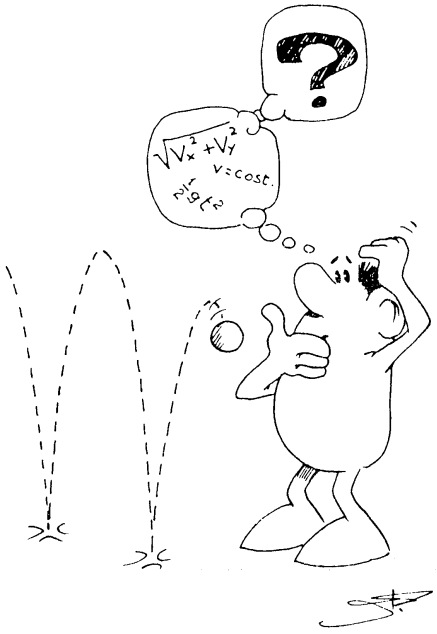
di Sergio Borsani

Palla che rimbalza



Un giorno, a casa di un amico, ho potuto vedere alcuni giochi di provenienza USA girare su un TI 99/4A equipaggiato con armamento pesante, cioè espansione, floppy, editor/Assembler, speech synthetizer, ecc. In quell'occasione rimasi colpito da un programma che simulava il gioco del tennis. Una voce da Odissea nello Spazio scandiva i punteggi, "fifteen - thirty", "fifteen - forty", e la pallina rimbalzava in modo molto realistico. Potere degli sprite. Sono stati i protagonisti anche all'ultimo corso di BASIC da me organizzato. Non c'è niente che attragga l'attenzione e stimoli la fantasia di un ragazzo, più di uno spiritello che aleggia sullo schermo. Qualcuno è tentato di creare un gioco lì per lì, come se si trattasse di oggetti reali da manipolare con la tastiera invece che con le mani; ma le difficoltà non tardano a venire ed è bene non farsi troppe illusioni.

Il computer o il video che è il mezzo elettivo con il quale ci comunica, ci presentano un microcosmo bidimensionale dalle leggi fisiche alterate; non c'è gravità nè attrito, i corpi, quando si urtano, invece di rimbalzare elasticamente si compenetrano senza deformarsi proseguendo nel loro moto inerziale. Ma che strano moto inerziale. Invece di allontanarsi indefinitamente dal punto di partenza, una volta varcato il bordo dello schermo, ricompaiono esattamente dal lato opposto. Districarsi tra i Print, gli Input e gli If-Then Else, poteva essere un'impresa ardua ma non impossibile, ma qui, per tutti i diavoli, per far saltare una pallina bisogna conoscere la geometria e la fisica, e ricreare a suon di formule il mondo che ci è familiare.

Il libro della Natura è scritto in lingua matematica, diceva Galileo Galei, che non a caso viene qui citato per essere stato il precursore nell'enunciazione delle leggi che regolano la caduta dei gravi e che descrivono le traiettorie dei corpi che si muovono sotto l'azione delle forza di gravità.

"Proiectum, dum fertur motu composito ex horizontali aequabili et ex naturaliter accelerato deorsum, ecc." (Discorsi, IV).

Gli sprite, nel TI 99/4A, hanno la caratteristica, non comune a tutti gli altri computer, di mantenere la loro velocità automaticamente grazie ai parametri presenti nell'istruzione Call Sprite e nella Call Motion. Ad esempio, nel Commodore 64 il movimento avviene modificando continuamente i registri che controllano la posizione.

Una pallina che rimbalza si può immaginare animata contemporaneamente da due movimenti, uno uniforme lungo la direzione orizzontale ed uno uniformemente accelerato lungo la direzione verticale. Nel primo la velocità è costante, la qual cosa si ottiene facilmente nel TI 99/4A definendo una volta per tutte l'intensità della componente orizzontale della velocità. Nel listato 1 è stato posto VX=20, un valore che sposta lo sprite verso destra consentendo qualche rimbalzo prima che la pallina attraversi tutto lo schermo. La componente verticale deve variare secondo la legge Vy=Voy−gt; se inizialmente la pallina viene lanciata verso l'alto, la sua velocità diminuisce di una quantità proporzionale al tempo. Quando, aumentando t, il prodotto gt diventa uguale a Voy, la velocità Vy risulta uguale a zero, la pallina pertanto smette di salire avendo raggiunto il punto più alto della sua traiettoria e poi comincia a scendere. Il nostro computer non possiede la funzione Time ed il passare del tempo dev'essere scandito incrementando una variabile: T=T+1. In fisica se t è espresso in

