



■ Secondo appuntamento col software Amiga dei lettori. Questo mese presenteremo un interessante gioco di navicelle, nemico e... mattonelle fatto interamente in AmigaBasic. Il divertimento è proprio questo: nonostante una velocità operativa non troppo elevata, il solo pensiero che sotto vi sia soltanto il Basic fa quasi accapponare la pelle. ■

F15

di Dante Sbrega - Roma

Il programma listato in queste pagine è il nucleo principale del gioco. Chi acquisterà il dischetto presso la redazione avrà in più una schermata iniziale con tanto di F15 in volo ed una alquanto stonata musicchetta ispirata a Mozart (speriamo non se la sia presa a male... Mozart).

Il gioco è assai semplice: ci troviamo in volo con il nostro F15 su di un paesaggio di piramidi mozzate. Il no-

stro scopo è quello di colpire tutte le piramidi che ci passano sotto. Ovviamente le piramidi mancate potremo colpirle al giro successivo dato che lo sfondo si ripete ciclicamente, come se stessi volando attorno ad un gigantesco cilindro.

Semplice, no?... NO. C'è un piccolo problema: mentre compiamo la nostra missione la sfera viola del nemico, come lo stesso autore l'ha battezzata, si diverte ad andare in giro per lo schermo rimbalzando a mo' di pallina quando sbatte contro i bordi. Se nella sua traiettoria ci intercetta, ci distrugge facendoci quindi perdere una navicella. Il caso è tutt'altro che fortuito, anzi per completare uno schermo bisogna praticamente "multiplexare" il no-

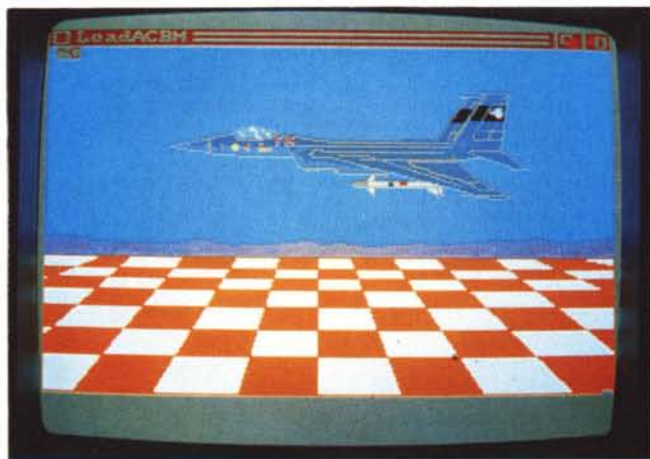
stro apparato visivo, seguendo con un occhio il nemico e con l'altro le piramidi da abbattere.

Si parte con un credito di 5 navicelle, e ogni quadro che completiamo otteniamo un bonus di 3 navicelle. Perse tutte le navicelle una allegra musicchetta ci avvisa che siamo deceduti e il relativo punteggio appare sullo schermo.

Vuoi giocare ancora (Y/N)?

Commento al programma

Come già anticipato, in apertura, il gioco non è eccessivamente divertente, ma il solo fatto di essere realizzato interamente in Basic fa aggiudicare a pieni voti la palma d'oro all'autore.



Schermata grafica di apertura.



