



```

received/sent via serial shift
register.
// Bit #4:
1 = Enable non
-
maskable interrupts
generated by positive edge on FLAG
pin.
lda #$01 //
st
a $d01a //Interrupt
control register

```

MAGGIO 2018 - [WWW.RETROMAGAZINE.NET](http://WWW.RETROMAGAZINE.NET)

## RetroMagazine

Anno 2 - Numero 7

Retrogaming, mini e Pac-Man, perché...

### ARTICOLI

- Una intro... per iniziare! - 2° puntata
- La migliore versione di BASIC
- **RetroMath** - La matematica delle carte di credito
- LABEN 70, una eccellenza italiana dimenticata
- Programmazione Atari 2600 - DPC+Kernel
- Una lunga storia iniziata nel 1947
- ABC della merenda... opsss... del neofita appassionato! - parte 2
- TheC64mini Joystick modding...
- **RetroTool** - datatool (C64)

### GIOCHI

- Anarchy (C64)
- Angolo Oscurita' - Lucky Luke (CD-i)
- Amos Newton - Mente Aliena (C64)
- Creatures 2 (C64)
- Manic Miner (ZX Spectrum)
- Intervista a Kevin Toms, il creatore di Football Manager + Recensione (ZX)

Eventi: Vicoretro' 2018

Chiusura ed anticipazioni...

IN EVIDENZA IN QUESTO NUMERO

## Retrogaming, mini e Pac-Man, perché la vita... non è piatta

di Edoardo Ullo

C'è qualche cosa di magico nel mondo dei videogiochi. Non per niente è uno dei miei passatempi preferiti e da qualche annetto sto provando a farlo diventare un lavoro.

E c'è qualcosa di magico se ancora discutiamo di Commodore 64, di Spectrum, di Amstrad, Amiga, Nintendo, Super NES, SEGA Master System, SEGA Megadrive e quant'altro. Perché quando torniamo a parlarne di questi computer e console d'epoca è un po' come se prendessimo la macchina del tempo. È un po' il nostro rifugio contro il logorio della vita moderna. C'è chi gioca, c'è chi programma, c'è chi scrive e diffonde. Ed io suono il gong provando a tenere il ritmo.

La tecnologia, che corre, offre però lo spazio a molte operazioni nostalgia, non ultima quella del THEC64 Mini che però già appartiene al recente passato. Pochi giorni fa, infatti, SNK (e qui rullo di tamburi) ha ufficializzato una nuova console in grado di fare girare alcuni grandi classici Neo Geo. Si chiama, manco a dirlo, Neo Geo Mini (poca fantasia nel nome, ma tant'è) ed ha le fattezze di un piccolo cabinato arcade con schermo da 3,5 pollici, 40 giochi pre-caricati e la possibilità di collegarlo tramite HDMI allo schermo. Da ufficializzare, invece, la lista ma i titoli come Metal Slug e King of Fighters non dovrebbero mancare. Tutto secondo copione salvo per la forma che ricorda i vecchi giochi da sala.

E c'è dell'altro che fa tornare in auge il mondo del retrogaming: Pac-Man, uno dei papà dei videogiochi, è balzato agli onori della cronaca per un motivo fuori di testa. I terrapiattisti (e già rido) avrebbero spiegato il principio per il quale, secondo loro, la Terra sia piatta prendendo ad esempio il classico di Toru Iwatani. Al limite dei confini si passa magicamente dall'altro lato da Est ad Ovest e viceversa come nei livelli di Pac-Man. E qui sorge il discorso teletrasporto. Chissà cosa ne pensa lo sviluppatore.

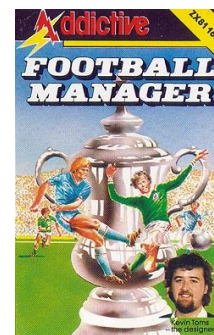
Ma non è soltanto di questo che volevo parlare nel mio editoriale perché c'è molta carne al fuoco anche nel numero 7 del nostro RetroMagazine. Il nostro Cercamon ha infatti realizzato un'intervista interessantissima a Kevin Toms, ovvero allo sviluppatore di Football Manager. Un vero e proprio mito vivente capace di realizzare un titolo che è poi stato capostipite di un genere ancora oggi in voga. Kevin Toms sta peraltro sviluppando un nuovo gioco di calcio manageriale su mobile. E quindi, corsi e ricorsi storici con il presente che è sempre più sospeso ed il retrogaming, e tante sue icone, che tornano più attuali che mai. Detto questo, vi auguro una buona lettura perché questo numero 7 è davvero ricco di contenuti da leggere.



### Per tutto il resto c'è RetroMath!

Senza badare a spese in questo numero Giuseppe ci parla dei calcoli matematici per la validazione delle carte di credito, ovviamente con codice retrocompatibile.

Articolo a pagina 8



### Intervista a Kevin Toms

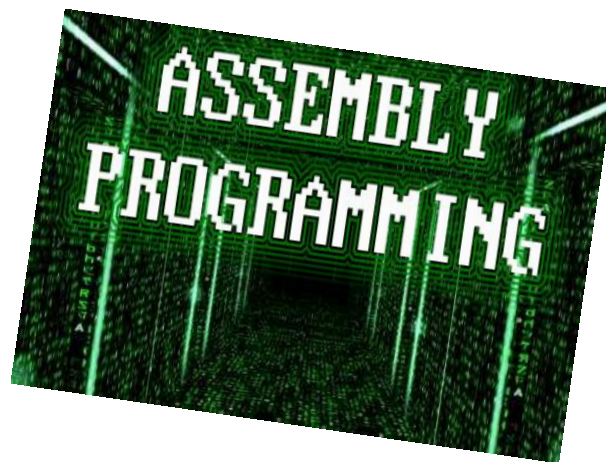
Il nostro David si è messo in contatto con Kevin Toms, il creatore di Football Manager, che non si è certo fatto pregare per rilasciare un'intervista piena di spunti di interesse per gli amanti del calcio manageriale simulato.

