

Attacco medioevale

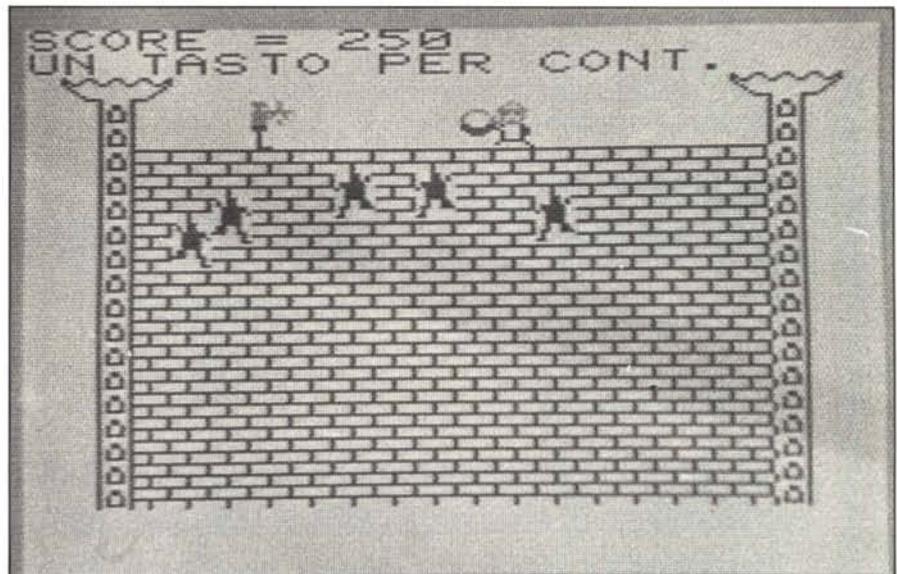
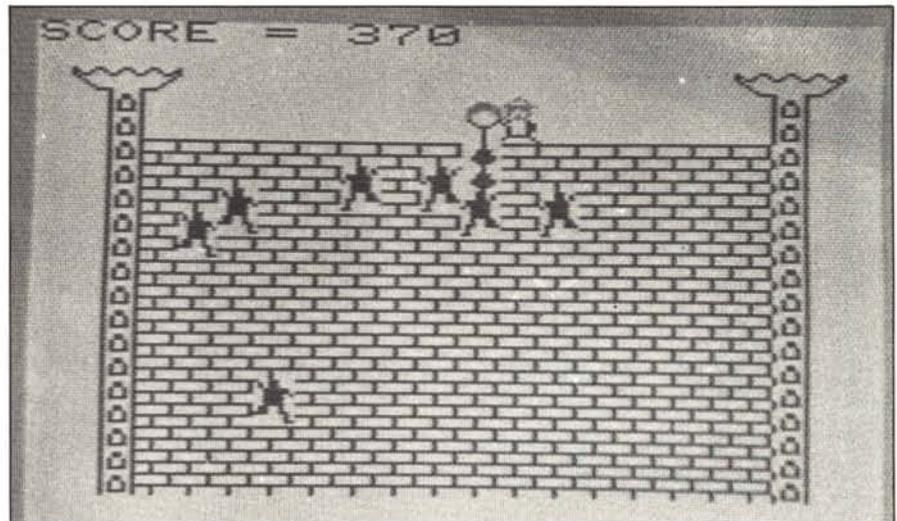
Il lettore Claudio Massa di Torino ci invia un gioco per il VIC 20 dalla presentazione originale e curata: consiste nella simulazione di un attacco barbaro ambientato nel mondo medioevale. Dopo l'inizializzazione del gioco lo schermo mostra un castello medioevale, con tanto di torri, visto frontalmente, attaccato da tredici omini — i barbari — che si arrampicano sul muro: oltre al semplice moto verso l'alto la simulazione della scalata è resa credibile dall'alternarsi di due posizioni per le gambe, proprio come chi realmente si arrampichi. Il nostro eroe, rappresentato da un omino posto sul cornicione superiore del castello, può muoversi a destra e a sinistra e deve colpire i barbari con l'olio bollente del suo pentolone. Non tutti gli invasori che giungono sul cornicione sono pericolosi: possono infatti scivolare in linea 910 (la condizione è affidata ad un controllo su un numero casuale) e tornare giù (ma decrementano il punteggio), oppure issarsi e minacciare il difensore con un arco. Il gioco finisce alla fine dei tredici omini, oppure quando uno di essi ha sotto mira l'unico, strenuo milite. Per ora abbiamo un record personale di 1080 punti...