Istruzioni

F15

```

'F15 1987 by Dante Sbraga
DEFINT a-z:COLOR 1,0
DEF FNsize(x1,y1,x2,y2,d)=3*(y2-y1+1)*INT((x2-x1+16)/16)*d
lm=34:hm=31 'larghezza altezza mattonella
fx=305:fy=185 'maxx maxy finestra
nx=15:ny=23 'maxx maxy pavl.s
mx=15:my=15 'maxx maxy mirl.s
sy=2 'scorrimento
no=6 'num object
delta=60 'distanza mir nav
DIM SHARED mat(FNsize(1,1,lm,hm,3)),temp(FNsize(0,0,fx,sy-1,3))
DIM SHARED mus(3,3) 'musica fondo
DIM SHARED colpre(no) 'collisione precedente
SCREEN 1,320,200,3,1
'finestra 306x186
WINDOW 2,"# LEVEL OK ##F15## B7 by Dante Sbraga",(3,0)-(3+fx,fy),0,1
CALL initcol:CALL copyright:CALL initmus:CALL initobj
PRINT "READY STAGE 0":PRINT
COLOR 7:PRINT "Premere RETURN per iniziare!":INPUT "",t$
CALL initmat:obiet=4:numnav=5:score=0:ultcol=-3
'main
WHILE -1
CALL initcol:CLS:FOR j=0 TO 3:SOUND mus(1,j),1,255:NEXT
CALL init:suono=1:lancio=0:distru=0:puntnot=0:puntmot=0:contmot=0
COLOR 6,0:'level
WHILE distru<54 AND numnav<0
IF STRIG(3) THEN CALL attac
ON COLLISION GOSUB coll
IF colpre(6)<0 THEN CALL defen
CALL update1((STICK(2)),(STICK(3)))
IF STRIG(3) THEN CALL attac
IF lancio THEN CALL expnem
IF colpre(2)=6 THEN CALL expnav
IF STRIG(3) THEN CALL attac
CALL music
GET(0,fy-sy+1)-(fx,fy),temp
IF lancio THEN
ynew=ynew+sy:OBJECT.Y obiet,ynew
END IF
SCROLL(0,0)-(fx,fy),0,sy
PUT(0,0),temp
WEND 'level
OBJECT.OFF:IF numnav=0 THEN CALL gameover ELSE CALL expend
WEND 'main

'gestione collisioni
coll:
obj=COLLISION(0)
WHILE obj<>0
colpre(obj)=COLLISION(obj):obj=COLLISION(0)
WEND
RETURN 'end coll

SUB music STATIC
SHARED puntmot,puntnot,contmot,suono
SOUND WAIT
ON suono GOSUB acc,colp,manc
SOUND RESUME:EXIT SUB
acc:
SOUND mus(puntmot,puntnot),1,70
IF puntnot<3 THEN puntnot=puntnot+1 ELSE puntnot=0
IF contmot=2 THEN
IF puntmot<3 THEN puntmot=puntmot+1 ELSE puntmot=0
END IF
RETURN
colp:
FOR j=5 TO 2 STEP -1:SOUND 50*j,,5,255,2:NEXT:suono=1:RETURN
manc:
FOR j=1 TO 4:SOUND 600*j,,3,255,3:NEXT:suono=1:RETURN
END SUB 'music

SUB attac STATIC
SHARED lancio,obiet,xnem,ynew
y1=OBJECT.Y(1):x2=OBJECT.X(2):y2=OBJECT.Y(2)
SOUND 1000,,5,255,2:xnem=x2:ynew=y1+7
OBJECT.X obiet,x2:OBJECT.Y obiet,ynew-2
lancio=1:OBJECT.VY 3,-50
OBJECT.X 3,x2:OBJECT.Y 3,y2+12
OBJECT.ON obiet,3:OBJECT.START 3
END SUB 'attac

SUB expnem STATIC
SHARED lancio,obiet,xnem,ynew,distru,contmot,suono,attacco,score
IF colpre(3)=4 OR colpre(3)=5 THEN
lancio=0:colpre(3)=0:attacco=0:OBJECT.OFF 3,obiet
IF POINT(xnem+1,ynew-11)=6 THEN
PAINT(xnem+1,ynew-11),4,2
suono=2:distru=distru+1:score=score+10
IF contmot<2 THEN contmot=contmot+1 ELSE contmot=0
ELSE
suono=3
END IF
IF obiet=4 THEN obiet=5 ELSE obiet=4
END IF
END SUB 'expnem

SUB update1 STATIC
SHARED nx,ny,fx,fy,delta
x2=OBJECT.X(2):y2=OBJECT.Y(2)