Articolo a pagina 31

# Una intro... per iniziare!

## Programmazione Assembly su Commodore 64 - seconda puntata

di Marco Pistorio



### Parte I – Introduzione

Benvenuti nella seconda "puntata" della intro dedicata a tutti i lettori di "RetroMagazine". Abbiamo già riscontrato un discreto interesse da parte di tutti Voi relativo alla prima "puntata", e questo è il motivo principale che mi spinge ad affrontare, insieme a Voi, questo seconda "chiacchierata" tra amici.

Ribadisco quanto già scritto nel precedente articolo, ovvero:

- 1) Lo scopo di questi articoli è quello di presentare una semplice intro, non una intro PERFETTA né tantomeno una intro che funzioni mediante tecniche particolarmente complesse.
- 2) Una infarinatura di base relativo all'assembly Vi sarà di aiuto per affrontare meglio tutti i passaggi più complessi che via via incontreremo
- 3) Ulteriori ragguagli potranno essere richiesti scrivendo, come sempre, alla nostra email redazionale.

Ed entriamo adesso più nel merito di questo secondo articolo sulle intro in assembly per C64.

Perché le barre risultano sfrangiate?

Bella domanda. Il motivo è essenzialmente uno, la "temporizzazione".

Cosa voglio dire?

Studiando più a fondo il funzionamento del raster video, scopriamo che ogni riga dello schermo viene "pennellata" in un tempo ben preciso.

Ad esempio, nella modalità video PAL ogni riga dello schermo verrà percorsa interamente in un tempo di 63 cicli macchina (65 in modalità NTSC). Non sto qui a farVi il calcolo preciso, ma posso assicurarVi che si tratta di un tempo molto breve.

Riflettiamo ora sul discorso che il pennello ottico scansisce ogni riga in un tempo ben preciso x, tempo abbastanza modesto, come Vi dicevo.

Nel nostro codice ho scelto arbitrariamente di fare una certa pausa prima di cambiare colore

allo schermo ed al bordo, poi, successivamente al loro cambiamento, perdo ancora un po' di tempo ed infine imposto nuovamente il colore di partenza, sia per il bordo che per lo schermo video, ovvero il colore nero.

Ma diventa chiaro a questo punto che se questi tempi non collimano perfettamente con il tempo necessario affinché le righe vengano percorse interamente, può succedere che il cambiamento del colore dello schermo parta nel momento "sbagliato", magari quando il raster si trova a metà di una certa riga che risulterà quindi colorata in parte.

Se, ad esempio, il raster si trova a percorrere la parte centrale di una riga ed io contestualmente ne cambio il colore, la parte più a sinistra (già percorsa) resterà di colore nero mentre la parte successiva risulterà colorata con un nuovo colore (rosso o blu nella nostra intro).

Stesso discorso dicasi per il momento in cui torniamo ad impostare il colore di base (nero) alla fine della nostra barra raster. Se il raster si trova a percorrere l'ultimo terzo della riga di schermo video, ed in quel momento coloriamo di nero la riga, tale cambiamento di colore varrà solo per la porzione di riga non ancora percorsa dal raster.

Le soluzioni da impiegare per evitare questo sgradevole effetto possono essere diverse.

Una di queste sarebbe di "contare" i cicli macchina per far sì che i cambiamenti di colore al bordo ed allo schermo video avvengano ESATTAMENTE ogni 63 cicli macchina.

Ci sarebbe una ulteriore complicazione però, le "bad lines". Studiando più approfonditamente l'argomento raster scopriamo che ogni 7 righe di schermo ce ne sta una ottava, una "bad line" appunto, dove abbiamo a disposizione meno cicli macchina dei 63 previsti.

I cicli macchina si riducono, in questi casi, a soli 23 in modalità video PAL (25 in modalità NTSC).

E' possibile comunque realizzare delle routines che tengano conto anche di queste "bad lines" e che permettano quindi una corretta temporizzazione.

Un'altra tecnica consiste nell'ingannare la CPU, facendogli credere che la bad-line non sia in realtà una bad line.

Altra tecnica ancora per ottenere una temporizzazione più precisa è l'uso della cosiddetta doppia interruzione di raster.

Esistono diverse altre tecniche. Tuttavia, in virtù del fatto che ho scelto di realizzare una intro quanto più semplice possibile, che non richieda particolari conoscenze e/o studi approfonditi sull'argomento raster, ho volutamente saltato "a piè pari" tutta questa parte relativa alla temporizzazione delle barre raster.

Il risultato (modesto) è quello che potete vedere da soli, osservando con attenzione le barre raster della intro che Vi ho presentato.

Alla fine di questo articolo ho inserito alcuni link che potranno esserVi di aiuto per approfondire meglio tutti questi aspetti del raster, metodi di temporizzazione etc.

### Parte II – Inserimento di una base musicale nella intro

Qui farò abbondantemente riferimento ad un precedente articolo dove ho affrontato già la questione, articolo presente sul numero "4" di "RetroMagazine".

Riporto da quell'articolo precedente alcuni stralci.

