill Bones), care nu se arată niciodată, dar trece din loc în loc și pe unde lasă urme nu se mai poate înainta, etc. Cu cît jucătorul înfruntă mai multe primejdii și reușește să ia comoara și să o scoată la lumină cu atît primește mai multe puncte. Pe de altă parte, cu cît va găsi un itinerar mai scurt, cu atît va scoate comoara dintr-un număr mai mic de pași.

Jocul se desfășoară printr-un dialog continuu între jucător (care ia decizii prin introducerea comenzilor) și calculator, care, după fiecare

Comoara din peșteră

comandă descrie situația și locul în care se află jucătorul. Comenzile disponibile pentru jucător sînt:

N - merge la nord; S - merge la sud; E - merge la est; V - merge la vest; U - urcare; C - coborîre; P - punctaj.

Deci are la dispoziție 6 comenzi pentru deplasare și o comandă prin care poate afla oricînd punctajul acumulat. Dacă se acționează o tastă care nu indică una din comenzile disponibile, atunci se va afișa din nou setul de comenzi disponibile, iar jucătorul se va menține în aceeași situație (loc).

În indicarea comenzilor, jucătorul are de înfruntat două tipuri de probleme. Primele sînt de natură logică. Dacă de exemplu, după o comandă de mers spre nord (N) apare mesajul: "Tunelul cotește. În ce direcție mergi?", în acest caz, jucătorul va trebui să indice ca direcție est (E) sau vest (V), dar în nici un caz nord (N) sau sud (S). Fapt aparent banal, care poate da însă bătăi de cap jucătorilor neexperimentați. Alte probleme sînt legate de orientarea într-un labirint, iar aceste probleme sînt mai dificile. De aceea considerăm necesar explicarea unor amănunte legate de geografia peșterii, precum și unele indicații referitoare la modul de abordare a jocului de către copii mai mici sau jucători neexperimentați.

Peștera este organizată pe 3 niveluri : parter (poate fi simbolizat cu 0), etaj (simbolizat cu 1) și subsol (simbolizat cu -1). De la un nivel la altul se poate trece numai prin anumite puncte. Unui punct de urcare de la parter la etaj (de exemplu), îi va corespunde un punct de coborîre de la etaj la parter. Comoara nu va fi ascunsă în același loc la fiecare joc.

Pentru copii mici recomandăm utilizarea hărții peșterii, iar pentru cei de 8-15 ani desenarea planului peșterii pe măsură ce jocul avansează. În acest fel se vor evita situațiile în care se merge în cerc (se trece de mai multe ori prin aceleași puncte) și se va forma deprinderea de a realiza planuri și hărți după descrieri.

Modificări posibile.

Deși în joc pare destul de complicată, peștera are numai 40 de locuri (încăperi), în plus existînd și suficientă memorie disponibilă pentru dezvoltarea jocului. Deci se poate crește numărul încăperilor, a etajelor, a situațiilor, etc. De asemenea, pe baza logicii programului se poate concepe un alt scenariu cu alte aventuri, în locul uriașului poate fi un balaur, în locul comorii poate fi o Ileană Cosînzeană furată, etc. Ar fi interesantă și posibilitatea adăugării graficii: tablouri pentru fiecare (sau unele), încăperi ale peșterii (lucru complicat, deoarece fiecare imagine ecran ocupă o memorie de aproape 8Ko) sau desenarea (reconstituirea) automată a planului (hărții) peșterii pe măsură ce ea este străbătută.

Descrierea programului

Linia 5 - dialog pe ecran fără mesajul "scroll?".

20 - rezervarea de spațiu de memorie pentru încăperile peșterii - variabila A, pentru încăperile (locurile) unde se pune comoara la începutul jocului (P) și a celor șapte comenzi disponibile (C\$).

30-40 - asignarea variabilelor pentru comenzile disponibile.

50 - inițializarea variabilelor: KT reprezintă numărul de comenzi (pași de program) pe care le-a dat jucătorul; K - reprezintă numărul încăperii (inițial se pornește de la 3, primii 3 pași fiind în pădure); PT reprezintă punctajul; INC - numărul încăperii în care se află jucătorul la un moment dat după indicarea unei comenzi; MC - numărul de încăperi.

60 - apelarea subrutinelor pentru cazul în care nu s-a indicat una din comenzile disponibile: GO SUB 300 - se vor afișa 4 rînduri goale fără mesajul "scroll?"; GO SUB 400 - se vor repeta comenzile disponibile.

66 - se citesc datele care indică locul în care este ascunsă comoara P(I). Acestea se interpretează astfel: prima oară comoara se ascunde în încăperea nr. 19, a 2-a oară în încăperea nr. 7, etc.

Comoara din peșteră

```

5 POKE 23692,255
10 BORDER 5: PAPER 5: INK 2: F
LASH 0: OVER 0: INVERSE 0: BRIGH
T 0: CLS
20 DIM A(40,8): DIM C$(7): DIM
P(10)
30 LET C$(1)="N": LET C$(2)="S
": LET C$(3)="E": LET C$(4)="V"
40 LET C$(5)="U": LET C$(6)="C
": LET C$(7)="P"
50 LET KT=0: LET K=3: LET PT=0
: LET INC=1: LET MT=26: LET MC=4
0
60 GO SUB 300: GO SUB 400: GO
SUB 300
66 FOR I=1 TO 10: READ P(I): N
EXT I
70 GO SUB 300
80 FOR I=1 TO MC: FOR J=1 TO 8
: READ A(I,J): NEXT J: NEXT I
85 LET A(P(INC),8)=10: LET CO=
0
90 LET T=A(K,7): GO SUB 500
100 PRINT ";TAB 5;"IN CE DIRECT
IE MERGI ?": POKE 23692,255: GO
SUB 102: LET A%=CHR$(CODE A%)
-32): PRINT A%
101 GO TO 110
102 PAUSE 0: LET A%=INKEY$
103 IF CODE A%(<99 OR CODE A%>11
8 THEN LET A%="-"
104 RETURN
110 PRINT : LET KT=KT+1
120 FOR I=1 TO 7
130 IF A%=C$(I) THEN GO TO 160
140 NEXT I
150 GO SUB 400: GO TO 100
160 IF I(<>7) THEN GO TO 170
162 IF PT(<20 THEN LET P%=" PUNC
TE.": GO TO 166
165 LET P%=" DE PUNCTE."
166 PRINT "AI ";PT;P%: GO SUB 3
00: GO TO 100
170 LET N=A(K,I)
180 IF N=0 AND CO(<>1) THEN PRINT
TAB 7;"NU POTI ";A%
190 LET K=K+N: LET T=A(K,7): LE
T C=A(K,8)
200 GO SUB 600: GO SUB 500: GO
TO 100
300 POKE 23692,255: FOR J=1 TO
4: PRINT : NEXT J: RETURN
400 PRINT " COMENZILE DISPONIB
ILE SINT : "
510 PRINT ";TAB 9;"N-NORD , S-
SUD"
420 PRINT ";TAB 9;"E-EST , U-VE
ST"
430 PRINT ";TAB 5;"U-URCARE , C
-COBORIRE"
440 PRINT ";TAB 11;"P-PUNCTAJ"
450 RETURN
500 GO TO 965
510 GO SUB 300
520 RETURN
600 IF C=0 THEN GO TO 640
610 IF INT (C/10)<>1 THEN GO TO
650
620 GO SUB 300: PRINT " FELICIT
ARI! Comoara este aici!" VRE
I SA O IEI ? (DA/NU)": PAUSE 0:
LET A%=INKEY$: PRINT
630 IF A%="D" OR A%="d" THEN LE
T CO=INC: LET PT=PT+50
640 LET PRU=0: RETURN
650 IF INT (C/10)<>2 THEN GO TO
700
660 IF CO=0 THEN GO TO 690
670 PRINT " Este un tunel i
ngust.?" Nu poti trece cu com
oara."
680 LET K=K-N: LET T=A(K,7): LE
T C=A(K,8)
690 LET PRU=0: RETURN
700 IF INT (C/10)<>3 THEN GO TO
790
710 IF CO=0 THEN GO TO 780
720 LET PRU=PRU+1
730 LET T=T+1: IF C-INT (C/10)*
10=5 THEN LET T=MT-1
740 IF PRU=1 THEN GO SUB 500
750 IF PRU>1 THEN GO TO 770
760 LET K=K-N: LET T=A(K,7): LE
T C=A(K,8): RETURN
770 LET T=T+1: GO SUB 500: LET
A(P(INC),8)=40: LET INC=INC+1: L
ET PT=PT-20: LET CO=0: LET PRU=0
: IF INC>10 THEN LET INC=1
775 LET A(P(INC),8)=10
780 RETURN
790 IF INT (C/10)=4 THEN PRINT
"Pe un billet scrie :"" Doar nu
crezi ca fac greseala sa o as
cund in acelasi loc!";TAB 20;"BI