IF y2+ny>fy+9 AND dy=1 THEN
dy=0:OBJECT.Y 2,fy+9-ny
END IF
IF y2<=delta+13 AND dy=-1 THEN
dy=0:OBJECT.Y 2,delta+13
END IF
IF x2+nx>=fx+4 AND dx=1 THEN
dx=0:OBJECT.X 2,fx+4-nx
END IF
IF x2<=9 AND dx=-1 THEN
dx=0:OBJECT.X 2,9
END IF
vx=dx*25:vy=dy*25
OBJECT.VX 2,vx:OBJECT.VY 2,vy:OBJECT.VX 1,vx:OBJECT.VY 1,vy
OBJECT.X 1,OBJECT.X(2):OBJECT.Y 1,OBJECT.Y(2)-delta 'sincroniz.
END SUB 'update1

SUB defen STATIC
SHARED ultcol
SOUND 4000,,25,255,2:RANDOMIZE TIMER:PALETTE 3,RND,RND,RND
ON ABS(colpre(6)) GOSUB top,left,bottom,right
ultcol=colpre(6):colpre(6)=0
OBJECT.START 6:EXIT SUB
top:
OBJECT.VY 6,100*RND+60:RETURN
left:
OBJECT.VX 6,100*RND+60:RETURN
bottom:
OBJECT.VY 6,-(100*RND+60):RETURN
right:
OBJECT.VX 6,-(100*RND+60):RETURN
END SUB 'defen
SUB expnav STATIC
SHARED numnav
colpre(2)=0:numnav=numnav-1
OBJECT.START 6:OBJECT.STOP 2,1
FOR i=1 TO 6
f=i MOD 2:PALETTE 0,f,0,0:PALETTE 1,f,0,0:PALETTE 3,f,0,0
IF f THEN OBJECT.OFF 1,2 ELSE OBJECT.ON 1,2
SOUND WAIT:FOR j=12 TO 2 STEP -1:SOUND 50*j,,25,255,1:NEXT:SOUND RESUME
NEXT
OBJECT.START 2,1
END SUB 'defen
SUB initcol STATIC
RESTORE datacol:'colori registri
datacol:
'0=nero sfondo,1=nero finestra
DATA 0,0,0,0,0,0
'2=nero,3=casuale per difesa
DATA 0,0,0,0,0,0
'4=verde,5=bianco
DATA .33,.87,0,1,1,1
'6=rosso fuoco,7=grigio
DATA .93,.20,0,.7,.7,.7
FOR i=0 TO 7:READ r',g',b':PALETTE i,r',g',b':NEXT
END SUB 'initcol
SUB initmus STATIC
DIM shape(255)
RESTORE datamus:'musica fondo
datamus:
DATA 1046.37,1567.46,1318.42,1567.46
DATA 1046.37,1760.1396,88.1760
DATA 783.73,1567.46,1174.02,1567.46
DATA 1174.02,2348.05,1760.2348.05
FOR i=0 TO 3:FOR j=0 TO 3:READ mus(1,j):NEXT:NEXT
WAVE 0,SIN:WAVE 1,SIN:WAVE 3,SIN
FOR j=0 TO 3:SOUND mus(1,j),1,255:NEXT
FOR i=0 TO 255:shape(i)=1.53*(i MOD 13 +1)-127:NEXT:WAVE 2,shape
END SUB 'initmus
SUB initmat STATIC
SHARED lm,hm
RESTORE datainit:'creazione mattonella
datainit:
LINE (1,1)-(lm,hm),5,bf:LINE (10,10)-(lm-9,hm-9),6,bf:COLOR 7
AREA(1,1):AREA(lm,1):AREA(lm-9,10):AREA(10,10):AREAFILL
AREA(lm,1):AREA(lm,hm):AREA(lm-9,hm-9):AREA(lm-9,10):AREAFILL
LINE (1,1)-(lm,hm),3,b:LINE (10,10)-(lm-9,hm-9),2,b
LINE (1,1)-(10,10),2:LINE (lm,1)-(lm-9,10),2
LINE (1,hm)-(10,hm-9),2:LINE (lm,hm)-(lm-9,hm-9),2
GET (1,1)-(lm,hm),mat
END SUB 'initmat
SUB init STATIC
SHARED lm,hm,fx,fy,ultcol
'ciclo creazione pavimento
FOR y=0 TO 5
FOR x=0 TO 3:PUT(x:lm,y:hm),mat:PUT((8-x):lm,y:hm),mat:NEXT
PUT(4:lm,y:hm),mat
NEXT
OBJECT.X 1,151:OBJECT.Y 1,194
OBJECT.X 2,151:OBJECT.Y 2,154
OBJECT.CLIP (14,14)-(fx+3,fy+8)
OBJECT.HIT 1,0,0:OBJECT.HIT 2,4,0:OBJECT.HIT 6,0,4
OBJECT.X 6,50:OBJECT.Y 6,40
IF ultcol=-3 OR ultcol=-2 THEN y=-100 ELSE y=100

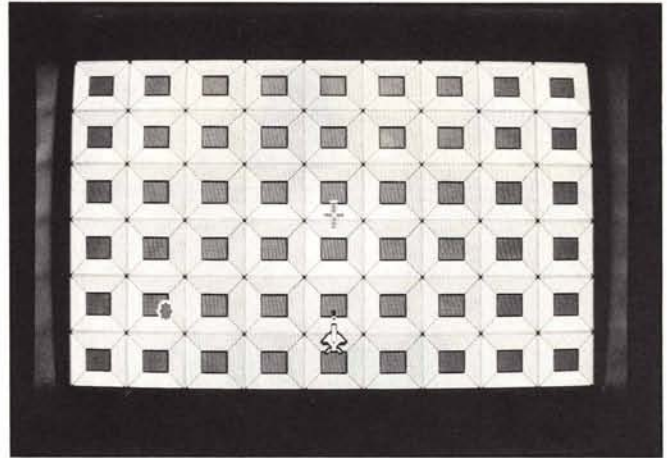
```