*(...)I vantaggi nell'utilizzare un file .sid all'interno di un programma assembly sono molteplici. Grazie alla loro rigorosa codifica, diventa semplice ed immediato impiegare un certo brano piuttosto che un altro, rendendo di*



fatto il proprio lavoro di codifica separato da ciò che serve per gestire il brano musicale. All'interno di ciascun file .sid possiamo individuare tre componenti importanti, che sono: l'header, la routine di inizializzazione e quella di esecuzione del brano. L'header contiene tutta una serie di informazioni utili, tra le quali gli indirizzi di partenza delle suddette routines di inizializzazione e di esecuzione del brano, il numero di brani presenti nel file .sid (solitamente 1 ma non necessariamente), il nome dell'autore, informazioni relative al copyright del brano e molto altro ancora. Per una trattazione più precisa, esaustiva e dettagliata, rimando i lettori interessati a questo [link](https://www.hvsc.c64.org/download/C64Music/DOCUMENTS/SID_file_format.txt):

[https://www.hvsc.c64.org/download/C64Music/DOCUMENTS/SID\\_file\\_format.txt](https://www.hvsc.c64.org/download/C64Music/DOCUMENTS/SID_file_format.txt)

(...)Lo schermo video del C64 viene disegnato un certo numero di volte al secondo. E' possibile, mediante una particolare tecnica ("interruzioni di RASTER"), sfruttare l'istante in cui viene disegnata una ben precisa linea dello schermo per lanciare l'esecuzione della routine play del file .sid

Ogni qualvolta verrà ridisegnata tale linea dello schermo, verrà rieseguita anche la routine play del file .sid.

Esistono brani impostati per la frequenza PAL (frequenza 50 Hz, 625 linee di schermo, sistema Europeo), altri per la frequenza NTSC (frequenza 60 Hz, 525 linee di schermo, sistema Americano) etc.

Perchè è stato scelto questo sistema di "temporizzazione" tramite raster a discapito di altri sistemi?

La risposta è semplice. E' un sistema affidabile, estremamente preciso e già disponibile, a differenza di altri metodi che si potrebbero implementare e gestire, sfruttando oltretutto ulteriori cicli di CPU.

Si tratterà ora di integrare all'interno del codice della nostra intro il codice già presentato in quel numero "4" di "RetroMagazine".

Il file .sid incorporato all'interno della intro è stato realizzato da **Gaetano Chiummo** (<http://csdb.dk/scener/?id=22645>), che ringrazio, ed è stato messo a disposizione per essere fornito insieme al codice sorgente di questa intro sotto licenza GPL.

Come si modificherà il codice della intro? Vediamolo subito.

Intanto, bisognerà linkare il nuovo file .sid mediante apposita istruzione:

```
*=$4000 "music"
```

```
.import          binary
"retromagfin.sid",126  //// rimuove
headers non necessari
```

Poi sarà necessario, prima di disegnare opportunamente la scritta "RetroMagazine" all'interno dei due muri, impostare una interruzione di raster per far sì che ad ogni interruzione venga lanciata l'esecuzione del brano, istante per istante.

La routine eseguita ad ogni interruzione di raster è banale:

```
irq:
{
    asl $d019          //
Acknowledge raster interrupt
    jsr $4003          // address
of play-sound subroutine
    jmp $ea31          // salta
alla normale routine di servizio IRQ
}
```

Viene mandato il segnale "ack" per far sì che non venga segnalata nuovamente la stessa interruzione relativa alla riga raggiunta, poi viene lanciata la subroutine che si occupa della esecuzione del brano ed infine si prosegue con l'esecuzione della normale routine di servizio IRQ.

La parte di codice che serve ad abilitare l'interruzione di raster è la seguente:

```
    lda #$00          //
inizializzazione sound
    tax              //
    tay              //
    jsr $4000        // address
of initialization subroutine
IRQ sei              // disabilita
                    //
    lda #%01111111  //
    sta $dc0d        // Interrupt
control and status register
                    // Bit #0:
1 = Enable interrupts generated by
timer A underflow.
                    // Bit #1:
1 = Enable interrupts generated by
timer B underflow.
                    // Bit #2:
1 = Enable TOD alarm interrupt.
                    // Bit #3:
1 = Enable interrupts generated by
a byte having been received/sent via
serial shift register.
                    // Bit #4:
1 = Enable interrupts generated by
positive edge on FLAG pin.
                    // Interrupt
control and status register
                    // Bit #0:
1 = Enable non-maskable interrupts
generated by timer A underflow.
                    // Bit #1:
1 = Enable non-maskable interrupts
generated by timer B underflow.
                    // Bit #2:
1 = Enable TOD alarm non-maskable
interrupt.
                    // Bit #3:
1 = Enable non-maskable interrupts
generated by a byte having been
```

received/sent via serial shift register.

// Bit #4:  
1 = Enable non-maskable interrupts generated by positive edge on FLAG pin.

```
    lda #$01          //
    sta $d01a        //Interrupt
control register
```

// Bit #0: 1 = Raster interrupt enabled.

```
    lda #$1b          //
    ldx #$c8          // Default
value for $d016
    ldy #%00010100
    sta $d011        // Raster
line to generate interrupt at (bit
#8)
    stx $d016        // Screen
control register #2
    sty $d018        // Memory
setup register
                    // Bits #1-
#3 pointer to character memory %010,
2: $1000-$17FF, 4096-6143.
                    // Bits #4-
#7 Pointer to screen memory %0001,
1: $0400-$07FF, 1024-2047.
```

```
    lda #<irq
    ldx #>irq
    ldy #$7e
    sta $0314        // modifica il
vettore per l'esecuzione
    stx $0315        // della routine
di servizio IRQ alla locazione -irq-
    sty $d012        // Raster line
to generate interrupt at (bit #0-
#7)
```

```
    lda $dc0d        // reset per
eventuali interrupts rilevati
    lda $dd0d        // reset per
eventuali interrupts rilevati
    asl $d019        // Acknowledge
raster interrupt
                    //
    cli              // abilita IRQ
```

Il codice è stato già commentato brevemente nell'articolo già citato sul numero "4" di RetroMagazine, e sono presenti ulteriori commenti su ciascuna riga di codice.

Tuttavia riporto per Vostra comodità i commenti già presentati.

(...)Vengono inizializzati l'accumulatore, i registri X ed Y e viene richiamata la procedura di inizializzazione del file .sid. Con i tre valori o impostati, inizieremo la riproduzione del primo brano contenuto nel file .sid, spesso l'unico ma non è sempre così.

Subito dopo, mediante istruzione "SEI" vengono disabilitate le interruzioni IRQ. Vengono settati opportunamente i contenuti delle locazioni \$dcod e \$ddod (Interrupt control and status register).