```

```

LL BONES": GO SUB 300
800 IF INT (C/10)<5 OR CO=0 TH
EN GO TO 860
810 PRINT " FELICITAR
I!" AI REUSIT SA FURI COMOARA
DIN "I,KT;" PASII"
820 GO SUB 300: PRINT " Mai vr
ea cineva sa linceze ?";TAB 13;"
(D/N)": PAUSE 0: LET A$=INKEY$
830 IF A$("<"D" AND A$("<"d" THEN
CLS : STOP
840 FOR I=1 TO 10: LET A(P(I),B
)=0: NEXT I: LET PT=0: LET K=3:
LET INC=INC+1: LET CO=0: LET KT=
0: IF INC>10 THEN LET INC=1
845 LET A(P(INC),B)=10
850 RETURN
860 IF INT (C/10)=6 THEN LET PT
=PT+20
870 IF (INT (C/10)=7 AND (CO-IN
T (CO/2)*2=0)) THEN PRINT "Acea
sa este o zona in curs de creer
e interzisa vizitatorilor.": GO
TO 680
880 RETURN
890 DATA 19,7,39,5,27,16,10,36,
32,21
900 DATA 0,2,0,0,0,0,3,1,-1,1,-
1,-1,0,0,2,0,-1,1,-1,-1,0,0,1,50
,-1,2,4,1,0,0,4,0,0,0,-1,0,0,0,5
,0,-2,3,1,0,0,0,0,7,35
910 DATA 1,6,0,-1,0,0,8,0,0,-1,
0,-4,0,0,6,20,-3,2,3,1,20,0,9,0,
0,0,-1,0,0,0,5,0,-2,0,0,0,0,0,5,
0,0,0,0,-3,0,0,5,0
920 DATA -6,3,1,0,12,8,10,60,1,
0,0,-1,0,0,6,0,0,-1,0,1,0,0,6,0,
-3,0,-1,0,0,0,6,35,0,0,1,0,0,5,2
1,0,1,0,2,-1,0,0,22,0
930 DATA 0,-1,0,0,0,0,5,60,0,0,
0,-2,15,0,15,0,0,0,1,0,-8,0,15,0
,0,2,1,-1,-5,0,11,30,0,0,0,-1,0,
0,5,0,-2,0,0,0,0,5,0
940 DATA 0,1,13,0,0,12,23,0,-1,
0,0,1,0,0,6,0,0,1,-1,0,0,0,20,0,
-1,1,2,0,0,0,22,0,-1,0,0,0,0,-20
,24,35,0,0,1,-2,0,0,20,0
950 DATA 0,0,1,-1,0,0,19,60,3,0
,1,-1,0,0,14,60,1,0,0,-1,0,0,16,
70,0,-1,0,0,0,6,17,1,0,-3,0,1,0,
-15,14,0,1,0,-1,0,0,0,6,20
960 DATA 0,-1,0,1,0,0,20,0,1,0,
-1,-13,0,0,22,0,0,-1,0,0,0,0,5,0
,-36,-36,-36,-36,-36,-36,18,50
965 IF T<0 THEN GO TO 965+T*5
970 PRINT "Esti in padure ; int
rarea in pestera este in sud.
": GO TO 520
975 PRINT "Nu cred ca in felul
acesta vei gasi pestera.": GO T
O 520
980 PRINT "Te-ai ratacit in pad
ure.": GO TO 520
985 PRINT "Esti intr-o sala joa
sa.Spre nord se vede lumina.": GO
TO 520
990 PRINT "Tunelul se infunda."
: GO TO 520
995 PRINT "Esti intr-un tunel c
are coteste.": GO TO 520
1000 PRINT "Esti in sala piratil
or ; din fericire sint plecat
i.": GO TO 520
1005 PRINT "Tunelul merge N-S ;
exista o ramificatie spre V":
GO TO 520
1010 PRINT "Sala de tortura.Pe j
os sint imprastiate schelete
.": GO TO 520
1015 PRINT "Treci pe langa un ho
rn.": GO TO 520
1020 PRINT "Esti in sala uriasul
ui.Calca usor : doarme!": GO
TO 520
1025 PRINT "S-a trezit uriasul!
FUGII!": GO TO 520
1030 PRINT "Prichindelule,vrei s
a furi comoara? Mai incearc
a!": GO TO 520
1035 PRINT "Pe zid scrie: BILL B
ONES A FOST AICI.": GO TO 520
1040 PRINT "Esti la baza unui pu
t.": GO TO 520
1045 PRINT " Palatul de cr
ystal . Muzica izvoraste
de pretutindeni": GO TO 520
1050 PRINT " Ratacesti in Labir
intul Rosu.": GO TO 520
1055 PRINT " Bine ai veni
t!": GO TO 520
1060 PRINT "In fata ta este FLUV
IUL MORTII. Pe o tablita scrie:
XANADU...": GO TO 520

```

Comoara din peșteră

```
1065 PRINT "Tunelul se lărgeste.  
Sînt gauri în pereti.": GO TO 5  
20  
1070 PRINT "Esti în hruba liliec  
ilor. Grabeste-te!": GO T  
O 520  
1075 PRINT "Tunelul se ramifica.  
": GO TO 520  
1080 PRINT "S-a lasat o ceata de  
asa . Din cînd în cînd se aud  
tipete.": GO TO 520  
1085 PRINT "Esti în galeria came  
rei de tortura.Se vad urme  
insingerate.": GO TO 520  
1090 PRINT "Se aud piratii certi  
ndu-se.FUGII!": GO TO 520  
1095 PRINT "Aha! La tine este co  
moara! S-o punem la loc si  
mai vorbim! Nu fi prosti FU  
GII!": GO TO 520
```

80 - se citesc datele referitoare la încăperi A(I,J). Fiecărei încăperi îi sînt asociate date care indică posibilitațiile pentru fiecare comandă precum și numărul de puncte pe care îl obține jucătorul trecînd prin acest loc.

85 - variabila CO este un indicator pentru comoară. Dacă CO = 0 comoara nu este în posesia jucătorului; CO = 1 comoara este în

posesia jucătorului. Cînd jucătorul găsește și ia comoara primește un număr de puncte (la începutul jocului 50 de puncte).

100 - jocul începe cu o întrebare; se păstrează prima literă din răspuns (comandă) și se tipărește cu litere mari.

110 - numărul de comenzi date (pași) a crescut cu o unitate.

965 - afișarea unui mesaj pentru fiecare comandă în funcție de încăperea în care se găsește jucătorul.

970 - 1095 - mesajele.

890 - 960 - datele care descriu peștera.

890 - datele referitoare la numărul încăperii în care se pune comoara: prima oară în camera 19, a 2-a oară în camera 7, etc.

900 - 960 - descrierea încăperilor peșterii. Fiecărei încăperei îi sînt asociate cîte 8 date (deci linia 900 conține descrierea primelor 6 camere) în următoarea ordine: Nord, Sud, Est, Vest, Urcare, Coborîre, numărul mesajului asociat fiecărei încăperi, punctajul pentru fiecare încăpere sau 10 în cazul în care comoara se află în încăperea respectivă. Pentru o deplasare datele se interpretează după

formula: $INC = INC + data$. Astfel data 0 înseamnă că nu se poate merge în direcția asociată (INC - numărul încăperii rămîne același, deci jucătorul se va găsi în aceeași încăpere). De exemplu, cele 8 date asociate primei încăperi se citesc astfel:

0 - la Nord nu se poate merge;

2 - dacă se indică Sud, la numărul camerei actuale se adaugă 2 și se obține numărul camerei în care se ajunge;

0 - la Est nu se poate merge;

0 - la Vest nu se poate merge;

0 - nu se poate Urcare;

0 - nu se poate Coborîre;

3 - mesajul asociat camerei actuale este cel cu numărul 3.

O dată negativă, de exemplu, prima dată (-1) pentru datele asociate celei de-a doua încăperi se va interpreta astfel: numărul încăperii în care se ajunge dacă se va da comanda asociată (Nord) se va găsi adunînd -1 la numărul camerei actuale (2). Deci jucătorul va ajunge în camera 1. A 24-a dată, (50), indică numărul de puncte obținut dacă se trece prin această cameră (3 deoarece este ultima dată din al 3-lea set de date).

JOCURI PENTRU SCOALA


Simultan

Este un joc cu ajutorul căruia copiii își pot dezvolta deprinderea de a efectua rapid calcule matematice, și, mai mult, de a identifica anumite numere care satisfac simultan mai multe egalități. Se poate spune că acest tip de joc va ajuta copiii (mai târziu) în problemele legate de rezolvarea sistemelor de ecuații și a modelelor matematice.

Pe ecran se afișează un caroiaj de 5*5 pătrate, din care 4 pătrate sînt blocate (colorate), iar 12 au înscrise în ele semnele matematice corespunzătoare adunării, scăderii sau înmulțirii. Rămîn de completat de către jucător 9 pătrate cu cifrele care vor rezolva simultan ecuațiile reprezentate. Rezultatul fiecărei ecuații se găsește pe linia și pe coloana respectivă în afara caroiajului.

□	-	□	X	□	= -27
X	■	X	■	+	
□	X	□	+	□	= 9
-	■	-	■	-	
□	-	□	□	□	= -22
=		=		=	
37		0		6	

Un exemplu de situație inițială a jocului se poate urmări în figură:

În vederea rezolvării ecuațiilor se deplasează cursorul  (el apare inițial pe prima poziție a caroiajului și clipește) cu ajutorul tastelor (D - cursor dreapta, S - cursor stînga, A - cursor sus, Z - cursor jos) și se înscrie sau modifică cifre în pătratul pe care este poziționat cursorul, încercîndu-se prin modificarea făcută ca egalitățile să fie satisfăcute atît pe orizontală cît și pe verticală.

Scopul jocului este de a indica rezultatul exact (completarea pătratelor libere cu cifre corespunzătoare) într-un interval cît mai mic de timp. Inițial sînt 3000 de puncte. Numărul acestora va scădea proporțional cu timpul consumat pentru aflarea soluției. Dacă soluția nu este descoperită se poate acționa tasta K și rezolvarea va apare în colțul din dreapta jos al fiecărui pătrat. Dacă soluția este bună, jucătorul va rămîne cu punctele din acel moment. Punctele se afișează permanent în partea de jos a ecranului.

Jucătorul are posibilitatea la începutul jocului de a opta pentru varianta de joc în care cifrele să se poată repeta sau cifrele să nu se poată repeta. La sfîrșitul unui joc se poate introduce opțiunea de a se juca în continuare. Dacă au fost rezolvări corecte și jucătorul nu mai dorește să se mai joace în continuare, se va afișa numărul de puncte obținut în medie pe un joc.

Descrierea programului

80 - respingerea răspunsurilor nelegale

100 - rezervarea de spațiu de memorie pentru variabile.