Anche la grafica è curata sufficientemente bene (un no! alla musica) ma soprattutto il bello del programma... il programma! (ricordate la pubblicità del pollo?, a me fa letteralmente impazzire!).

È veramente ben strutturato, ha finalmente la vera forma di un programma (dichiarazioni-inizializzazioni, main program, procedure) i nomi delle variabili e delle procedure sono intuitivi e non le solite x, y, a, b, c. Troviamo ad esempio la procedura inimat che disegna una mattonella e l'associa al nome mat; troviamo inimus per la musica, inicol per i colori, inif per inizializzare il campo di gioco. Discorso analogo per le procedure usate nel main, attac, defen, expnem, expnav. A tutto questo aggiungiamo la musica che cambia ogni due mattonelle colpite e ci festeggia *amighevolmente* ogni quadro completato, più la gestione delle collisioni sprite-sprite sottoforma di interrupt, altra caratteristica interessantissima dell'AmigaBasic.

Volendo, a tutti i costi, fare un piccolo appunto potremmo consigliare l'autore e, giacché siamo nelle pagine di una rivista, i lettori che si accingono a scrivere in AmigaBasic, di non abusare troppo del costrutto SHARED per la condivisione delle variabili tra main e procedura, brutto informatica-

Il campo di gioco.



mente parlando e sostituibile senza sforzi con una più elegante lista di parametri formali con la quale interfacciare la procedura. Prendendo questa abitudine, anche se non è il caso del giochino navicelloso, potremmo cominciare a pensare in termini di procedure in modo da crearci una collezione di utility personalizzate da utilizzare anche in programmi futuri. Il meccanismo del passaggio dei parametri, infatti, permette di svincolare i nomi usati dal programma principale da quelli adoperati nel corso della proce-

dura e quindi una stessa procedura, senza effettuare nessuna modifica, può essere usata in programmi differenti. Ma anche di questo avremo modo di riparlarne in Amighevole, a tutti voi buon lavoro e un arrivederci.

Questo programma è disponibile su disco presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 229.

```

OBJECT.VX 6,-y:OBJECT.VY 6,y
OBJECT.HIT 3,2,0:OBJECT.HIT 4,0,2:OBJECT.HIT 5,0,2
OBJECT.ON 1,2,6:OBJECT.START 1,2,6
COLLISION ON
END SUB 'init
SUB copyright STATIC
CLS:COLOR 6
PRINT "Copyright 1987 by Dante Sbraga"
PRINT "via dei Della Bitta n 13"
PRINT "00148 Roma":PRINT "tel 06/6855868":PRINT
PRINT "Regolare il volume":PRINT
COLOR 4:PRINT "Usare il joystick nella porta #2":PRINT:COLOR 6
PRINT "Premere a fondo il pulsante":PRINT
PRINT "Attenzione alla sfera del nemico":PRINT
PRINT "Missione:distuggere piramidi rosse":PRINT
COLOR 4:PRINT "Regolare il volume":PRINT :COLOR 6
PRINT "Attendere prego...";
END SUB
SUB expend STATIC
  SHARED numnav,score
FOR i=1 TO 8
  SOUND WAIT:FOR j=1 TO 12:SOUND 200*j,..3,255,3:NEXT:SOUND RESUME
  PALETTE 4,i MOD 2,(i+1) MOD 2,0
NEXT
FOR i=1 TO 8
  FOR j=12 TO 1 STEP -1:SOUND 200*j,..5,255,3:NEXT
  PALETTE 3,i MOD 2,(i+1) MOD 2,0
NEXT
CLS
FOR i=1 TO 10
  FOR j=5 TO 10:SOUND 2.5*j,..25,255,3:PALETTE 0,j MOD 2,0,0:NEXT
NEXT
numnav=numnav+3
PRINT "***** READY STAGE ":score\540;" *****":PRINT
PRINT "Score=":score:PRINT
PRINT "F15=":numnav:FOR i=1 TO 8000:NEXT
END SUB 'expend
SUB gameover STATIC
  SHARED numnav,score
CLS:PRINT "***** GAME OVER *****":PRINT
PRINT "score=":score:PRINT