Viene settato opportunamente il contenuto della locazione `$do1A` in maniera tale da abilitare le interruzioni di raster.

Per maggiore chiarezza ho espresso il valore da impostare in tali locazioni in notazione binaria, così da comprendere, bit per bit, il significato di tali valori.

Viene impostato il valore delle locazioni `$do11` (linea per esecuzione interruzione di raster), `$do16` (Screen control register #2), `$do18` (Memory setup register).

Viene modificato infine il vettore che punta alla routine di servizio IRQ reimpostandolo ad una nostra routine di servizio IRQ personalizzata.

Eliminate eventuali interruzioni in sospenso, riabilitiamo le interruzioni con apposita istruzione "CLI" ed infine usciamo dalla esecuzione del programma con l'istruzione "RTS".

Quando il pennello ottico si troverà sulla riga di scansione che è stata impostata (`$1B` ovvero la 27-esima), partirà un segnale IRQ che interromperà il normale lavoro del microprocessore, verranno salvati i valori contenuti nei suoi registri sullo stack e verrà eseguita la routine indicata con la tag "irq", ovvero la routine banale già discussa, che si occupa semplicemente di riprodurre il brano, istante per istante, ad ogni interruzione di raster.

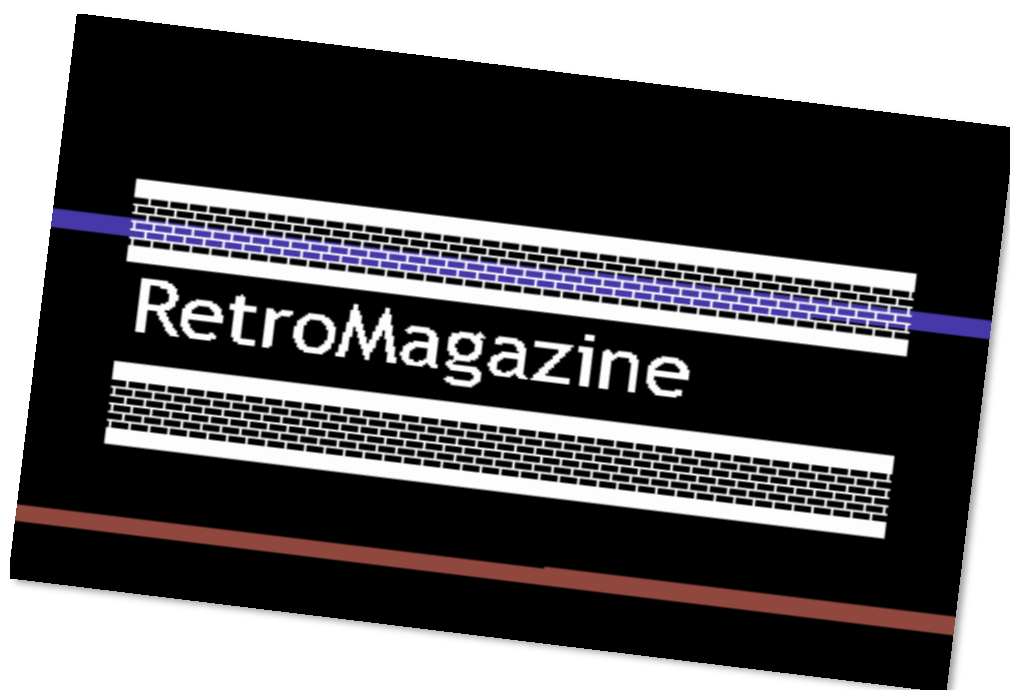
Il codice della intro, comprensivo delle modifiche illustrate, lo troverete su GITHUB al seguente indirizzo:

[https://github.com/marcus73/retromagazine\\_02](https://github.com/marcus73/retromagazine_02)

E con questa seconda "puntata" direi che è tutto.

Nella prossima introdurremo un semplice effetto di movimento dei muri, verso destra e verso sinistra, e discuteremo inoltre di alcuni particolari non ancora affrontati.

Ciao a tutti!



#### Riferimenti per approfondire:

Raster Interrupt da C64 Wiki:

[https://www.c64-wiki.com/wiki/Raster\\_interrupt](https://www.c64-wiki.com/wiki/Raster_interrupt)

Stable Raster Routine da Codebase 64:

[http://codebase64.org/doku.php?id=base:stable\\_raster\\_routine](http://codebase64.org/doku.php?id=base:stable_raster_routine)

Ancora Stable Raster Routine per C64, ma da StackOverflow:

<https://stackoverflow.com/questions/24375150/stable-raster-on-c64>

Vi rimando inoltre alla prima parte dell'articolo sempre a mia firma, Marco Pistorio, pubblicato nel numero 6 di Aprile 2018 e reperibile dal link diretto:

<http://bit.ly/RetroMagazine6>

Oppure scaricabile dal sito web di RetroMagazine:

<http://www.retromagazine.net>

## Intro for "RetroMagazine" V1.1 [2018]

User rating: awaiting 5 votes (3 left) [See votestatics](#)

Released by :  
MP Software

Release Date :  
20 April 2018

Type :  
C64 Intro

#### Credits :

Code .... MP Software  
Music .... Gaetano Chiummo of Hokuto Force, Samar Productions  
Graphics .... MP Software  
Charset .... MP Software

#### Download :

<http://csdb.dk/getinternalfile.php/166718/democsdb.prg> (downloads: 95)

#### Look for downloads on external sites:

[Pokefinder.org](http://Pokefinder.org)



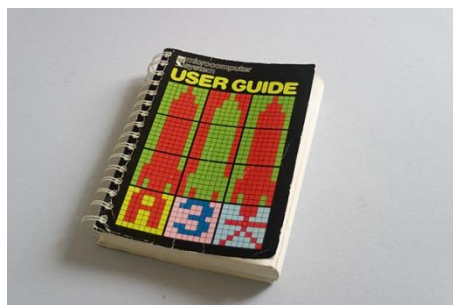
[HTTP://WWW.RETROMAGAZINE.NET](http://www.retromagazine.net)

La Intro di RetroMagazine ver. 1.1 su csdb.dk reperibile all'indirizzo: <http://csdb.dk/release/?id=164022> grazie ad MP Software!