Variabila C va memora cifrele (vor fi cîte 3 cifre pe fiecare linie sau coloană); variabila S va memora semnele generate (vor fi 12 semne generate aleator), variabila R va memora rezultatele (vor fi 9 rezultate, cîte căsuțe libere sînt), variabila T va memora răspunsurile introduse în căsuțele libere.

```

10 BORDER 7: PAPER 7: INK 0: C
LS
70 PRINT AT 21,0;"Cifrele sa s
e repete(D/N)?": LET P$=INKEY$
80 IF P$("<"D" AND P$("<"N" AND
P$("<"d" AND P$("<"n" THEN GO TO 7
0: CLS
85 CLS
100 RANDOMIZE : DIM C(3,3): DIM
S(12): DIM I(3,3): DIM F(3): DI
M R(9): DIM T(9)
110 LET A$="+-x": LET JOC=1: LE
T SUMA=0
130 FOR K=48 TO 168 STEP 24
135 BEEP .01,K/8
140 PLOT K,39: DRAW 0,120
150 PLOT 48,K-9: DRAW 120,0: NE
XT K
160 FOR K=0 TO 1: FOR L=1 TO 2
170 FOR M=5 TO 7: FOR N=3 TO 5
180 PRINT AT 6*K+M,6*L+N;"<CAPS
8>": NEXT N: NEXT M: NEXT L: NE
XT K
205 FOR K=1 TO 9: LET T(K)=0: N
EXT K
210 LET PCT=3000: FOR K=1 TO 3:
FOR L=1 TO 3
215 BEEP .01,K*10-L*15
220 LET C(K,L)=INT (RND*9)+1
230 IF P$="N" OR P$="n" THEN GO
TO 930
240 PRINT AT 6*L-3,6*K+1;" ":
PRINT AT 3+(K-1)*6,22;"="
@50 PRINT AT 6*L-2,6*K+1;" ":
PRINT AT 18,7+(K-1)*6;"="
260 NEXT L: NEXT K
280 FOR K=1 TO 12
290 LET S(K)=INT (RND*3)+1: NEX
T K
300 FOR L=1 TO 2: FOR K=1 TO 3
305 BEEP .01,K*10-L*15
310 PRINT AT 3+(K-1)*6,10+(L-1)
*6;A$(S(L+(K-1)*3)): NEXT K: NEX
T L
320 FOR L=1 TO 3: FOR K=1 TO 2
325 BEEP .01,L*15-K*10
330 PRINT AT 6+(K-1)*6,7+(L-1)*
6;A$(S(2+(K-1)*5+L)): NEXT K: NE
XT L
350 FOR K=1 TO 3
355 BEEP .1,RND*30-40
360 LET B1=S(1+(K-1)*5): LET B2
=S(2+(K-1)*5)
370 FOR L=1 TO 3
380 LET F(L)=C(K,L): NEXT L
390 GO SUB 860
395 PRINT AT 3+(K-1)*6,24;"
"
400 PRINT AT 3+(K-1)*6,24;R(K):
NEXT K
410 FOR K=1 TO 3
415 BEEP .1,RND*30-40
420 LET B1=S(K+2): LET B2=S(K+7
)
430 FOR L=1 TO 3
440 LET F(L)=C(L,K): NEXT L: LE
T K=K+3
450 GO SUB 860
455 PRINT AT 20,11-(K-1)*6;"
"
460 PRINT AT 20,11-(K-1)*6;R(K)
: LET K=K-3: NEXT K
480 LET Y=4: LET X=8
490 PRINT AT Y,X;"<CAPS 4>": PA
USE 5
495 BEEP .01,10
500 PRINT AT Y,X;" "
510 LET X$=INKEY$
520 LET X=X+6*(X$="d")*(X(16)-
(X$="s")*(X)9))
530 LET Y=Y+6*(X$="z")*(Y(14)-
(X$="a")*(Y)9))
540 LET VX=CODE X$-48: IF VX=0
AND VX(<=9 THEN LET I((Y+2)/6,(X-
2)/6)=VX: PRINT AT Y-1,X-1;VX
560 PRINT AT 21,0;"Puncte:";PCT
;"Gata-apasa K"
570 LET PCT=PCT-1: IF PCT=0 THE
N GO TO 800
580 IF X$="K" OR X$="k" THEN GO
TO 600
590 GO TO 490
610 FOR K=1 TO 3
620 LET B1=S(1+(K-1)*5): LET B2
=S(2+(K-1)*5)
630 FOR L=1 TO 3: LET F(L)=I(K,
L): NEXT L: LET K=K+6
640 GO SUB 860
650 IF R(K)<>R(K-6) THEN GO TO
800
660 LET K=K-6: NEXT K
670 FOR K=1 TO 3

```

```

680 LET B1=S(K+2): LET B2=S(K+7)
)
690 FOR L=1 TO 3: LET F(L)=I(L,
K): NEXT L: LET K=K+6: GO SUB 84
0
710 IF R(K)<>R(K-3) THEN GO TO
800
715 LET K=K-6: NEXT K
730 LET SUMA=SUMA+PCT
740 PRINT AT 21,25;"
750 PRINT AT 17,0;"
" BRAVDIAI:";PCT;" puncte!";"I
n medie, in ";JOC;" jocuri ai:";S
UMA/JOC;" pct/joc!"
770 PRINT AT 21,0;"Mai joci o d
ata?(D/N) " : LET Z#=INKEY
$: IF Z#("<"D" AND Z#("<"N" AND Z#
("<"d" AND Z#("<"n" THEN GO TO 770
780 IF Z#="N" OR Z#="n" THEN CL
S : PRINT "Sper ca ti-a placut..
": BEEP 2,RND*24: STOP
790 LET JOC=JOC+1: PRINT AT 19,
": GO TO 190
810 PRINT AT 21,0;"Din pacate,r
au.Iata rezolvarea:"
820 FOR K=1 TO 3: FOR L=1 TO 3
830 PRINT AT 4+(K-1)*6,8+(L-1)*
6;C(K,L)
840 NEXT L: NEXT K
845 FOR K=-30 TO 30 STEP 5: BEE
P .1,K: NEXT K
850 GO TO 760
870 IF B1=3 THEN LET W=F(1)*F(2
): GO TO 900
880 IF B2=3 THEN LET W=F(2)*F(3
): GO TO 890
885 GO TO 910
890 LET R(K)=(F(1)+W*((B1=1)-(B
1=2))): RETURN
900 LET R(K)=(W+F(3))*((B2=1)-(B
2=2))*((B2<>3)+(F(3)*W)*((B2=3):
RETURN
910 LET W=F(1)+F(2)*((B1=1)-(B1
=2))
920 LET R(K)=W+F(3)*((B2=1)-(B2
=2)): RETURN
940 IF T(C(K,L))=1 THEN GO TO 2
20
950 LET T(C(K,L))=1: PRINT AT 1
3,0;"SCRIE""CIFRE""DIFE-""RIT
E!": GO TO 240 .

```

110 - inițializarea unor variabile; A\$ - semnele pentru operații joc - va reprezenta numărul jocului (inițial 1). Suma va reprezenta suma punctelor acumulate (inițial este 0).

130 - 180 - desenarea tablei de joc.

205 - alegerea cifrelor de către calculator.

210 - punctajul pentru un joc, inițial este 3000.

220 - generarea aleatoare a cifrelor întregi.

230 - dacă răspunsul privitor la întrebarea "cifrele să se repete?" a fost N (nu) se vor alege cifre diferite de către calculator.

280 - 290 - alegerea semnelor.

340 - scrierea semnelor.

350 (GO SUB 860) apelarea subrutinei de calcul a rezultatelor.

870 - 900 calcul rezultate.

870, 880 - calcul rezultate parțiale.

890, 920 - calcul rezultate R.

400 - scrierea rezultatelor.

460 GO SUB 860 apelarea subrutinei de calcul a rezultatelor și scrierea rezultatelor.

480 - 540 - bucla principală de introducere a datelor.

480 - stabilirea coordonatelor inițiale de afișare a cursorului.

510 - 530 - stabilirea coordonatelor de afișare a cursorului dacă se introduce o comandă de deplasare a sa (tastele D, Z, S sau A).

560 - afișarea mesajului din partea de jos a ecranului.

570 - la fiecare trecere prin buclă punctajul scade cu un punct.

Dacă se ajunge la 0 se trece la indicarea soluției (ca și cum s-ar fi dat un răspuns greșit).

580 - dacă se acționează tasta K se indică soluția.

610 - rezolvare terminată.

640 - se apelează subrutina de calcul a rezultatelor.

650 - dacă rezultatul nu coincide cu soluția se va afișa mesajul și se va indica soluția.

730 - 750 - rezultatul pentru rezolvare corectă.

760 - joc nou?

810 - 840 - rezolvare corectă (soluția).

890 - joc nou?

Animale

Un joc pe care copiii îl joacă deseori între ei, sub numele de "ghicește animalul". Un jucător se gîndește la un animal, iar ceilalți încearcă să-l "ghicească" punînd întrebări și obținînd astfel informațiile necesare pentru identificarea lui. Cu cît întrebările vor fi puse mai inteligent, restrîngînd mai rapid clasa din care face parte animalul, cu atît numărul de încercări din care acesta va fi identificat va fi mai mic. Caracterul instructiv al jocului rezultă din învățarea, pe această cale, a unor noi specii de animale precum și a caracteristicilor importante prin care animalele se deosebesc între ele (mediul și locul în care trăiesc, cum se înmulțesc, dacă este domestic, caracteristici fizice definitorii etc).

În jocul simulat pe calculator, acesta are rolul de a "ghici" animalul. Deci jucătorul va fi invitat să se gîndească la un animal. Apoi calculatorul pune diverse întrebări (trăiește la noi?, trăiește în apă?, are blană?, are pene?, este domestic?, etc), la care jucătorul va răspunde prin "da" sau "nu" (se poate răspunde și numai prin indicarea primei litere a răspunsului "d" sau "n"). După mai multe întrebări calculatorul va indica un animal. Dacă răspunsul a fost corect (este chiar animalul la care se gîndește jucătorul), în urma opțiunii jucătorului, se va putea începe (sau nu) alt joc. Dacă răspunsul nu a fost corect, atunci calculatorul va cere jucătorului să formuleze o întrebare care deosebește animalul indicat de animalul care trebuie ghicit. Astfel programul are o caracteristică foarte importantă: el poate învăța lucruri noi, adică își adaugă animalul care trebuia ghicit printre animalele pe care le cunoștea deja. În acest fel programul își poate îmbogăți neîncetat zestrea de cunoștințe ajungînd în timp să reprezinte pentru un jucător un adevărat atlas zoologic.