secondi e V in m/s, la costante g assume il valore 9,7805 o, come riportano la maggior parte dei testi, 9,8. Noi, poiché aumentando T nel modo descritto in precedenza, non abbiamo un tempo espresso in secondi, non potremo usare questo valore di g. Nel programma 1, alla linea 150, abbiamo posto g=2. Perché abbiamo scritto VY+2★T invece di VY−2★T come suggerisce un confronto con la formula data in precedenza?

La ragione sta nel sistema di riferimento adottato. Normalmente l'asse verticale è orientato verso l'alto mentre il computer (non tutti) pone l'origine nell'angolo in alto a sinistra del video e pertanto orienta l'asse verticale verso il basso. A questo si rimedia attribuendo a VY valori negativi e a g valori positivi. Basta un meno o un più per farci andare a gambe all'aria.

Eseguendo il programma vedrete la pallina rimbalzare da sinistra verso destra seguita dalla fedele ombra più in basso. A proposito, come si fa in modo che la pallina non scenda al di sotto della sua ombra? Se controlliamo la posizione dovremo farlo quando la pallina è ancora al di sopra dell'ombra, ma in tal caso il computer non distingue se lo sprite sta scendendo o se sta salendo. Conviene controllare la velocità. Se è negativa la pallina sta ancora salendo, se è positiva, allora sta scendendo. E quando sappiamo che la pallina è prossima al rimbalzo? Quando la componente verticale della velocità è vicina al valore iniziale VY con il segno cambiato; la pallina cioè cade con la stessa velocità con la quale era salita. Poiché inizialmente era VY=−50, se V diventa maggiore di 45 significa proprio che siamo quasi arrivati, poniamo lo sprite al centro dell'ombra con una provvidenziale Call Locate, invertiamo il segno a V, azzeriamo l'orologio e via per un altro rimbalzo.

Qualcuno osserverà giustamente che la pallina prima o dopo dovrà fermarsi; gli urti non sono perfettamente elastici, parte dell'energia viene assorbita e a ogni rimbalzo viene raggiunta una minore altezza mentre la traiettoria si fa meno ampia. La modifica del programma risulta abbastanza agevole, basta inserire un ciclo For-Next che faccia diminuire ad ogni rimbalzo la componente verticale della velocità. Nel programma Palla 2, VY viene fatta variare da −60 a −5 ottenendo così l'effetto desiderato senza ulteriori sostanziali cambiamenti (listato 2).

Che dire poi della palla da tennis dalla quale ha preso inizio il nostro discorso? Una pallina da tennis rimbalza continuamente da sinistra a destra e da destra a sinistra. Le modifiche rispetto al primo programma sono ancora più semplici ed invece di tante spiegazioni proponiamo una specie di quiz. Se qualcuno non trova una soluzione

Listato 1 - Il programma Palla 1.

```
100 REM  PALLA 1
110 CALL CLEAR :: CALL CHAR(128,"071F3F7
F7F"&RPT$("F",12)&"7F7F3F1F07E0F8FCFEFE"
&RPT$("F",12)&"FEFEFCF8E0")
120 CALL CHAR(132,RPT$("0",16)&"0F3F7FFF
7F3F0F"&RPT$("0",18)&"F0FCFEFFFEFCF000")
130 VX=20 :: VY=-50 :: T=1
140 CALL SPRITE(#1,128,12,150,10,VY,VX,#
2,132,2,158,10,0,VX):: CALL MAGNIFY(3)
150 T=T+1 :: V=VY+2*T :: IF V<45 THEN CA
LL MOTION(#1,V,VX):: GOTO 150
160 V=-50 :: T=1 :: CALL POSITION(#2,Y,X
):: CALL LOCATE(#1,150,X):: GOTO 150
```

