



Lander 64

Rosario De Chiara

```
1000 REM INIT
1010 V=53248:GOSUB 2400

1020 REM INIT LEVEL
1100 X=150: Y=50: G=0.1: VV=0: HV=0: EN=13:GOSUB2900

1200 REM GAME LOOP
1210 J=255-PEEK(56320):U=(JAND2)/2-(JAND1):C=(JAND8)/8-(JAND4)/4
1300 Y=Y+VV: VV=VV+G*0.5
1400 X=X+HV: HV=HV+C*0.5
1500 IF X>252 OR X<10 THEN VX=0
1600 IF Y>230 OR Y<44 THEN VY=254
1700 EN=13: IF U <> 0 THEN EN=14:POKE 54296,255:POKE 54296,0
1800 POKE V,X:POKE V+1,Y: POKE 2040,EN
1900 SX=INT(X/8)-2:SY=INT(Y/8)-5:LL=PEEK(1024+5X+5Y*40)
1910 POKE781,1:SY$59903
2000 PRINT "{home}x:";SX;" y:";SY;" vert vel:";INT(VV*100);" "
2100 IF LL=98 THEN GOTO 4300
2200 IF LL<>32 THEN PRINT "crashed":END
2300 GOTO 1200

2400 REM INIT SPRITES
2500 BS=832:SL=127
2600 FOR Z=0 TO SL: READ J: POKE BS+Z,J: NEXT Z
2700 POKE V+21,3: POKE V+39,7: POKE V+33,0: POKE V+32,0:POKE V+40,1
2800 POKE 56322,224
2810 RETURN

2900 REM INIT FIELD
2910 PRINT PEEK(V+30):PEEK(V+31):CHR$(147)
3000 PX=1024+15*40:BP=INT(RND(TI)*20)+10
3100 FOR I=0 TO 30
3200 D=INT(RND(TI)*3+1)
3300 IF I=BP THEN POKE PX,98:POKE PX+1,98:I=I+2:PX=PX+2:BH=INT(PX/40)
3400 IF PX>2023-40 AND D=3 THEN D=1
3500 IF PX<1024+10*40 AND D=1 THEN D=3
3600 IF D=1 THEN POKE PX,75:PX=PX-40:POKE PX,85
3700 IF D=2 THEN POKE PX,67
3800 IF D=3 THEN POKE PX,73:PX=PX+40:POKE PX,74
3900 PX=PX+1
4000 NEXT I
4200 RETURN

4300 REM LANDING CONDITIONS
4400 IF VV < 1.5 THEN PRINT "you win. prepare for next
level":FOR I=0 TO 1000:NEXT I:GOTO 1100
4500 PRINT "too fast":END

4600 REM ASTROBLASTER SPRITE
4800 DATA 0,0,0,0,0,0,0,0,0,15,240,0,16,8,0,39,4,0,40,148,0,42,164,0,40
4900 DATA 148,0,39,4,0,16,8,0,15,240,0,15,240,0,16,8,0,32,4,0,112,14,0
5000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
5100 DATA 0,15,240,0,16,8,0,39,4,0,40,148,0,42,164,0,40,148,0,39,4,0,16,8
5200 DATA 0,15,240,0,15,240,0,16,8,0,32,4,0,112,14,0,0,0,0,48,12,0,48,12
5300 DATA 0,16,8,0,0,0,0,0
```

Inizializzazione degli sprite e del livello. La riga 1000 viene eseguita una volta soltanto, mentre la 1020 viene eseguita ad ogni nuovo livello. Le variabili X,Y sono la posizione del lander, G è la gravità, HV e VV sono le velocità orizzontale e verticale

Loop del gioco: J contiene la posizione del Joystick e viene scomposto in posizione orizzontale e verticale (C ed U risp.) Righe 1300 e 1400: vengono aggiornate le posizioni X ed Y e le velocità HV e VV (notare che VV tiene conto della gravità G) 1500-1600: viene mantenuto lo sprite nella parte a sinistra dello schermo.

1700: viene controllato il pulsante del joystick, se premuto allora viene generato un suono (variando il volume). Con la variabile EN manteniamo il puntatore al banco dello sprite in modo da scegliere opportunamente quale visualizzare

Viene aggiornata la dashboard e verificato il carattere sotto lo sprite: in base al codice (righe 2100-2200) si capisce se è atterrato o se si è schiantato

Inizializzazione degli sprite

Inizializzazione del profilo delle montagne: PX contiene la posizione del prossimo carattere che dobbiamo disegnare, ad ogni iterazione D contiene un numero casuale fra 1 e 3. In base a D viene scelto se «salire», «pianeggiare» oppure «scendere»: a secondo della scelta PX viene aggiornato (risp. sommando -40, invariato o +40) e vengono disegnati i caratteri opportuni. Alla riga 3500 viene disegnata la base d'atterraggio la cui posizione casuale è contenuta in BP

Condizioni di fine gioco oppure di passaggio di livello