Dacă la întrebarea de continuare a jocului jucătorul răspunde cu "nu" (sau "n") calculatorul va întreba dacă se dorește înregistrarea programului (jocului) cu toate animalele pe care le cunoaște. Dacă jucătorul răspunde afirmativ, va apare mesajul "Start cass and press any key": se va poziționa caseta magnetică. În dreptul unei porțiuni neînregistrate, se vor acționa clapele casetofonului pentru înregistrare și apoi orice tastă a calculatorului. În acest fel programul se va înregistra (salva) din nou, dar cînd se va încărca cu altă ocazie pentru a se juca jocul, va "cunoaște" toate animalele pe care le știa în momentul înregistrării.

Două aspecte noi intervin în acest joc față de jocul tradițional (fără calculator). Primul este legat de achiziționarea de cunoștințe și, pe această bază, a dezvoltării deprinderii de arhivare și sistematizare a informațiilor (la copii), iar al doilea este legat de faptul că jucătorul interpretează în timpul jocului atît un rol (aparent) pasiv (specific

jucătorului care se gîndește la un animal), cît și un rol activ (specific jucătorului care "ghicește" animalul). Într-adevăr, în momentul în care jucătorul este solicitat de către calculator să pună o întrebare care face o deosebire esențială între două animale, rolul său devine activ, semănînd într-o mare măsură cu rolul jucat de calculator.

Nu recomandăm modificări ale jocului (cu toate că s-ar putea realiza astfel de jocuri pe diferite specii de animale: pești, păsări, animale sălbatice, animale domestice, mamifere, etc.), în schimb recomandăm adăugarea de cît mai multe animale în baza de cunoștințe a programului (programul va putea "înmagazina" pînă la 100 de animale).

Descrierea programului

În realizarea programului s-au folosit tehnici specifice inteligenței artificiale. Astfel pentru structurarea datelor (care constau din nume de animale sau întrebări) s-a utilizat un arbore de tip binar.

Acesta este format din noduri, fiecărui nod fiindu-i asociată o întrebare (dacă este un nod intermediar) precum și posibilitatea unei ramificații corespunzătoare răspunsului pozitiv sau negativ la întrebare (de unde și denumirea de arbore binar). Arborele pornește de la un nod inițial (rădăcina arborelui) și se continuă cu noduri moștenitoare. Nodurile sînt numerotate: rădăcina este nodul 1, pe următorul nivel (primii moștenitori) sînt nodurile 2 și 3, iar pe următorul nivel nodurile 4, 5, 6 și 7 (moștenitorii nodurilor 2 și respectiv 3) și așa mai departe. În program arborele este descris în variabila $a(nq,j)$ (vezi linia 15), în care nq reprezintă numărul nodului "tată", iar j poate fi 1 sau 2, după cum nodul este pe ramura din partea dreaptă. Modul de creștere a arborelui este următorul:

Dacă nodul i este un nod "tată", atunci moștenitorii săi vor fi nodurile k și l , iar acestea vor avea asociate informațiile (pot fi întrebări sau animale) $a(i,1)$ pentru nodul moștenitor de pe ramura stîngă și $a(i,2)$ pentru nodul moștenitor de pe ramura dreaptă. Dacă un nod nu are moștenitori, atunci el va reprezenta în arbore un nod frunză. Aceste noduri au ca informații asociate animale, iar informațiile asociate moștenitorilor au valoarea 0.

Programul parcurge arborele de sus în jos (de la rădăcină spre frunze) și prin intermediul răspunsurilor la întrebări va ajunge la o frunză (deci un animal), existînd în continuare posibilitatea de a se adăuga un nou animal, frunza transformîndu-se în nod intermediar (i se asociază întrebarea care deosebește animalul care trebuia ghicit de cel indicat) iar animalul care era în frunză, mutîndu-se într-un nod frunză, pe un nivel superior.

Întrebările sînt valabile numai pentru o ramură, parcurgerea arborelui făcîndu-se de sus în jos, fără posibilitatea de întoarcere. Din această cauză este posibilă existența aceleiași întrebări în mai multe noduri intermediare (care nu se află pe aceeași ramură) în cadrul arborelui, cum se poate observa și în construcția programului prezentat, de exemplu, cu întrebarea: "trăiește în apă?". Acest lucru este logic, deoarece atît animalele care trăiesc la noi, cît și cele care nu trăiesc la noi pot avea (sau nu) ca mediu de viață apa.

Linia 10 - variabila nq reprezintă numărul maxim de întrebări și animale pe care le poate "ține minte" programul. În exemplul nostru s-a luat un număr "rotund" (100), dar acesta se poate modifica pînă la 635.

15 - rezervarea de spațiu de memorie pentru întrebări și nume de animale. Variabile: întrebarea pentru nodul i se află în $a\$(i)$. Lungimea maximă a întrebării poate fi de 50 de caractere; $a(i,1)$ pointer stînga pentru nodul i ; $a(i,2)$ pointer dreapta pentru nodul i ; $r\$(1)$ răspunsul de la tastatură, da sau nu.

Animale

```

10 LET nq=100
15 DIM q$(nq,50): DIM a(nq,2):
DIM r$(1)
20 LET qf=22
30 FOR n=1 TO qf/2-1
40 READ q$(n): READ a(n,1): RE
AD a(n,2)
50 NEXT n
60 FOR n=n TO qf-1
70 READ q$(n): NEXT n
110 PRINT "'Gindeste-te la un
animal.'"; PRINT #1;"'Apasa ori
ce tasta"; BEEP 0.1,10
120 PAUSE 0: BEEP 0.05,-5
130 LET c=1
140 IF a(c,1)=0 THEN GO TO 300
150 LET p=q$(c): GO SUB 910
160 PRINT "?": GO SUB 1000
170 LET in=1: IF r$="d" THEN GO
TO 210
180 IF r$="da" THEN GO TO 210
190 LET in=2: IF r$="n" THEN GO
TO 210
200 IF r$("<nu") THEN GO TO 150
210 LET c=a(c,in): GO TO 140
310 PRINT "Te gindesti la"
320 LET p=q$(c): GO SUB 900: P
RINT "?"
330 GO SUB 1000
340 IF r$="d" OR r$="da" THEN G
O TO 400
350 IF r$="D" OR r$="DA" THEN G
O TO 400
360 IF r$="n" OR r$="nu" THEN G
O TO 500
370 IF r$="N" OR r$="NU" THEN G
O TO 500
380 PRINT "Raspunde-mi clar cin
d vorbesc "'cu tine!": GO TO
300
410 PRINT "M-am gindit atita!":
GO TO 800
510 IF qf>nq-1 THEN PRINT "Sint
sigur ca animalul tau este foar
te","interesant,dar nu mai am de
stul loc acum.": GO TO 800
520 LET q$(qf)=q$(c)
530 PRINT "Ce este atunci?": IN
PUT q$(qf+1)
540 PRINT "Pune o intrebare car
e face o'" deosebire intre ";
550 LET p=q$(qf): GO SUB 900:
PRINT " si ";
560 LET p=q$(qf+1): GO SUB 900
: PRINT
570 INPUT s$: LET b=LEN s$
580 IF s$(b)="" THEN LET b=b-1
590 LET q$(c)=s$( TO b): REM in
sereaza intrebaraa
600 PRINT "'Care este raspunsul
la"
610 LET p=q$(qf+1): GO SUB 900
: PRINT "?"
620 GO SUB 1000
630 LET in=1: LET io=2
640 IF r$="d" OR r$="da" THEN G
O TO 700
650 IF r$="D" OR r$="DA" THEN G
O TO 700
660 LET in=2: LET io=1
670 IF r$="n" OR r$="nu" THEN G
O TO 700
680 IF r$="N" OR r$="NU" THEN G
O TO 700
690 PRINT "Incearca sa fi mai p
recis!": GO TO 600
710 LET a(c,in)=qf+1: LET a(c,i
o)=qf
720 LET qf=qf+2
725 GO TO 730+10*INT (RND*5)
730 PRINT "Asta ma ameteste. ":
GO TO 800
740 PRINT "Nu m-as fi gindit !"
: GO TO 800
750 PRINT "Foarte interesant.":
GO TO 800
760 PRINT "Daca spui tu sigur e
asa.": GO TO 800
770 PRINT "Aflu lucruri noi.":
GO TO 800
810 LET r$=INKEY$: PRINT #1;"In
ca un joc?": BEEP 0.5,2: LET r
$=INKEY$: BEEP 0.2,6: PAUSE 50:
LET r$=INKEY$: PAUSE 50
815 LET r$=INKEY$: PRINT : PRIN
T r$: PRINT : IF r$="" THEN GO T
O 815
816 INPUT ""
820 IF r$("<n" AND r$("<N" THEN
GO TO 100
840 INPUT "Vrei sa salvezi prog
ramul cu toate animalele? "; r
$

```

```

850 IF r$(1)="d" THEN INPUT "Nu
mele programului: ";r$: SAVE r$
LINE 100
860 STOP
905 PRINT " ";
910 FOR n=50 TO 1 STEP -1
920 IF p$(n)<>" " THEN GO,TO 93
5
930 NEXT n
935 IF p$(1)>>" " THEN LET p$(1)
=CHR$(CODE p$(1)-32)
940 PRINT p$( TO n);: RETURN
1010 PDKE 23658,16: BEEP 0.05,20
: INPUT LINE r$: IF r$="" THEN G
O TO 1010
1020 LET r$=r$(1): RETURN
2010 DATA "Traieste la noi",2,3
2020 DATA "Traieste in apa",4,5
2030 DATA "Traieste in apa",6,7
2040 DATA "Are solzi",11,12
2050 DATA "Are blana",8,13
2060 DATA "Are solzi",14,15
2070 DATA "Are pene",9,10
2080 DATA "Este un animal domest
ic",20,21
2090 DATA "Zboara",16,17
2100 DATA "Are coada stufoasa",1
8,19
2110 DATA "o stiuca","un crab","
o vrabie","un rechin","o balena"
,"o agvila","un pinguin","un sco
ncs","un elefant","o vaca","o vu
lpe"

```

20 - variabila gf reprezintă numărul inițial de noduri ocupate în arbore. Așa cum este construit programul se pornește cu 21 de întrebări și răspunsuri (10 întrebări și 11 animale, totdeauna numărul de animale - frunze - fiind cu o unitate mai mare decât numărul de noduri intermediare).