# La migliore versione di BASIC

di Robin Jubber – *traduzione e adattamento di David La Monaca (Cercamon)*

Per moltissimi programmatori di giochi, specialmente per la vecchia generazione di questa giovane industria, la "BBC Micro User Guide", che illustra il linguaggio BBC BASIC, è stata la nostra Bibbia. Ho due o tre copie originali nella mia libreria, nel caso in cui perda la mia copia principale. Ormai tutte cadono a pezzi dopo trent'anni di onorato servizio.



*La Guida Utente del BBC Micro*

Sono un programmatore di giochi di circa 40 anni e come quasi tutti i programmatori di giochi della stessa età, la mia prima esperienza con la programmazione è iniziata con il BASIC. Ogni micro-computer a 8 bit a partire dalla metà degli anni '70 in poi ha offerto ai propri utenti una variante integrata su ROM di questo influente linguaggio, con diversi gradi di successo nelle sue molte implementazioni. Da allora ho avuto la fortuna di programmare in C e C#, la sfortuna di programmare in linguaggi come C++ e PHP e ho sperimentato la maledizione scagliata dagli dèi stessi per districarmi in quegli spettacoli dell'orrore che sono Objective C e LISP. E questi non sono neppure i peggiori linguaggi in circolazione. Alcuni dei linguaggi in cui ho programmato non esistono più, o erano esclusivi di un'azienda o anche di un singolo gioco. Altri, come il C, hanno perso un po' di popolarità nel corso degli anni, ma ancora influenzano tutti i nuovi linguaggi, ispirando una sintassi familiare a piattaforme come Java e C#.

Ma molto prima di incontrare puntatori, classi e gestione della memoria, come tutti i miei contemporanei, ho iniziato a programmare in BASIC (l'acronimo di Beginners' All-purpose Symbolic Instruction Code), un linguaggio

inventato per la prima volta più di mezzo secolo fa.

Microsoft ha dominato a lungo il mercato del BASIC fin dalla prima versione creata da Bill Gates e Paul Allen per l'Altair intorno al 1975. Questo linguaggio su ROM di 8K finì in molti modelli di micro-computer negli anni '70 e '80, tra cui Commodore, IBM, Tandy, MSX, computer CP/M e alcuni modelli di macchine a 8-bit Atari ed Apple. Nonostante la natura essenziale, la sua implementazione fu una scommessa sicura per le aziende in stile "garage" che costruivano kit di computer e anche i produttori di hardware più affermati si rivolsero a Microsoft. MS-BASIC è stato di fatto il fondamento dell'intero colosso Microsoft che oggi conosciamo.

Naturalmente non tutti i produttori di computer si sono adeguati al monopolio della casa di Redmond. Il mio primo contatto con una forma di programmazione è stato lottare strenuamente per costruire semplici programmi BASIC a scuola per il Sinclair ZX Spectrum. Il BASIC della Sinclair era una versione piuttosto povera del linguaggio, in gran parte a causa del fatto che era limitato da un sistema di input a singolo tasto dello Spectrum (n.d.T. per inserire una parola chiave del BASIC occorre infatti premere un singolo tasto o una combinazione di tasti precisa) e da un ambiente di editing tutt'altro che sofisticato. Anche gli odiosissimi tasti in gomma dello Spectrum hanno rappresentato un fattore profondamente limitante.



*Una classe primaria in UK negli anni '80*

Provate a digitare sullo Spectrum un programma usando la tastiera originale: è incredibilmente difficile. Anche altri

produttori stavano sperimentando le loro versioni del linguaggio, per esempio Atari BASIC per la serie XL, il BASIC di Wozniak per Apple II ed il Tiny Basic, che in qualche modo riuscivano a compattare in pochi kilobyte una versione funzionale del linguaggio.

Fortunatamente per me, il mio primo personal computer è stato un BBC Micro - una macchina davvero notevole per l'epoca che dominò il mercato dell'istruzione nel Regno Unito, ma che ebbe solo un successo limitato al di fuori di queste isole (UK e Irlanda). Devo supporre che mio padre abbia venduto un rene per pagare la macchina. E a me andava più che bene, visto che il mio unico interesse era quello di giocare a Elite e di sparare a qualche Thargoid.



*Il BBC Micro diffusissimo nelle scuole UK*

Il processo di alfabetizzazione informatica promosso dalla BBC nel Regno Unito è stato progettato dall'emittente radiotelevisiva pubblica britannica per presentare a bambini ed adulti il nuovissimo mondo degli home computer che si profilava all'inizio degli anni '80. Parte del progetto di alfabetizzazione prevedeva la ricerca di un produttore di computer in grado di costruire una macchina multiuso che fosse al centro dei programmi televisivi associati al progetto. Alcuni produttori, soprattutto nel Regno Unito, parteciparono alla gara per costruire la macchina che sarebbe diventata il microcomputer ufficiale della BBC. Acorn Computers, che aveva iniziato la propria attività qualche anno prima a Cambridge,



risultò il vincitore finale, in gran parte grazie alla versione potente e completa di BASIC incorporata nella macchina. La BBC voleva un linguaggio che potesse fare di tutto, che fosse ben strutturato e che permettesse agli utenti di accedere alla vasta gamma di sistemi di input e output specializzati presenti sulla macchina.