Listato 2 - Il programma Palla 2.

```
100 REM  PALLA 2
110 CALL CLEAR :: CALL CHAR(128,"071F3F7
F7F"&RPT$("F",12)&"7F7F3F1F07E0F8FCFEFE"
&RPT$("F",12)&"FEFEFCF8E0")
120 CALL CHAR(132,RPT$("0",16)&"0F3F7FFF
7F3F0F"&RPT$("0",18)&"F0FCFEFFFEFCF000")
130 VX=10 :: VY=-60 :: T=1
140 CALL SPRITE(#1,128,12,150,10,VY,VX,#
2,132,2,158,10,0,VX):: CALL MAGNIFY(3)
150 FOR VY=-60 TO -5 STEP 5
160 T=T+1 :: V=VY+2*T :: IF V<ABS(VY+4)T
HEN CALL MOTION(#1,V,VX):: GOTO 160
170 V=VY :: T=1 :: CALL POSITION(#2,Y,X)
:: CALL LOCATE(#1,150,X)
180 NEXT VY :: T=1 :: GOTO 140
```

Listato 3 - Il programma Palla 3.

```
100 REM  PALLA 3
110 CALL CLEAR :: CALL CHAR(128,"071F3F7
F7F"&RPT$("F",12)&"7F7F3F1F07E0F8FCFEFE"
&RPT$("F",12)&"FEFEFCF8E0")
120 CALL CHAR(132,RPT$("0",16)&"0F3F7FFF
7F3F0F"&RPT$("0",18)&"F0FCFEFFFEFCF000")
130 VX=30 :: VY=-50 :: T=1
140 CALL SPRITE(#1,128,12,150,10,VY,VX,#
2,132,2,158,10,0,VX):: CALL MAGNIFY(3)
150 T=T+1 :: V=VY+2*T :: IF V<45 THEN 17
0
160 V=-50 :: T=1 :: CALL POSITION(#2,Y,X
):: CALL LOCATE(#1,150,X):: VX=-VX :: CA
LL MOTION(#2,0,VX)
170 CALL MOTION(#1,V,VX):: GOTO 150
```

Listato 4 - Il programma Palla 4.

```
100 REM  PALLA 4
110 CALL CLEAR :: CALL CHAR(128,"071F3F7
F7F"&RPT$("F",12)&"7F7F3F1F07E0F8FCFEFE"
&RPT$("F",12)&"FEFEFCF8E0")
120 CALL CHAR(132,RPT$("0",16)&"0F3F7FFF
7F3F0F"&RPT$("0",18)&"F0FCFEFFFEFCF000")
130 VX=60 :: VY=-50 :: T=1
140 CALL SPRITE(#1,128,12,150,10,VY,VX,#
2,132,2,158,10,0,VX):: CALL MAGNIFY(3)
150 T=T+1 :: V=VY+4*T :: IF V<45 THEN 17
0
160 V=-50 :: T=1 :: CALL POSITION(#2,Y,X
):: CALL LOCATE(#1,150,X):: VX=-VX :: CA
LL MOTION(#2,0,VX)
170 CALL MOTION(#1,V,VX):: GOTO 150
```

soddisfacente può esaminare il listato 3, ma è invitato a farlo non prima di aver eseguito qualche tentativo.

Vogliamo infine aggiustare il tiro. Dopo tutto la risposta contenuta nel programma 3 sembra piuttosto un "pallonetto". Sapreste ottenere un colpo teso e all'apparenza dotato di una maggiore energia? Un consiglio: aumentare VX e simulare un campo gravitazionale più intenso. Non arrendetevi subito, una possibile soluzione, è contenuta nel listato 4. ■

di Paolo Agostini

File Examiner

Molto spesso succede di incontrare difficoltà nell'esaminare programmi in BASIC o in linguaggio macchina, a causa del fatto che hanno qualche sistema di protezione oppure perché nel corso del caricamento, vengono posizionati in locazioni di memoria difficilmente accessibili (Per esempio RAM posta sotto la memoria ROM), o altro.

Per ovviare a tale inconveniente, abbiamo progettato il programma Filex per il C 64 che ha le seguenti funzioni:

1) legge la directory del disco senza disturbare la memoria del computer;

2) legge da disco e visualizza i programmi in BASIC convertendone i "token" in parole-chiave;

3) legge da disco e disassembla i programmi in linguaggio macchina, visualizzandoli sullo schermo sotto forma di codice in linguaggio Assembly;

4) legge da disco e visualizza file numerici o di altro tipo, quali ad esempio i file di un word processor.