30-50 - se încarcă nodurile (se citesc întrebările și pointerii din stînga și din dreapta).

60-70 - se încarcă frunzele.

110 - începerea jocului.

130 - variabila c reprezintă numărul nodului. Se începe parcurgerea arborelui de la rădăcină (cu o întrebare).

140 - dacă $a(c,1)=0$ înseamnă că s-a ajuns la o frunză, dacă nu, înseamnă că este o întrebare.

150 - se salvează întrebarea asociată nodului în variabila ps. Apelarea subrutinei (GO SUB 910) prin intermediul căreia se tipărește întrebarea fără blaturi la coadă.

Animale

160 - se apelează subrutina pentru citirea răspunsului (GO SUB 1000).

1010 - numai litere mici

160 - se apelează subrutina pentru citirea răspunsului (GO SUB 1000).

170-200 - se selectează ramura. Indicatorul in poate fi 1 (s-a răspuns prin "da", ramura stîngă), sau 2 (s-a răspuns prin "nu", ramura dreaptă).

210 - se trece la nodul următor și procedeul se repetă (GO TO 140).

310 - s-a ajuns într-o frunză, deci la un animal.

320 - se tipărește animalul la care s-a ajuns (GO SUB 900 - fără spații la coadă).

330 GO SUB 1000 - se apelează subrutina pentru citirea răspunsului.

340-350 - dacă s-a ghicit se trece la opțiunea pentru alt joc sau salvarea programului (800).

360-370 - dacă nu s-a ghicit se trece la inserarea unui nou animal prin lungirea arborelui.

510 - dacă nu mai este loc pentru un nou animal se trece la opțiunile pentru sfîrșit joc (800).

520 - introducerea numelui noului animal; numărul de noduri ocupate din arbore (gf) crește cu o unitate.

590 - inserează întrebarea.

630-680 - în funcție de răspuns, ramurile se schimbă între ele.

710 - inițializează efectiv nodurile, reface răspunsurile.

720 - rezervă următorul spațiu liber pentru animal.

725 - alegerea aleatoare a unuia din cele cinci mesaje.

730-770 - mesaje de afișat.

810 - opțiuni de final de joc.

816 - șterge mesajul de pe liniile din partea de jos ale ecranului.

2010 - 2110 - datele privind întrebările și numele animalelor.

2010 - nodul 1 și moștenitorii săi (nodul 2 și nodul 3).

2020 - nodul 2 și moștenitorii săi (nodul 4 și nodul 5).

2080 - nodul 15 și moștenitorii săi (20 și 21).

2110 - numele animalelor (din noduri frunze)

JOCURI PE CASETE

Prezentăm în continuare câteva dintre cele mai răspândite și mai reprezentative jocuri realizate pentru calculatoare compatibile SINCLAIR SPECTRUM. În general prezentarea ține seama de descrierea jocului (scenariu) punându-se în special accentul pe utilizare (taste folosite în defășurarea jocului). În anumite cazuri se examinează posibile modificări ale jocurilor prin utilizarea comenzii POKE.

BRUCE LEE - joc de îndemânare și reflexe

Personajul principal (care poate fi și celebrul caratist Bruce Lee) este atacat de doi indivizi și se va apăra prin lovituri de mâini sau picioare, încercînd în același timp, să evite loviturile pe care aceștia încearcă să i le aplice. Jucătorul va tasta comenzile, astfel încît personajul principal să facă față celor doi și să culeagă toate florile din cele trei încăperi pe unde poate trece. Dacă reușește să culeagă florile, atunci va putea trece printr-o poartă (sub o scară), jucătorul avînd posibilitatea să reia jocul la un grad sporit de dificultate. Florile stau atîrnate de grinzii și se pot culege prin sărituri înspre ele.

Dacă cei doi îl lovesc pe Bruce Lee de trei ori, acesta pierde una din viețile pe care le are la dispoziție. Dacă le pierde pe toate, jocul ia sfîrșit.

Utilizare

Jocul se încarcă cu comanda: LOAD "" și se lansează în execuție automat.

Taste folosite:

ENTER (CR) - pornire joc sau întrerupere temporară (în timpul jocului);

Q - sare (pentru ca să culeagă flori sau să se urce pe scară);

A - coboară (de pe scară și se culcă la pământ;

O - merge la sînga;

Oricare din tastele de pe rîndul al 4-lea de la Z la M lovește cu mîna (cînd lovește stă pe loc).

Q sau P - face cîțiva pași și apoi Z-M - lovește cu piciorul (la stînga sau la dreapta);

Q apoi O sau P în același timp - sare într-o parte (ca să meargă mai repede sau să ia flori din zbor).

BREAK - începere joc fără prezentare;

SS - cu sau fără sunet.

Cîteva modificări care se pot realiza:

POKE 51795,0 pentru un număr nelimitat de vieți.

Kung Fu - joc de îndemînare și reflexe

KUNG FU este un joc care simulează o luptă KUNG FU între doi jucători. Aceștia pot lovi și se pot apăra cu mîinile și picioarele, apărarea realizîndu-se atît prin pararea loviturilor cu mîinile și picioarele, cît și prin evitarea loviturilor prin sărituri (în față și spate), aplecări, etc.

Jocul se poate desfășura în două variante după cum se selectează inițial opțiunea:

a) între jucător și calculator: jucătorul va conduce unul dintre luptători (cel din stînga ecranului) prin acționarea tastelor, celălalt luptător fiind condus de calculator.

b) între doi jucători oponenti care vor acționa în mod independent tastele dînd comenzi luptătorului pe care îl conduce.

Kung Fu - joc de îndemînare și reflexe

În cazul a) puterile adversarului se afișează, în partea dreaptă jos a ecranului, sub forma unor mingii, care dispar atunci cînd luptătorul primește lovituri.

Puterile luptătorului condus de către jucător se înegresc pe măsură ce scad, începînd să clipească atunci cînd sînt aproape epuizate.

În cazul b) cînd un luptător va cădea, jocul se termină. Puncte nu se dau.

Utilizare

Încărcarea se face cu LOAD ""; iar lansarea în execuție este automată.

Taste folosite:

a) pentru joc împotriva calculatorului (tasta 4 la început). (Comenzile se referă la jucătorul din stînga):

- 0 - pornire joc;
- 4 - lovitură picior stîng la cap;
- 3 - lovitură picior drept la corp;
- 2 - lovitură pumn drept;
- 1 - lovitură (carată) mîină stîngă;

BREAK - înainte;

SS - înapoi;

b) joc de doi jucători (tasta 5 la început):

0 - pornire joc;

Taste jucător stînga	Taste jucător dreapta	
4	0	lovitură picior stîng la cap
3	9	Lovitură picior drept la corp
2	8	lovitură pumn drept
1	7	carată mîină stîngă
z	SS	înainte
CS	BREAK	înapoi

Robin - joc de strategie, îndemînare și reflexe

Legendarul personaj Robin Hood, haiducul pădurilor britanice înfruntă mai multe primejdii și trece prin diferite aventuri. Jucătorul se va identifica cu personajul principal pe care îl ajută să depășească primejdiile.

Jocul poate începe cînd a) Robin este în închisoare (de aici trebuind, bineînțeles, să evadeze) sau cînd b) Robin este în codru (un labirint de poteci din care Robin trebuie să iasă fără ca următorii - puterile - să-l prindă).

În codru, Robin este urmărit de soldați înarmați cu arbalete. El se va apăra lovindu-i cu toiagul, singura sa armă. Pot apărea și puternicii cîini de vînătoare ai autorităților, pe care Robin nu poate să îi omoare, ci numai să se ferească de ei (în sus sau în jos).

În drumul său, Robin poate întîlni mai multe lucruri și personaje:

- tolba cu săgeți, pe care o poate lua (dar numai dacă nu mai are și altă tolbă);

- floricelele care se pot lua de fiecare dată;

- coroniță cu lauri, care dacă se va lua, Robin va căpăta o viață. (La începutul jocului are o singură viață);

- omul cu arcul (poate apare după ce s-a luat coronița cu lauri), care îi indică lui Robin numărul de vieți pe care le are;

- un popă (este păzit de un soldat, deoarece are asupra sa o mare sumă de bani). Dacă Robin omoară soldatul păzitor, popa va fugi lăsînd în urmă două pungii cu bani pe care Robin le poate lua;

- o cheie, care trebuie luată, deoarece îi va fi necesară lui Robin, în închisoare, pentru a evada.

Obiectele se pot lua prin deplasarea lui Robin de jos în sus (înspre obiecte), iar pungile cu bani se pot lua oricum deplasîndu-l pe Robin înspre ele).

În drumul său Robin trebuie să fie foarte atent și să ia repede florile, coronița, pungile cu bani, etc., deoarece poate apărea contesa cea rea care va lua toate florile sau o pungă cu bani. De asemenea, își poate face apariția guvernatorul, ai cărui soldați îl vor prinde pe Robin și îl vor băga la închisoare.

Utilizare

Jocul se încarcă cu LOAD " " și se lansează automat, apărînd pe ecran următoarea listă de opțiuni:

0 START

1 keyboard

2 kempston

3 interface 1

4 cursor

Se va acționa tasta 1, corespunzătoare utilizării jocului cu tastatura (keyboard) și va apărea o altă listă de opțiuni, corespunzătoare tastelor utilizate în timpul jocului:

1 - lovitură;

Q - sus;

A - jos;

N - stînga;

M - dreapta;

Tastele se pot modifica la dorința utilizatorului.

Jocul va începe cînd se va acționa tasta 0 (START).

