

BREAKVIC

BREAKVIC è un videogioco scritto in BASIC per il Commodore VIC-20. E' stato scritto per una competizione utilizzando solo 20 linee di codice.

L'obiettivo del progetto è stato quello di inserire in 20 linee di codice un gameplay completo, suoni e grafica. Per la grafica sono stati utilizzati solo i caratteri PETSCII. Il gameplay consiste nel distruggere tutti i blocchi presenti sullo schermo. Il gioco termina con un punteggio finale quando tutti i blocchi sono stati distrutti. Il punteggio finale è calcolato nel seguente modo:

- 300 punti vengono assegnati all'inizio del gioco
- -100 punti quando la palla colpisce il bordo inferiore dello schermo senza colpire la barra del giocatore
- +100 punti quando la palla distrugge un blocco o rimbalza sulla barra del giocatore

Il giocatore controlla la barra utilizzando il joystick.

Note

- Per ridurre la dimensione del codice le parole chiave **print** e **poke** sono state sostituite dalle loro forme BASIC abbreviate (**?** e **pO**).
- Il codice è scritto utilizzando l'IDE "CBM .prg Studio". Il codice contiene alcune particolari sequenze di controllo, come {clear}, {white}, ... Dopo aver caricato il file prg con un emulatore o su hardware reale è possibile visualizzare il programma con il comando LIST e vedere i caratteri originali che sostituiscono le sequenze di controllo.

Note per la competizione

- Il file breakvic_joy.bas è il sorgente per la competizione. I file breakvic.prg e breakvic_joy.prg sono generati da questo file sorgente.

Problemi noti

- Il colore della palla cambia quando la palla è in una locazione dello schermo precedentemente occupata da un blocco
- Le coordinate x,y della palla sono sempre incrementate/decrementate di 1, solo quando la palla colpisce la barra del giocatore la coordinata x è incrementata/decrementata di 2 per simulare un effetto velocità

Commenti al codice

La linea 10 inizializza il volume del suono (poke 36878,15) e il colore del bordo e dello sfondo dello schermo (poke 36879,12), dopo pulisce lo schermo e imposta a bianco il colore dei caratteri. In questa linea vengono inizializzate alcune variabili:

- **x, y**: le coordinate della palla
- **dx,dy**: i valori utilizzati per incrementare/decrementare x e y
- **bx**: la coordinata x della barra del giocatore
- **c**: la locazione della memoria dello schermo dove viene inserito il primo blocco
- **p**: come c ma punta alla memoria colore
- **v**: la velocità della palla

```
10 pO36878,15:pO36879,12:"?">{clear}  
{white}":x=10:y=21:dx=-1:dy=-1:bx=10:c=7727:p=38447:v=1
```

La linea 20 inizializza le variabili **t** (il punteggio del giocatore) e **m** (il numero totale di blocchi). Le altre istruzioni sulla linea disegnano i blocchi sullo schermo: 15 blocchi per ogni linea e una linea vuota tra due file di blocchi. Nel ciclo per disegnare i blocchi sia la memoria dello schermo (poke c+i,102) sia la memoria colore (poke p+i,3) vengono scritte.

```
20  
t=300:m=75:fori=1to15:pOc+i,102:pOp+i,3:nexti:c=c+44:p=p+44:ifc<7944  
goto20
```

La linea 30 disegna la barra del giocatore (carattere 160) a centro dell'ultima linea sullo schermo, la barra è lunga due caratteri. Inoltre, il carattere spazio (code 32) è scritto sulle coordinate della palla (poke7680+y*22+x,32) per cancellare la precedente palla sullo schermo.

```
30 poke8164+bx,160:poke8164+bx+1,160:poke7680+y*22+x,32
```

Le linee 40 e 50 contengono del codice standard per leggere il joystick. Se **left** è individuato e **bx>0** la barra è spostata verso sinistra saltando alla linea 110. La linea 50 individua la direzione **right** e se **bx<20** sposta la barra a destra saltando alla linea 120.

```
40 j=peek(37151):if(jand16)=0andbx>0thengoto110  
  
50  
poke37154,127:j=peek(37152):poke37154,255:if(jand128)=0andbx<20the  
ngoto120
```

La linea 60 silenzia la prima voce del VIC e incrementa le coordinate x,y. La coordinata x è moltiplicata per la velocità. L'istruzione **if** controlla se la palla colpisce il bordo sinistro/destro dello schermo o se la palla colpisce un blocco (peek(7680+y*22+x)=102).

60

```
poke36874,0:x=x+dx*v:y=y+dy:ifx<=0orx>=21orpeek(7680+y*22+x)=102t  
hendx=-dx
```

La linea 80 silenzia la seconda voce del VIC e controlla se la coordinata y coincide con i bordi dello schermo o un blocco.

```
poke36875,0:ify=0ory=21orpeek(7680+y*22+x)=102thendy=-dy
```

Le linee 90 e 91 controllano la coordinata x. Questo controllo è necessario perché quando la coordinata x è incrementata/decrementata con una velocità maggiore di 1 la coordinata può superare il margine sinistro/destro dello schermo.

```
90 ifx<0thenx=0
```

```
91 ifx>21thenx=21
```

La linea 92 controlla se la palla colpisce la barra del giocatore. Se la palla colpisce la barra viene eseguita la sub routine alla linea 140.

```
92 ify=21andx>bx-2andx<bx+3thengosub140
```

La linea 93 controlla se la palla colpisce un blocco. Se la palla colpisce un blocco viene eseguita la sub routine alla linea 130.

```
93 ifpeek(7680+y*22+x)=102thengosub130
```

Se la palla colpisce il bordo inferiore dello schermo ma non la barra del giocatore allora viene decrementato il punteggio del giocatore.

```
94 ify=21andnot(peek(7680+(y+1)*22+x))=160thent=t-100
```

Se il numero dei blocchi è uguale a 0 allora vai alla linea 150 (dove il gioco termina).

```
97 ifm=0thengosub150
```

La linea 100 disegna la palla sulle coordinate x,y e ritorna all'inizio del ciclo del gioco (linea 30).

```
100 poke7680+y*22+x,81:goto30
```

Le linee 110 e 120 sono eseguite quando il giocatore utilizza il joystick (vedi linee 40 e 50) per muovere la barra del giocatore.

```
110 poke8164+bx,32:poke8164+bx+1,32:bx=bx-1:goto60
```

```
120 poke8164+bx,32:poke8164+bx+1,32:bx=bx+1:goto60
```

La linea 130 è eseguita quanto la palla colpisce un blocco (vedi linea 92). La linea emette un suono sulla voce 1 del VIC, modifica la velocità della barra, incrementa lo score e decrementa il numero dei blocchi.

```
130 poke36874,135:v=1:t=t+100:m=m-1:return
```

La linea 140 è eseguita quanto la palla colpisce la barra del giocatore (vedi linea 93). La linea emette un suono sulla voce 2 del VIC e modifica la velocità della palla.

```
140 poke36875,135:v=2:return
```

Questa linea è eseguita quando il gioco termina (vedi linea 97). La linea imposta il volume a 0 e visualizza il punteggio del giocatore.

```
150 poke36878,0:?"{clear}{white}{down*9}{right*5}game over{down*2}  
{left*9}score: ";t:end
```