Il linguaggio fu progettato e realizzato da Sophie Wilson (all'epoca Roger Wilson) - un ingegnere informatico britannico che andò poi incontro ad una carriera senza precedenti. Sophie Wilson è infatti un nome da ricordare: è stata la metà del team che ha progettato il processore ARM RISC, inizialmente venduto come modulo coprocessore per BBC Micro. Il chip ARM può essere considerato un'eredità davvero notevole del BBC Micro, con oltre 100 miliardi di unità prodotte. Ce ne sono probabilmente due o tre proprio accanto a voi in questo momento, nascosti in ogni dispositivo elettronico concepibile che si trova sulla vostra scrivania. Uno degli obiettivi dichiarati per il chip ARM era quello di far girare il BBC BASIC alla stessa velocità con cui il computer originale a 8 bit poteva far funzionare l'assembler integrato. Non è sbagliato dire che Sophie Wilson ha raggiunto e superato (o anche abbattuto) quell'obiettivo anche meglio di quanto ci si aspettasse. L'ARM era davvero veloce. Il BBC Basic V, la versione che realizzò per la serie di computer Archimedes, fu interamente incorporato nella cache dei successivi chip ARM, garantendo che l'interprete dovesse accedere solo raramente alla RAM principale.

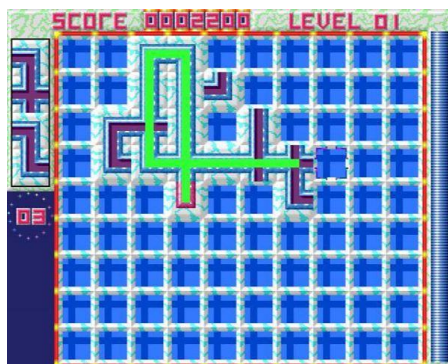
Questo naturalmente ha reso un linguaggio già molto veloce ancora più rapido ed è stato uno dei motivi per cui tante applicazioni desktop multitasking su Archimedes potevano essere facilmente scritte in BASIC invece che in C o in assembly. All'epoca, io stesso ho scritto e pubblicato un paio di giochi commerciali per Archimedes (Antigrav e GLoop) e li ho quasi interamente creati in BBC Basic V. Questi giochi non hanno di certo cambiato la storia dei videogiochi per computer, ma la possibilità di scrivere giochi a schermo intero utilizzando un linguaggio interpretato non era una cosa che si potesse fare su qualsiasi altra macchina. Sophie Wilson aveva basato la sua versione di BASIC sulla versione per Atom, una delle prime macchine Acorn, insieme ad alcune modifiche

chiave mutate da linguaggi come COMAL e Pascal.



*Sophie Wilson – designer BBC Micro e chip ARM*

Queste modifiche includevano adeguate caratteristiche di programmazione strutturata, in particolare funzioni, procedure, cicli di ripetizione, costrutti "if-then-else" ed un editor assembler integrato direttamente nel linguaggio BASIC. Queste novità erano davvero all'avanguardia per l'epoca e senza dubbio hanno assicurato ai giovani programmatori abbastanza fortunati da avere accesso ad una macchina Acorn un ottimo bagaglio pratico quando hanno iniziato i loro corsi di informatica e avevano bisogno di capire come impostare una corretta struttura di programma. Tutto ciò può essere utile per spiegare come mai il Regno Unito abbia avuto una così importante presenza nell'allora giovane settore dei videogiochi per computer, dato che i BBC Micro erano disponibili in quasi tutte le scuole del paese.



*Una schermata di GLoop per Archimedes*

Su altre macchine era impossibile scrivere programmi BASIC complessi senza usare i

comandi GOTO e GOSUB. Come sapete, si tratta di comandi che impongono salti incondizionati nel codice, che costringevano l'interprete BASIC a spostare l'esecuzione in una nuova area del programma. Questi stessi comandi rendevano però il codice poco flessibile e difficile da mantenere o addirittura da comprendere. GOTO ha un suo posto nella programmazione, specialmente nel trapping degli errori, e non rappresenta per forza la cattiva programmazione alla quale è stato tradizionalmente associato, ma è un comando di basso livello. GOTO agisce essenzialmente come un'istruzione di condizione (o di "branching", ramificazione) in assembly, che ordina al processore di passare a una nuova posizione di memoria per continuare l'esecuzione.

Il grosso problema è che solo dando uno sguardo ad una linea come `10 GOTO 90` non si capisce cosa ci si può aspettare di trovare alla linea 90, dove a sua volta si potrebbe saltare altrove nel programma con un'altra istruzione di salto GOTO. Questo inevitabilmente porta a creare programmi intricati in stile "spaghetti code" e ad una forma di programmazione che non si applica bene a progetti estesi. Con il BBC BASIC i comandi GOTO e GOSUB diventavano (e sono) di fatto opzionali. Si possono usare procedure e funzioni e scrivere direttamente `PROC_Disegna_Quadrato` oppure `FN_CalcolaRadice(10)`, il che rende il

codice molto più facile da analizzare e mantenere.



**JSBeeb – emulatore online della serie BBC**

Utilizzo ancora oggi il BBC BASIC, soprattutto quando lavoro ad uno dei miei progetti collaterali per macchine più vecchie, come il Vectrex. Di solito mantengo un emulatore BBC Micro in esecuzione sul desktop, da usare come “scratch pad” (o taccuino per gli schizzi e appunti) per semplici esperimenti di codice o per generare tabelle dati in un formato che il Vectrex sia poi in grado di integrare. Ho utilizzato il BBC BASIC anche per generare tabelle dati che poi ho utilizzato per giochi che giravano su hardware molto più potente, come ad esempio la PSP (PlayStation Portable) – quindi per me è proprio uno strumento pratico da avere a portata di mano quando serve. L'ambiente di editing è potente e facile da usare esattamente come lo ricordo fin dagli anni '80 e consiglio vivamente ai fan delle macchine a 8-bit di caricare JSBeeb (scritto da Matt Godbolt, un mio amico programmatore che ha iniziato anche lui su BBC BASIC) per avere un assaggio di com'era programmare in questo ambiente molto immediato. Non solo si potevano copiare sezioni dello schermo in nuove linee, ma si avevano anche funzionalità di tracciamento dell'esecuzione, sofisticati strumenti di rinumerazione delle linee, il già menzionato assembler che poteva essere richiamato dall'interno del BASIC (e utilizzare funzioni e costanti dichiarate nella parte BASIC) ed il controllo completo delle capacità grafiche del BBC Micro. Altre versioni di BASIC, come sapete, non avevano comandi dedicati alla grafica o al suono ed i comandi PEEK e POKE erano necessari per inserire i valori nelle singole locazioni di memoria e per interagire con le funzioni offerte dai chip audio, video e di I/O con le periferiche. È corretto dire che il BASIC sul C64 era appena utilizzabile, il che significa che a molti aspiranti giovani