Cîteva modificări care se pot realiza:

POKE 48690,0 pentru un număr nelimitat de vieți (nemurire);

POKE 55857,0

POKE 57827,0

POKE 57828,0

POKE 57151,0

POKE 57152,0

Skool Daze - joc de îndemînare și reflexe

Personajul principal (Eric) este condus într-o școală, cu ajutorul diverselor comenzi date prin tastare. Scopul este de a se realiza cît mai multe puncte. Pentru aceasta, Eric are o mare libertate de acțiune (poate da pumni elevilor mai mici, poate sări sau trage cu praștia, etc.), dar toate acestea fără a fi văzut de vreun profesor. Dacă însă, profesorul observă vreo neregulă, atunci el va trage linii la palmă (în funcție de gravitatea faptei), iar dacă Eric primește mai mult de 1000 de linii la palmă, jocul se termină, jucătorul terminînd cu punctele obținute pînă în acest moment.

Personajele jocului sînt prezentate ca într-o piesă de teatru:

Eric - eroul nostru

Boy Wonder - Fugărețul

Angelface - Boacănă

Einstein - Savantul

MR. Wacker - directorul școlii

MR. Rockitt - prof(esor)ul de științe

MR. Withit - prof(esor)ul de geografie

MR. Creak - prof(esor)ul de istorie

Jocul se încarcă cu comanda LOAD "", iar lansarea în execuție este automată.

Iată tastele utilizate și regulile jocului:

5 - mers la stînga;

6 - coborîre scară;

7 - urcare scară;

8 - mers la dreapta;

0 - tras cu praștia;

S - se așază/se ridică (alternativ);

H - lovește;

J - sălt (sare).

Punctele și liniile la palmă:

- tras cu praștia în ANGELFACE: 10 p
- pumn într-un elev mai mic: 10 p
- pumn în ANGELFACE: 10 p
- numărul liniilor primite de ANGELFACE și EINSTEIN este de asemenea punctat: 1 linie = 1 p
- numărul liniilor primite de BOY WONDER sau de ERIC nu se socotește la puncte, dar dacă ERIC primește mai mult de 10000 de linii, jocul este pierdut.

Reguli de joc

Eric trebuie condus în camera indicată jos și acolo va avea ora cu profesorul.

Eric va primi linii dacă profesorul va observa că:

- nu merge la oră;
- dă pumni;
- sare;
- trage cu praștia;
- intră în cancelarie;
- stă în clasă în pauză;
- stă culcat;
- nu stă în bancă la oră.

Urcarea scărilor

Pentru urcarea scărilor, Eric trebuie deplasat în partea opusă față de locul unde este înclinată scara, apoi se va înainta pînă în dreptul scării și se va da comanda pentru urcare (tasta 7).

Coborîrea scărilor

Pentru coborîrea scărilor, Eric trebuie deplasat în partea unde este înclinată scara, apoi se va înainta pînă în dreptul scării și se va da comanda pentru coborîre (tasta 6).

Jucătorul poate opta pentru schimbarea numelor personajelor.

Flag - joc de îndemănare și reflexe

Jocul simulează o cursă de mașini "Formula 1" în care jucătorul va conduce prin diverse comenzi mașina aleasă, cu scopul de a termina cursa (în cazul unui accident, jocul se termină, putându-se relua) într-un timp cât mai scurt. În timpul conducerii mașinii, apar pe ecran diverse mesaje de avertizare și explicare care ajută jucătorul. De exemplu: "Atenție! Pete de ulei pe șosea"; "Petele de ulei pot provoca deraparea mașinii și accidente.

Utilizare

Jocul se încarcă cu comanda LOAD "" și se lansează în execuție automat. Jucătorul poate opta pentru o demonstrație sau își poate alege mașina și/sau circuitul convenabil.

Tastele folosite:

BREAK - alegerea circuitului și a mașinii;

D - opțiunea pentru demonstrație (DEMO);

ENTER (CR) - începerea jocului;

7 - ponire mașină și accelerare;

M - acționare manetă viteză pentru o viteză sporită;

N - acționare manetă viteză pentru o viteză mai mică;

A - viraj stînga;

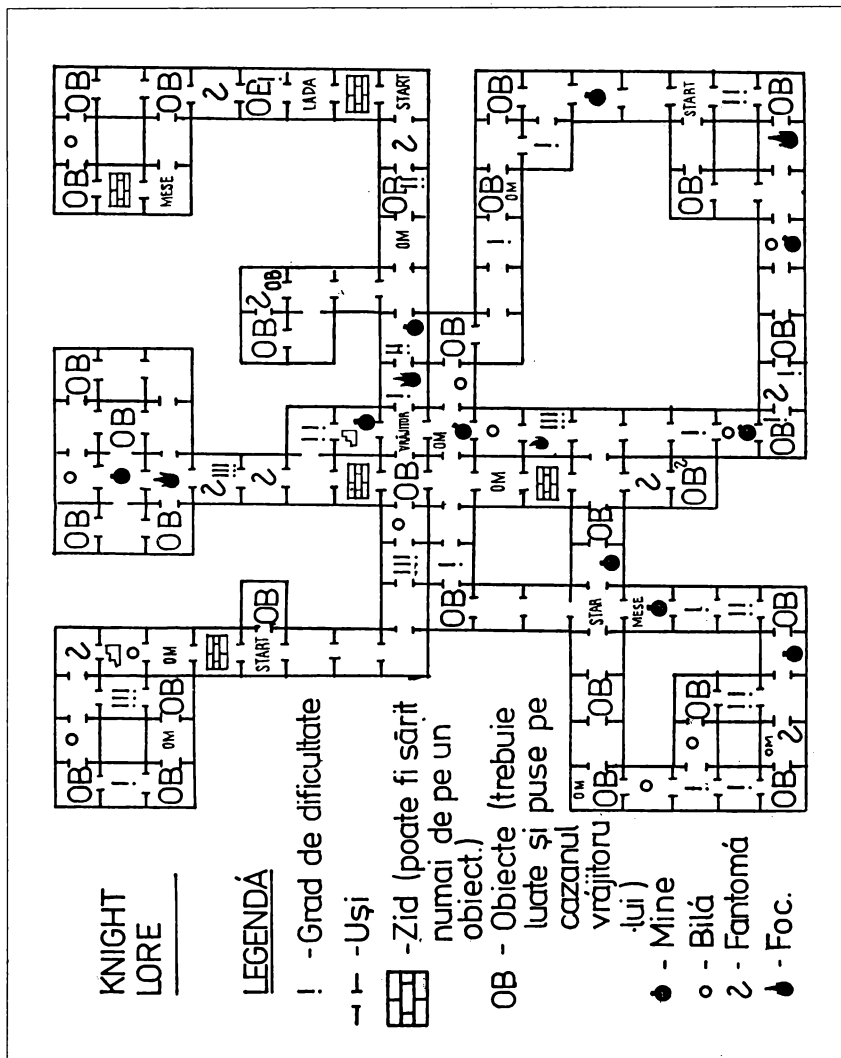
F - viraj dreapta;

H - încetinire și oprire mașină.

Jocul se poate termina forțat, iar dacă se parcurge tot circuitul se afișează timpul realizat, precum și recordul circuitului (timpul minim realizat).

Knight Lore - joc de aventuri

În jocurile de aventuri contează mai puțin reflexele, accentul punându-se pe abilitatea de a lua decizii, indiferent dacă jocul se referă la strategia conducerii unor expediții sau a unor întreprinderi sau dacă



este vorba pur și simplu de un joc în care personajul principal trebuie dirijat în sensul luării celor mai înțelepte decizii pentru a rezolva situații mai delicate.

Vom lua ca exemplu de joc de aventură, jocul Knight Lore, utilizabil pe calculatoarele Sinclair ZX Spectrum sau compatibile (HC, TIM, COBRA).

Jocul se încarcă cu comanda LOAD "", lansarea în execuție fiind automată. Jucătorul trebuie să conducă personajul principal, cavalerul Sabreman printr-un castel ciudat, ale cărui încăperi sînt dispuse ca într-un labirint (vezi desenul harta castelului). Tastele utilizate în acest scop sînt:

Q - pentru mers înainte ;

A - pentru sărit;

1 - luare obiect.

Rîndul de taste de jos, pentru rotire (perechi de 2 taste pentru care rotirea alternează dreapta/stînga).

Deseori jocurile de aventuri au la bază o poveste, un basm, iar jucătorul trebuie să se identifice cu personajul principal, avînd ca scop rezolvarea unor situații (eliberarea unui castel de fantome, găsirea unei comori, omorîrea unui balaur, aducerea la castel a Ilenei Cosînzene furată de un zmeu, etc.).

În jocul Knight Lore povestea este următoarea: asupra cavalerului Sabreman a fost aruncată o vrajă grea: ziua să aibă o înfățișare normală, iar noaptea să se transforme în lup. Sabreman trebuie să scape de această vrajă. În acest scop el este condus prin castelul labirint și trebuie să ajungă la vrăjitorul Melkhior (în centrul castelului), singurul capabil să îl scape de vrajă. Dar drumul este plin de capcane și neprevăzut. Există la tot pasul fantome, bile în mișcare, gărzi (soldați care păzesc camerele), mine care pot exploda, focuri, etc.). La orice atingere a acestora, Sabreman își va pierde o viață. Sabreman are 6 vieți.

La începutul jocului, Sabreman se află într-una din camerele castelului aleator în camerele notate în desen cu START). El trebuie condus pentru a ajunge la vrăjitorul Melkhior, care îi va cere un obiect (obiectul apare deasupra unui cazan pus la foc). Există în castel circa 21 de obiecte (ceașcă, diamant, cupă, bilă, gheată, sticlă, otravă, etc.), iar Melkhior va cere, pe rînd, 2/3 din aceste obiecte pentru a rupe vraja.

Dar atenție! lui Melkhior nu-i plac lupii, așa că, la venirea nopții, Sabreman trebuie să aibă grijă să fugă din camera vrăjitorului, căci altfel își va pierde o viață.

Obiectul cerut de vrăjitor va fi căutat de Sabreman prin castel, adus la vrăjitor și pus în cazan. Aceste situații se repetă de 14 ori, de fiecare dată vrăjitorul cerînd alt obiect dorit. Dacă se vor îndeplini sarcinile, jocul se va termina, Sabreman va fi salvat, iar jucătorul va primi felicitări.