```

```

FOR j=0 TO 3:SOUND mus!(1,j),1,255:NEXT
INPUT "Vuoi giocare ancora (y/n)":n$
FOR j=0 TO 3:SOUND mus!(1,j),1,255:NEXT
IF n$<>"n" AND n$<>"N" THEN
  numnav=5:score=0:EXIT SUB 'gioco ancora
END IF
OBJECT.CLOSE:WINDOW CLOSE 2:SCREEN CLOSE 1
PALETTE 0,..25,..25,0:PALETTE 1,1,1,1:END
END SUB
SUB initobj STATIC
RESTORE nif
nif:
DATA $H0000,$H0000,$H0000,$H0000,$H0000,$H0000,$H0002,$H0000,$H0010,$H0000,$H0010
DATA $H0019,$H0003,$H0000,$H03C0,$H0000,$H0240,$H0000,$H0240,$H0000,$H0240
DATA $H0003,$H0003,$H0240,$H0000,$H0240,$H0000,$H0240,$H0000,$H03C0,$H03C0
DATA $H03C0,$H03C0,$H03C0,$H03C0,$H03C0,$H03C0,$H03C0,$H03C0,$H03C0,$H03C0
DATA $H03C0,$H03C0,$H03C0,$H03C0,$H03C0,$H000F,$H000F,$H00F1
n=48:GOSUB readdata:OBJECT.SHAPE 1,s$
'nav
DATA $H0000,$H0000,$H0000,$H0000,$H0000,$H0002,$H0000,$H0010,$H0000,$H001B
DATA $H0019,$H0003,$H0000,$H0000,$H0000,$H0000,$H001B,$H0000,$H01B0,$H01B0
DATA $H0000,$H01B0,$H01B0,$H01B0,$H01B0,$H01B0,$H05A0,$H0DB0,$H1DB8,$H3DBC
DATA $H0660,$H03C0,$H01B0,$H01B0,$H05A0,$H0810,$H0000,$H01B0,$H01B0,$H01B0
DATA $H0000,$H01B0,$H0000,$H0000,$H03C0,$H0240,$H0240,$H03C0,$H07E0,$H00F0
DATA $H1FB8,$H3FFC,$H7FFE,$HFFFF,$H7FFE,$H07E0,$H00F0,$H1FB8,$H3DBC
DATA $H1B1B,$H0FB0,$H0000,$H04BF
n=64:GOSUB readdata:OBJECT.SHAPE 2,s$
'mis
DATA $H0000,$H0000,$H0000,$H0000,$H0000,$H0002,$H0000,$H0010,$H0000,$H000B
DATA $H0019,$H0003,$H0000,$H1B1B,$H0000,$H0000,$H0000,$H1B1B,$H1B1B,$H1B1B
DATA $H1B1B,$H1B1B,$H1B1B,$H1B1B,$H3C3C,$H3C3C,$H7E7E,$H7E7E,$H0000,$H00F0
DATA $H0111,$H0F00
n=32:GOSUB readdata:OBJECT.SHAPE 3,s$
'mark
DATA 0,0,0,0,0,2,0,16,0,6,27,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DATA $HFFF,$HFFF,$HFFF,$HFFF,$HFFF,$HFFF,$HFFF,$HFFF,$HFFF,$HFFF,$HFFF,$HFFF
n=34:GOSUB readdata:OBJECT.SHAPE 4,s$:OBJECT.SHAPE 5,s$
'fire
DATA $H0000,$H0000,$H0000,$H0000,$H0000,$H0002,$H0000,$H0010,$H0000,$H000C
DATA $H0019,$H0003,$H0000,$H03C0,$H0660,$H0B10,$H0B10,$H0B10,$H100B,$H100B
DATA $H0B10,$H0B10,$H0B10,$H0660,$H03C0,$H03C0,$H07E0,$H00F0,$H00F0,$H00F0
DATA $H1FB8,$H1FB8,$H0F0,$H0F0,$H0F0,$H0F0,$H07E0,$H03C0,$H0000,$H00F0,$H00F1
n=40:GOSUB readdata:OBJECT.SHAPE 6,s$
EXIT SUB
readdata:
s$="":FOR i=1 TO n:READ w:s$=s$+H$(w):NEXT:RETURN
END SUB

```