programmatore dell'epoca è stato negato l'accesso diretto da BASIC alle impressionanti capacità grafiche e sonore della macchina Commodore (n.d.T. o ha spinto/costretto molti aspiranti coder in tutta Europa, Regno Unito incluso, a studiare da vicino l'assembly!).

Il BBC BASIC ha avuto una lunga storia, finendo sulle piattaforme Amstrad, Sinclair, Windows, Acorn Archimedes e Texas Instruments solo per citarne alcune. Sta ancora svolgendo dei compiti concreti a tutt'oggi, quando quasi ogni altra versione di BASIC (con l'onorevole eccezione di Visual Basic) è scomparsa da tempo. Un paio d'anni fa mio fratello, che all'epoca gestiva un negozio di dolci, aveva problemi con il computer di cui era dotato il suo punto vendita. Ho smontato il computer, ho caricato il codice che andava continuamente in crash e con mio grande stupore ho scoperto che la macchina eseguiva una versione moderna di BBC BASIC. Inutile dire che questo ha reso particolarmente facile il mio compito di risolvere il problema. E ciò significa anche che qualcuno nel mondo degli affari se la sta ancora cavando scrivendo software commerciale tramite l'uso di un linguaggio che ha imparato per la prima volta da ragazzino. Quello sconosciuto programmatore è senza dubbio il mio genere di eroe preferito.

## Riferimenti

Acorn Computers

[https://it.wikipedia.org/wiki/Acorn\\_Computers](https://it.wikipedia.org/wiki/Acorn_Computers)

Tutto sul BBC BASIC

<http://www.bbcbasic.org>

Microsoft Basic

[https://it.wikipedia.org/wiki/Microsoft\\_BASIC](https://it.wikipedia.org/wiki/Microsoft_BASIC)

Tiny Basic

[https://en.wikipedia.org/wiki/Tiny\\_BASIC](https://en.wikipedia.org/wiki/Tiny_BASIC)

Elite per BBC Micro

[https://it.wikipedia.org/wiki/Elite\\_\(videogioco\)](https://it.wikipedia.org/wiki/Elite_(videogioco))

Sophie Wilson

[https://it.wikipedia.org/wiki/Sophie\\_Wilson](https://it.wikipedia.org/wiki/Sophie_Wilson)

JSBeeb: emulatore online di macchine Acorn BBC

<https://bbc.godbolt.org>

Antigrav, uno dei giochi di Robin per Archimedes

<http://www.apdl.org.uk/apdlpd/library/files/g/g131.zip>

Fred The Needle, un gioco scritto da Robin interamente in BASIC per BBC/Electron

<https://jubberbbcmicro.webnode.com/fred-the-needle/>

La storia del chip ARM da Acorn a Apple

<https://www.telegraph.co.uk/finance/newsbysector/lepic/arm/8243162/History-of-ARM-from-Acorn-to-Apple.html>

*Elite – il gioco più famoso per BBC Micro*



# RetroMath: La matematica delle Carte di Credito

di Giuseppe Fedele



*Cosa si nasconde dietro i codici numerici presenti in una qualsiasi carta di credito?*

*E' possibile individuare un algoritmo per validare i codici identificativi durante una qualunque transazione.*

*In queste pagine prendiamo in esame uno degli algoritmi più utilizzati oggi, l'algoritmo di Luhn.*

Le carte di credito, rilasciate da enti bancari o offerte dalle grandi catene di distribuzione, sono identificate dagli stessi gruppi di cifre e da uno stesso algoritmo di calcolo e controllo, basato sull'aritmetica modulare (vedi riquadro di approfondimento). La maggior parte delle carte ha 16 cifre, il cui valore può essere uno qualsiasi fra 0 e 9, raggruppate in gruppi di 4 cifre:

ABCD EFGH IJKL MNOP

Ogni gruppo di cifre codifica un tipo di informazione: le prime quattro corrispondono all'identificativo della banca o ente che rilascia la carta. Ciascuna banca ha un numero diverso, che può variare a seconda del continente e che è in relazione al tipo di carta.

Il primo numero rappresenta la tipologia della carta di credito ovvero il circuito a cui essa appartiene:

3	American Express o Diners Club
4	Visa
5	Master Card
6	Discover Card

Le successive cinque cifre identificano l'istituto finanziario che ha emesso la carta; le altre nove cifre sono l'identificativo univoco del titolare di carta (questa parte è diversa per ogni carta ed è generata da un algoritmo); l'ultima cifra è di controllo.

Tale cifra di controllo è collegata alle precedenti attraverso l'algoritmo di Luhn, chiamato così in onore dell'ingegnere dell'IBM Hans Peter Luhn che lo sviluppò e brevettò il 6 gennaio del 1954 (U.S. Patent No. 2,950,048). L'algoritmo fu ideato come semplice metodo per distinguere numeri di carte di credito validi da quelli che contengono errori. L'algoritmo è ora di pubblico dominio e fa parte della normativa ISO/IEC 7812. Lo stesso Luhn, come descritto nel testo del brevetto, affermò che i principali obiettivi della sua ricerca erano, da una parte la progettazione di un dispositivo portatile per calcolare le cifre di validazione e per verificare una stringa di numeri che ha una cifra di controllo in coda, dall'altra un algoritmo per calcolare, in modo semplice e veloce, le cifre

di validazione da aggiungere alla stringa numerica o per verificare stringhe numeriche con cifre di controllo aggiunte.