În camerele castelului se pot găsi obiecte, dar și niște omuleți, care, pentru Sabreman reprezintă vieți suplimentare. În unele camere sînt obstacole care îl împiedică pe Sabreman să iasă. În acest scop vor fi folosite lăzi și mese (cîteodată și obiecte pe care le cere vrăjitorul) care sînt puse pentru a putea sări de pe ele peste obstacole.

În joc intevine și problema timpului, Sabreman avînd la dispoziție numai 40 de zile sau nopți pentru a rupe vraja. Jucătorul știe cînd este zi și noapte, cavalerul transformîndu-se în lup, în timpul nopții. Dacă timpul se termină și Sabreman nu reușește să găsească toate obiectele, jucătorul pierde.

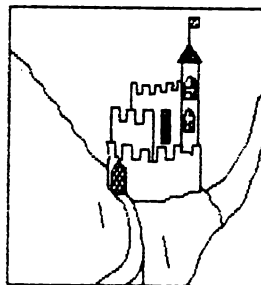
Este un fapt obișnuit ca jocurile de aventuri să dureze mult, rezolvarea unor situații dificile nefăcîndu-se cît ai bate din palme. Deci, nu dezarmați dacă nu veți reuși chiar după 2 ore de joc să rupeți vraja pentru Sabreman. Unele jocuri de aventuri, pentru a fi rezolvate au fost jucate de mai multe ori, luni în șir, pînă au fost dezlegate. Oricum,

pentru Knight Lore aveți la dispoziție și harta castelului (vezi figura) care a fost realizată de jucători care au terminat și studiat jocul. Dacă reușiți să opriți jocul și să faceți POKE 53567,0. Sabreman va avea un număr nelimitat de vieți, iar dacă veți face POKE 36212,0 nu se va mai pune problema timpului (veți putea juca oricât).

Făt Frumos - joc de aventuri

Este un joc realizat pe un scenariu inspirat din basmele populare românești în care apar personaje binecunoscute ca: Făt Frumos, Ileana Cosânzeana, Zmeul, Împăratul Roșu, Bătrînul Sihastru, etc. Bineînțeles Ileana Cosânzeana a fost furată de Zmeu iar Făt-Frumos pleacă în căutarea ei avînd de întîmpinat diferite piedici: capcane, balauri, zmei etc. Jocul se desfășoară sub formă de dialog în care jucătorul (care se identifică cu personajul principal - Făt Frumos) alege diverse opțiuni (decizii) prin care încearcă să o salveze pe Ileana Cosânzeana. Opțiunile pot fi de deplasare sau de luare a unor obiecte. După alegerea unei opțiuni va apare o nouă situație : un nou tablou (desen) și o listă de opțiuni. Practic acțiunea se desfășoară într-un labirint (asemănător cu cel din jocul Comoara din peșteră, dar cu mai multe încăperi), fiecărei încăperi corespunzîndu-i o anumită situație (desen și listă de opțiuni).

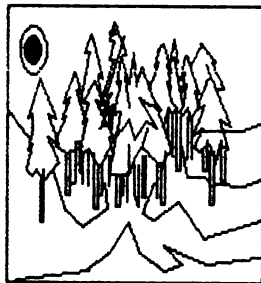
Deplasările pot fi la Nord, Sud, Est, Vest, Urcare sau Coborîre iar obiectele care se



pot lua pot fi: arc cu săgeți, paloș, Vifor (calul său) etc.

În fiecare situație se afișează și lista obiectelor pe care Făt Frumos le are asupra sa. Deoarece labirintul este destul de complicat recomandăm desenarea hărții jocului pentru rezolvarea lui (lăsăm jucătorilor plăcerea de a realiza harta jocului cu găsirea soluțiilor de ieșire din labirint).

În drumul pe care îl străbate Făt Frumos pot apărea situații care vor fi greu de depășit dacă cititorul nu cunoaște basmele culese de Petre Ispirescu, Ion Creangă etc. Se solicită de exemplu rezolvări de șarade sau răspunsuri la ghicitori. Deseori, deasemenea, jucătorul nu va putea rezolva o anumită situație dacă Făt Frumos nu are asupra sa un anumit obiect. De exemplu, nu va putea ieși din fântină dacă nu are la el frînghia pe care o găsește în Coliba Sărmanilor.



PROGRAME UTILITARE

PENTRU INTRETINEREA JOCURILOR PE CASETE MAGNETICE

Deseori pentru un posesor de calculator personal se pune problema întreținerii și evidenței programelor pe casete magnetice, mai cu seamă în momentul în care numărul programelor aflate în posesie depășește o anumită limită. Pentru a rezolva această problemă se folosesc, de obicei, programe special concepute. Deoarece ele servesc (ajută) utilizatorul de programe, se mai numesc programe utilitare.

Banda magnetică, pe care sînt înregistrate jocuri pentru calculator este supusă unor mișcări mecanice. Datorită mecanismelor casetofonului pot apărea "șifonări" sau ruperi de bandă ceea ce va avea ca urmare deteriorarea jocului și, de obicei, imposibilitatea facerii lui. De asemenea, nici înregistrările magnetice nu au o viață infinită. După circa 2-3 ani pot apărea demagnetizări, mai cu seamă în condițiile în care caseta respectivă a fost păstrată sau ținută în locuri neindicate (de exemplu în apropierea surselor de tensiune și aflate sub tensiune). Iar rezultatul va fi același: pierderea înregistrării; pentru că, spre deosebire de o înregistrare muzicală, a cărei calitate poate scădea (dar fără o pierdere totală a înregistrării), orice deteriorare a înregistrării, în cazul unui program, oricît de mică, va avea ca urmare pierderea sa definitivă.

Pentru a evita pierderile de această natură se procedează la copierea programelor pe casete magnetice. În cazul jocurilor, însă, multe din acestea sînt protejate la copiere, astfel încît nu va fi posibilă

o manevră simplă, de întrerupere a programelor în execuție și apoi o salvare a lor, cu comanda SAVE, ca în cazul programelor scrise în limbaj BASIC. Pentru a depăși această dificultate se va recurge la un program utilitar de copiere, care va asigura copierea programelor.

Menționăm că acest procedeu nu asigură o rețetă definitivă, deoarece unele jocuri sînt protejate față de copierea lor cu anumite programe de copiere (o modalitate de protejare, este, de exemplu, salvarea mai rapidă a programelor, asigurîndu-se o densitate a înregistrării pe bandă mai mare, prin procedeul QUICK SAVE). Cu toate acestea, majoritatea jocurilor nu "rezistă" în fața programelor de copiere. Modalitatea de lucru cu un program utilitar de copiere este următoarea: se încarcă programul de pe casetă magentică (comanda LOAD ""), apoi se vor da comenzile necesare încărcării programelor care se doresc a se salva, copierile asigurînd acest lucru în mod interactiv. Comenzile sînt, de obicei, afișate pe ecran sub forma unei liste de opțiuni. În sfîrșit, după încărcare (care nu va trebui să depășească capacitatea de memorie disponibilă, de obicei afișată pe ecran) se va trece la salvarea programelor cu (eventual) verificarea înregistrării realizate, dacă programul utilitar o permite. După ștergerea programelor astfel salvate din memoria calculatorului, procedeul se poate repeta și cu alte programe. Menționăm că, de obicei programele-joc sînt formate din mai multe înregistrări (blocuri sau fișiere). Acestea pot fi de tip BASIC (notate deseori cu P), de tip cod (în cod mașină, notate deseori cu C sau Bytes), de tip antet, de tip ecran (salvate în prealabil cu SAVE SCREEN), de tip date, etc.

Pentru a realiza salvarea unui program de acest tip se poate recurge la salvarea lui pe bucăți (fișiere), mai ales în cazul în care există în structura sa un fișier foarte lung, care ocupă, de obicei, aproape toată memoria disponibilă (memoria rămasă la dispoziție după încărcarea programului de copiere, care ocupă și el o parte).

Modalitatea de lucru constă în încărcarea pe rînd a unui fișier (sau a mai multor fișiere mai mici) și apoi salvarea lor pe caseta magnetică, bineînțeles, cu păstrarea aceleiași ordini a fișierelor. Cîteodată, se poate întîmpla ca fișierului cel mai lung din cadrul programului să nu îi fie suficientă memoria disponibilă; în acest caz, se recurge la opțiunea pentru copierea fișierelor foarte lungi (dacă programul utilitar permite această opțiune) - MAX BYTE.

În cazul în care înregistrările (chiar repetate) vor conține erori, se va proceda la analizarea acestor înregistrări cu alt program utilitar - pentru testarea și diagnosticarea înregistrărilor pe casete magnetice. Se poate apela la programul DIAG (vezi revista Tehnium nr.4 1988).

Prezentăm în continuare cîteva din cele mai cunoscute și mai folosite programe utilitare de copiere pentru calculatoare Sinclair Spectrum și compatibile (HC 85, TIM-S, COBRA).

COPIER FM3

este un copier foarte răspîndit, fiind și printre primele copiere performante apărute. El permite copierea fișierelor standard și a blocurilor fără antet, detectează erorile de încărcare, afișează informațiile din antet pentru fișierele standard și lungimea pentru blocurile fără antet. După instalarea sa, copierul va încărcă în memorie primul fișier pe care îl va întîlni. De remarcat că FM3 permite încărcarea mai multor fișiere odată, practic el asigurînd încărcarea atît cît va permite memoria sau pînă cînd utilizatorul va comanda trecerea de la opțiunea de încărcare la cea de salvare a fișierelor citite. Numele fișierelor și informațiile despre ele vor rămîne afișate pe ecran. Opțiunea de salvare se va selecta prin acționarea tastei SPACE și în acest caz va apărea o listă de opțiuni prin care se poate asigura copierea fișierelor unul cîte unul (opțiunea Copy - tasa C); copierea fișierelor, toate odată (opțiunea Auto - tasta A) sau copierea selectivă

ZOTYOCOPY +

a unor fișiere prin trecerea peste alte fișiere a căror copiere nu se dorește (opțiunea Skip - tasta S).