Le tecniche di programmazione si risolvono nella definizione di caratteri da parte dell'utente, e nella loro allocazione in uno solo dei quattro blocchi da 64 caratteri previsti dall'ASCII (e implementati sul VIC): questa condizione, resa possibile modificando il contenuto del registro di controllo numero 6 — posto nella locazione 36869 — aggiungendo il decimale 15 al suo normale contenuto (240) toglie solo mezzo KB al Basic destinato al programma, che quindi assomma a circa 3 KB, sufficienti per un discreto programma. In effetti tolti i REM la simulazione dell'attacco medioevale lascia libero circa un KB, utile per ulteriori modifiche (maggior complessità, uso dei paddle, etc.). I normali caratteri del VIC vengono a trovarsi al

posto di quelli in campo inverso: avremmo preferito un'altra soluzione.

Le osservazioni al listato originale sono

poche, solo due, e abbastanza marginali dato che il programma funziona. La prima riguarda la posizione dei tasti usati per



REMARKs essenziali

- 120-360 Ciclo di caricamento dei caratteri speciali e relativo blocco DATA
- 370 Nuovo valore del puntatore dei caratteri e del volume
- 400-450 Presentazione gioco
- 490-650 Disegna il castello e gestisce gli invasori
- 670-740 Disegna ed anima il nostro eroe
- 790-800 Chiude il ciclo di spostamento degli avversari
- 830-940 Subroutine che determina cosa accade se un invasore è giunto in cima
- 945-1000 Subroutine in gioco quando è stato colpito un avversario
- 1010-1120 Subroutine che disegna la caduta dell'olio bollente e che verifica se il bersaglio è stato colpito.

Principali VARIABILI usate

- X posizione dell'omino difensore, mosso da tastiera
- J è il codice del simbolo impostato dalla tastiera
- SC dà il punteggio
- SV contiene il numero di avversari centrati
- Z è la locazione di memoria di schermo ove si trova l'olio

```

60 REM*****
70 REM** SIMULAZIONE DI ATTACCO MEDIOEVALE**
80 REM** BY CLAUDIO MASSA**
90 REM*****
100 PRINT"J":POKE56,28
120 FORT=0T0511:READA$:IFA$="*"THEN370
140 POKE7168+T,VAL(A$):NEXTT
160 DATA255,1,1,1,255,16,16,16,134,201,176,64,32,16,8,7,24,36,195
200 DATA0,0,0,0,129,97,147,13,2,4,8,16,224,129,153,165,165
220 DATA165,189,129,1,25,25,37,37,165,189,129,152,88,88,62,62,61,60
230 DATA60,60,100,100,230,2,2,3,0,25,26,26,124,124,188,60
240 DATA60,60,38,38,103,64,64,192,0,0,0,28,34,65,65,65,0,0
270 DATA28,34,65,65,65,16,40,68,254,108,68,84,184,99,127,62,28,0
300 DATA0,0,0,68,134,70,70,68,124,130,1,0,0,28,62,127,99,65,65
330 DATA65,34,28,8,8,8,8,8,8,28,28,62,62,28,8,39,39,66,255,87
350 DATA79,39,39,7,7,2,2,2,2,14,228,228,68,255,234,242,228,228
360 DATA224,224,64,64,64,64,112,*
370 POKE36869,255:POKE36866,PEEK(36866)OR128:POKE36878,15
400 FORT=1T0505:PRINT"J":NEXTT
410 PRINT"SIMULAZIONE DI ATTACCO":PRINT"J:DDDDDD:MEDIOEVALE"
430 PRINTTAB(9);"XBY":PRINT"X:DDDD:CLAUDIO MASSA":PRINT"X:DDDD:TASTI DA USARE"
450 PRINTTAB(8);"XJUNK":PRINT"X:DDDD:UN TASTO IMPERICO.":
480 IFPEEK(197)=64THEN490
490 FORT=38400T038905:POKET,2:NEXTT
500 FORT=38400+3*22+4T038400+3*22+17:POKET,5:POKET+22,6:NEXTT
510 FORT=7680T08185:POKET,160:NEXTT
520 FORT=7790T08098STEP22:POKET+2,4:POKET+19,5
540 FORR=3T018:POKET+R,0:NEXTR,T
550 FORT=7725T07742STEP17:POKET,1:POKET+1,2:POKET+2,3:POKET+23,4:POKET+45,4:NEXT
570 DIMY(14),A(14):X=8
580 FORT=1T013:Y(T)=INT(RND(1)*10)+2:A(T)=1:NEXTT
610 FORR=0T02STEP2
630 FORT=1T013:IFR(T)=0THEN800
640 IFY(T)=0THENGOSUB830
650 POKE7790+3+T+22*Y(T),6+R:POKE7812+3+T+22*Y(T),7+R:POKE7834+3+T+22*Y(T),0
670 J=PEEK(197)
680 IFJ=43ANDX>4THENX=X-1
690 IFJ=44ANDX<16THENX=X+1
700 POKE7746+X-1,160:POKE7746+X,11:POKE7746+X+1,12:POKE7746+X+2,160
740 POKE7768+X-1,160:POKE7768+X,13:POKE7768+X+1,14:POKE7768+X+2,160
780 IFJ=20THENGOSUB1010
790 Y(T)=Y(T)-1
800 NEXTT,R
810 IFSV>12THEN1130
820 GOT0610
830 REM*SE L'INVASORE E' GIUNTO IN CIMA*
840 FORS=255T0128STEP-4:POKE36876,S
850 FORS1=1T020:NEXTS1,S:POKE36876,0:SC=SC-20
860 PRINT"SCORE=";SC;" "
880 POKE7790+3+T+22*Y(T),0:POKE7812+3+T+22*Y(T),0:POKE7834+3+T+22*Y(T),0
900 Y(T)=10
910 IFRND(1)>.3THENRETURN
920 IFT+2>XTHENPOKE7746+T+3,18:POKE7768+T+3,19:GOTO1130
930 IFT+2<XTHENPOKE7746+T+3,20:POKE7768+T+3,21:GOTO1130
940 RETURN
945 REM*AVVERSARIO COLPITO*
950 FORS=128T0255STEP8:POKE36876,S
960 FORS1=1T010:NEXTS1,S:POKE36876,0
970 SC=SC+T*10
980 PRINT"SCORE=";SC;" "
990 A(X-3)=0:SV=SV+1
1000 RETURN
1010 REM*OLIO VERSATO*
1020 POKE36877,253:POKE7746+X,15:POKE7768+X,16
1040 FORZ=7790+XT07878STEP22
1050 IFPEEK(Z)=0THENGOSUB950:GOTO1080
1060 POKEZ,17:NEXTZ
1080 POKE36877,0
1100 FORZ=7790+XT07878+XSTEP22:POKEZ,0:NEXTZ
1120 RETURN
1130 PRINT"UN TASTO PER CONT."
1140 FORT=1T01000:NEXT
1150 IFPEEK(197)=64THEN1150
1160 CLR:GOTO490