Il programma, proprio a causa della diversificazione dei compiti che si propone, è abbastanza lungo, e necessita della massima attenzione nella trascrizione dei dati relativi ai codici operativi. Noterete infatti la presenza, in coda ad ogni codice, di quella che viene chiamata dagli addetti ai lavori "opcode tag", cioè di una lettera che segue le tre lettere del codice operativo e che serve a specificarne il "modo":

1) codice di 3 lettere seguito da uno spazio

Listato 1 - Il programma Filex.

```
5 POKE53280,4:POKE53281,7:PRINTCHR$(147);
  CHR$(31);"ATTENDERE PREGO"
7 FORX=1TO18:X$=X$+CHR$(32):NEXTX:GOSUB10
  00
9 REM ------------------------------
10 REM ***  MENU PRINCIPALE ***
11 REM ------------------------------
20 PRINTCHR$(147);TAB(7);CHR$(176);:FORI=
   1TO23:PRINTCHR$(192);:NEXT
30 PRINTCHR$(174)
40 PRINTTAB(7);CHR$(221);SPC(5);"FILE EXA
   MINER";SPC(5);CHR$(221)
50 PRINTTAB(7);CHR$(221);SPC(5);"DI P.AGO
   STINI";SPC(5);CHR$(221)
60 PRINTTAB(7);CHR$(173);:FORI=1TO23:PRIN
   TCHR$(192);:NEXT
70 PRINTCHR$(189)
80 PRINT:PRINT:PRINTTAB(17);CHR$(18);"MEN
   U";CHR$(146)
90 PRINT:PRINT:PRINT
100 PRINTTAB(6);CHR$(18);" (1) ";CHR$(146
    );SPC(2);"LEGGI LA DIRECTORY"
110 PRINT:PRINTTAB(6);CHR$(18);" (2) ";CH
    R$(146);SPC(2);"LEGGI IL FILE"
130 PRINT:PRINTTAB(6);CHR$(18);" (3) ";CH
    R$(146);SPC(2);"FINE"
140 PRINT:PRINT:PRINTTAB(16);CHR$(18);"QU
    ALE?";CHR$(146)
150 GETA$:A=VAL(A$):IFA<1ORA>3THEN150
160 ONAGOTO 200,500,3200
170 END
179 REM ------------------------------
180 REM CHIUSURA CANALI APERTI
181 REM ------------------------------
190 PRINT#15,"I0":CLOSE1:CLOSE15:GOTO3100
199 REM ------------------------------
200 REM **** LEGGE LA DIRECTORY ****
201 REM ------------------------------
210 PRINTCHR$(147):PRINT"DISK NAME = ";
220 CLOSE1:CLOSE15:FL=0:L$="":B$=""
230 OPEN1,8,0,"$":OPEN15,8,15
240 INPUT#15,E1,E1$,E2,E3:IF E1 THENPRINT
     E1,E1$:GOTO190
250 FORI=1TO33:GET#1,A$
260 L$=L$+A$:NEXT:CLOSE1:PRINTL$
270 FORI=1TO40:PRINTCHR$(45);:NEXTI
280 PRINT"{RVS}BLOCKS{ 3 SPAZI}FILE NAME
    { 6 SPAZI}TIPO "
290 OPEN1,8,0,"$"
300 GET#1,A$:GET#1,A$:GOSUB350
310 GOSUB350:IF FL=1 THENPRINT:GOTO190
320 PRINTRIGHT$(X$+STR$(L),4);CHR$(32);CH
    R$(32);CHR$(32);
330 PRINTLEFT$(B$+X$,18);FT$:GETA1$:IFA1$
    <>""THEN3070
340 GOTO310
349 REM ------------------------------
350 REM SUBROUTINE LETTURA DIRECTORY
351 REM ------------------------------
360 B$="":GET#1,A$:GET#1,A$
370 GET#1,A$:L=ASC(A$+CHR$(0))
380 GET#1,A$:L=L+ASC(A$+CHR$(0))*256
390 GET#1,A$:INPUT#15,E1,E1$,E2,E3:IF E1
    THENPRINT E1,E1$:GOTO190
400 IFA$=""THENFL=1:RETURN
410 IFA$<>CHR$(34)THEN390
420 GET#1,A$
425 IFPEEK(653)THEN425
430 IFA$=CHR$(34)THEN450
440 B$=B$+A$:GOTO420
450 GET#1,A$:IFA$=CHR$(32)THEN450
460 FT$=A$
```