Afișarea informațiilor din antetele fișierelor se face în baza 16 (în acest caz pe ecran va apărea afișat RADIX: HEX) sau în baza 10 (RADIX: DEC), schimbarea bazei de afișare făcându-se prin acționarea rastelor SYMBOL SHIFT (SS) și 4.

După fiecare operație, FM3 poate emite un semnal sonor, care se poate inhiba/activa prin acționarea tastelor CAPS SHIFT (CS) și 1.

Cînd se detectează o eroare la citirea unui fișier, FM3 va marca blocul de date incorect cu semnul * în ultima coloană.

La încheierea lucrului cu FM3, acesta poate fi șters din memorie prin acționarea tastelor CAPS SHIFT (CS) și E (opțiunea END).

ZOTYOCOPY +

Este un copier foarte răspîndit, în special datorită modului ușor de manipulare. Opțiunile apar în permanență afișate în partea dreaptă a ecranului (lăsînd un spațiu mare pentru afișarea informațiilor din antet), la un moment dat fiind disponibile cele care sînt afișate cu contrast.

ZOTYOCOPY + permite copierea fișierelor standard și a blocurilor fără antet; opțiunile fiind accesate prin acționarea tastei laterale corespunzătoare opțiunii afișate pe ecran (prima literă).

Opțiunile sînt:

Load (tasta L) - încărcarea fișierelor în memorie;

Forget (tasta F) - ștergerea fișierelor din memorie, dacă apoi se acționează ENTER (CR);

Save (tasta S) - salvarea fișierului pe care se află cursorul (dacă în prealabil a fost accesată opțiunea Hand) sau a tuturor fișierelor afișate (memorate) (dacă în prealabil a fost accesată opțiunea Auto);

Verify (tasta V) - verificarea unui fișier sau a tuturor fișierelor salvate (dacă în prealabil a fost accesată opțiunea Hand sau respectiv Auto).

Detectarea unei erori la citire va fi marcată chiar în momentul respectiv cu afișarea mesajului "Tape Error". În acest caz sistemul se va bloca. Deblocarea se va face prin acționarea tastei BREAK și apoi L (Load) prin care se va reintra în opțiunea de încărcare cu pierderea fișierului cu eroare;

Auto (tasta A) - salvarea tuturor fișierelor. Este activă după acționarea opțiunii Save;

Hand (tasta H) - salvarea a câte unui fișier, prin poziționarea ulterioară a cursorului cu opțiunea Up sau Down. Este activă după acționarea opțiunii Save;

Up (tasta U) - cursorul în sus (după opțiunea Hand);

Down (tasta D) - cursorul în jos (după opțiunea Hand);

ENTER (CR) - confirmarea opțiunii și trecerea la executarea comenzii după opțiunile Forget, Auto, Hand, Up, Down;

BREAK - oprire operație/activarea unui alt set de opțiuni.

Copierul afișează pe ecran informații din antetul fișierelor: tipul fișierului (P = BASIC, C = cod sau ecran), numele, lungimea în octeți și adresa de încărcare.

După acționarea tastei BREAK, dacă s-a terminat o sesiune de lucru, prin acționarea tastelor CS și E se va realiza opțiunea End prin care se va părăsi copierul și se va șterge din memorie.

Dacă la citirea unui fișier se va depăși capacitatea de memorie disponibilă (41780 octeți inițial), acest lucru se va indica prin afișarea mesajului End of Ram.

De remarcat că ZOTYCOPY + permite și copierea unor fișiere mai lungi de 41780 de octeți prin opțiunea MAX BYTE. Operația se va realiza astfel:

- se golește toată memoria disponibilă (cu Forget);
- se acționează tasta BREAK;
- se acționează împreună tastele CS și M (opțiunea MAX BYTE);

ULTRACOPY

- se dă drumul la casetofon, în prealabil banda fiind potrivită exact la începutul fișierului lung;

- la terminarea încărcării, se va insera în casetofon caseta pe care se dorește salvarea și se va porni casetofonul pentru înregistrare; în acest moment, la acționarea oricărei taste, calculatorul va transmite fișierul către casetofon. La terminarea înregistrării, copierul se va autodistrage, fiind necesară o reîncărcare a sa pentru a se continua operația de copiere.

ULTRACOPY

(realizat la cercul de la Casa Științei și Tehnicii Tîrgu-Mureș, autor Adrian Pop) are avantajul opțiunilor, mesajelor și comenzilor în limba română.

Opțiunile se accesează tot prin acționarea tastei corespunzătoare primei litere a opțiunii (această literă este în contrast), iar posibilitatea accesării unei opțiuni se indică foarte ingenios prin cursor și săgeți în contrast. O opțiune activă este indicată printr-un cursor care clipește. Modul de lucru este asemănător cu cel de la ZOTYOCOPY +. Opțiunea MAXBYTE este afișată pe ecran. Opțiunile sînt: citesc, salvez, verific, anulez, Maxim, Termin. Acestea sînt afișate pe coloana din partea stîngă. Opțiunile derivate din cele anterioare: auto, man(ual), cursor, sus (se obține cu CS și 7), cursor jos (se obține cu CS și 6), precum și ENTER (CR) și BREAK sînt afișate pe coloana din partea dreaptă.

KOPY-KAT

- copiază doar fișiere standard Spectrum, încărcînd în memorie fișierele unul cîte unul, salvîndu-le apoi pe bandă.

La încărcare afișează tipul și numele fișierului. În cazul unei erori apare mesajul "R Tape Loading error" sau "B Break - continue", execuția reluîndu-se cu RUN. Ca urmare, programul încărcat se pierde. Copierul ghidează utilizatorul prin mesajele START

TAPE/LOAD TAPE ce apar pe ecran. KOPYKAT se poate autoduplica cu comanda: MERGE "KOPY-KAT": RUN 100.

COPY-TAPE (ICE)

este o versiune de KOPY-KAT care lasă utilizatorului un buffer mai mare pentru încărcarea fișierului de copiat. Lansarea se face cu LOAD "COPY-TAPE" CODE: RAND USR 23300.

COPY-TAPE se încarcă în bufferul de imprimantă de la adresa 23296 și are o lungime de 256 de octeți.

THE KEY

permite copierea mai multor fișiere standard odată, operarea făcându-se conform indicațiilor de pe ultima linie a ecranului. Salvarea se realizează o singură dată. Nu poate copia blocuri de date fără antet. Este important deoarece unele programe salvate cu THE KEY nu mai pot fi copiate decât cu același copier.

COPY-COPY

este un utilitar de copiere important deoarece are încorporate o serie de opțiuni (ele sînt instrucțiuni sau comenzi BASIC, acționîndu-se ca în BASIC), cu ajutorul cărora se pot insera în program noi linii de instrucțiuni sau comenzi, realizîndu-se în acest mod modificări (cu POKE) ale unor reguli de joc în programele jocuri.

Iată lista opțiunilor și modul în care se pot accesa:

Opțiune	Accesare (tastă/ taste)	Efect
LOAD	J	Încarcă programele (fișierele) înfîlnite pe bandă. Dacă după se va introduce un număr, de exemplu 2, atunci se va încărca al doilea program înfîlnit
LIST n	K	Listează valorile din locațiile. Se vor afișa cite 15 valori pe fiecare pagină. Cu ENTER se vor afișa următoarele 15 valori.
BREAK	BREAK	Înterupere încărcare/salvare program
LET n	L	Modificarea informațiilor referitoare la programul încrcat. Se utilizează astfel: Se face BREAK după încărcarea unui program, apoi LET n = noul nume, adresa de start, lungimea. n este numărul programului; se poate modifica astfel numele programului, adresa de start și lungimea.
DELETE	CS și 0	Sterge comanda introdusă. Comanda introdusă se afișează în partea de sus a ecranului.
SAVE	S	Salvează pe casetă programele încărcate în memorie (AUTO). Se poate utiliza și forma SAVE n1 TO n2 STEP n3, unde TO se obține ca în BASIC, SS și F. n1 și n2 reprezintă numere de ordine ale programelor încărcate (deci se vor salva programele de la n1 și n2), iar n3 va reprezenta pauza în secunde dintre programe.
VERIFY n	V	Verificare program.
POKE addr,x	O	Modificarea conținutului adresei de memorie addr cu valoarea x.
RAND USR n	U	Executarea unei rutine de la adresa n.
RETURN in BASIC	Y	Întoarcere în BASIC.
FORGET	CS și 7	Sterge ultimul program din listă.
PRINT	P	Tipărire la imprimantă a numelor programelor înfîlnite
COPY	Z	Salvare programe lungi
CATALOG	C	Afișează numele programelor citite fără să le încarce și fără să le distrugă.

Mecanismul de folosire a unor programe mai recente (Copy86, TFcopy etc) este asemănător cu al celor descrise. Ele asigură copierea unor fișiere mai lungi (asigurînd o memorie disponibilă care ajunge cîteodată la 52 Ko) folosind în acest scop memoria ecranului și/sau compactarea fișierelor la încărcare cu decompactarea lor la salvare.

CUPRINS

CUVÎNT ÎNAINTE	3
JOCURI DE ÎNDEMÎNARE.....	5
Cursa cu obstacole.	5
Labirint.	9
Firul Ariadnei.	15
Robac.	17
JOCURI LOGICE	26
Cuvinte încrucișate.	26
Dame (variantă).	28
Vânătoarea engleză.	32
Reversi.	41
JOCURI DE AVENTURI.....	50
Pierdut în junglă.	50
Comoara din peșteră.	57
JOCURI PENTRU ȘCOALĂ.....	64
Simultan.	64
Animale.	68
JOCURI PE CASETE.....	76
BRUCE LEE.....	76
KUNG FU.....	77
Robin.	79
SKOOL DAZE.....	81
FLAG.....	83
Knight Lore.....	83
Făt Frumos.....	87
PROGRAME UTILITARE.....	89
Copier FM3.....	91
ZOTYOCOPY +.....	91
ULTRACOPY.....	94
THE KEY.....	95
COPY-COPY.....	95