```

READY.

controllare il moto dell'omino difensore e del relativo 'versamento olio': il programma usa il trio H-J-K, ove il tasto centrale che serve allo sparo risulta poco maneggevole, sia per l'eccessiva vicinanza dei controlli che per la loro scarsa maneggevolezza, con evidenti ritardi nell'ergersi a difesa del presidio (buona questa!): è comunque un fattore secondario.

Leggermente più importante è il RUN in linea 1160. La funzione dell'istruzione è evidente dal listato stesso: a fine programma c'è un'istruzione ricorsiva in linea 1150 la cui esecuzione termina premendo una qualsiasi chiave, passando al succitato RUN. L'esecuzione di questa istruzione comporta una nuova inizializzazione di tutte le variabili e zone di memoria usate nel programma, con lettura del blocco DATA e successiva presentazione del gioco: un ritardo notevole nell'economia del gioco. Il blocco iniziale può esser saltato a piè pari ponendo in linea 1160 un CLR:GOTO 490 al posto del RUN.

Al termine dell'articolo vogliamo spendere alcune parole sulle particolarità mostrate dal programma. In primis la possibilità di sostituire più Next consecutivi, solitamente nella forma

```
NEXT I : NEXT J
```

nella più sintetica NEXT I,J come mostrato nel listato ad esempio in linea 540, 800, 840 e 960 per le variabili di conteggio R e T, oppure S1 ed S.

In secundis facciamo osservare la sostanziale — e sostanziosa — divergenza tra le linee

```
100 IF A=1 THEN PRINT C :
PRINT D
```

e

```
100 IF A=1 THEN PRINT C
110 PRINT D
```

infatti, nel caso di più istruzioni su una linea (separate dai due punti) successive ad un IF, queste verranno eseguite se e solo se è verificata la condizione sull'IF. Nel primo esempio il VIC stamperebbe sia C che D solo per A=1, mentre nel secondo D verrebbe comunque scritto, eventualmente preceduto da C nel caso in cui fosse A=1.

Leo Sorge

Stiamo vagliando i programmi che ci avete inviato. Ricordate che la preferenza va ai lavori che girino sul VIC inespanso, nudo e crudo con 3,5 K Ram, e che piuttosto che giochi, vecchi o nuovi (a parte alcune eccezioni, ehm) interessano subroutine e trucchetti di programmazione.

Nel caso di programmi di una certa lunghezza, comunque, l'ideale è accludere al listato una cassetta con il programma registrato tre o quattro volte di fila, per evitare notti insonni al povero operatore...