*"The principal object of the invention is to provide a simple, inexpensive and portable computer for computing check digits and to provide a simple device for verifying numbers which have a check digit appended.*

*A further object of the invention is to provide apparatus for computing, in a fast and simple manner, check digits to append to the numbers or to verify numbers with check digits attached".*

Ed infatti tra le figure allegate al testo del brevetto troviamo anche il disegno di un dispositivo portatile su cui doveva essere implementato l'algoritmo di controllo della validità (Figura 1).

Il funzionamento dell'algoritmo per una carta con un identificativo a 16 cifre è il seguente:

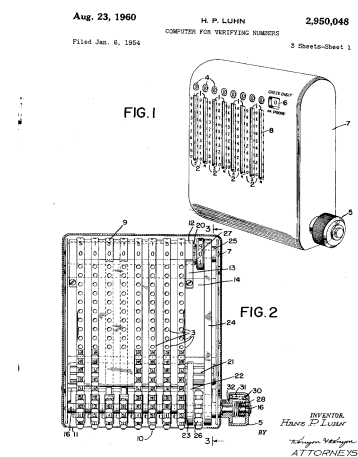


Figura 1. Dispositivo portatile brevettato da Luhn.

1. Partendo da destra e spostandosi verso sinistra si moltiplicano per 2 tutte le cifre poste in posizione pari.
2. In caso una delle moltiplicazioni porti a numeri di due cifre queste vanno sommate per ottenerne una sola.
3. Si sommano le cifre ottenute mediante le moltiplicazioni con quelle dispari. Il numero ottenuto è la cifra di Luhn.

## Aritmetica Modulare

Come calcolare 218 in modulo 15?  
Innanzitutto dividiamo 218 per 15 ottenendo: 14,5333.  
Moltiplichiamo poi 14 per 15 ottenendo 210. La quantità 218-210=8, cioè il resto della divisione di 218/15 è 218 in modulo 15 e si indica con  $218 \equiv 8 \pmod{15}$

## Bibliografia

[1] [http://it.wikipedia.org/wiki/Carta\\_di\\_credito](http://it.wikipedia.org/wiki/Carta_di_credito)

[2] M. Barnabei, F. Bonetti, *Elementi di aritmetica modulare*. Esculapio, 2014.

[3] <http://patents.google.com/patent/US2950048>.



4. Se il resto della divisione della cifra di Luhn per 10 è nullo (cioè vale 0 in modulo 10) allora il codice è valido in caso contrario no.

Si noti che la cifra di controllo è quella che fa sì che la somma totale sia un multiplo di 10.

Notiamo anche che sommare le cifre qualora il prodotto della moltiplicazione per 2 sia maggiore di 9 equivale a sottrarre 9 al risultato della moltiplicazione, infatti:

$$\begin{aligned} 2 \times 5 = 10 &\rightarrow 1 \quad (10 - 9 = 1) \\ 2 \times 6 = 12 &\rightarrow 3 \quad (12 - 9 = 3) \\ 2 \times 7 = 14 &\rightarrow 5 \quad (14 - 9 = 5) \\ 2 \times 8 = 16 &\rightarrow 7 \quad (16 - 9 = 7) \\ 2 \times 9 = 18 &\rightarrow 9 \quad (18 - 9 = 9) \end{aligned}$$

Applichiamo l'algoritmo a due potenziali codici di carte di credito (mi raccomando, evitate di fare acquisti con queste!!!)

#### Esempio 1:

1234 5678 9012 3452

$$\begin{aligned} 1 \times 2 = 2 \\ 3 \times 2 = 6 \\ 5 \times 2 = 10 &\rightarrow 1 \\ 7 \times 2 = 14 &\rightarrow 5 \\ 9 \times 2 = 18 &\rightarrow 9 \\ 1 \times 2 = 2 \\ 3 \times 2 = 6 \\ 5 \times 2 = 10 &\rightarrow 1 \end{aligned}$$

$$\begin{aligned} 2 + 4 + 6 + 8 + 0 + 2 + 4 + 2 + \\ 2 + 6 + 1 + 5 + 9 + 2 + 6 + 1 = \\ 60 \end{aligned}$$

Il risultato dell'algoritmo è quindi 60 che è multiplo di 10 ( $60/10 = 6$  con resto 0,  $60 \equiv 0 \pmod{10}$ ), quindi il codice della carta è corretto.

Vediamo adesso un codice di carta di credito errato.

#### Esempio 2:

4132 2709 5198 9687

$$\begin{aligned} 4 \times 2 = 8 \\ 3 \times 2 = 6 \\ 2 \times 2 = 4 \\ 0 \times 2 = 0 \\ 5 \times 2 = 10 &\rightarrow 1 \\ 9 \times 2 = 18 &\rightarrow 9 \\ 9 \times 2 = 18 &\rightarrow 9 \\ 8 \times 2 = 16 &\rightarrow 7 \end{aligned}$$

$$\begin{aligned} 1 + 2 + 7 + 9 + 1 + 8 + 6 + 7 + \\ 8 + 6 + 4 + 0 + 1 + 9 + 9 + 7 = \\ 85 \end{aligned}$$

Il risultato non è multiplo di 10 ( $85/10 = 8$  con resto di 5,  $85 \equiv 5 \pmod{10}$ ), quindi il codice della carta non è corretto.

#### Identificare una cifra mancante nel numero della carta

Abbiamo mostrato come dato un codice completo l'algoritmo di Luhn ci permette di concludere se questo è valido o meno. Possiamo anche affrontare però il problema inverso. Ovvero dato un codice che manca di una cifra, vogliamo risalire al codice completo. Procediamo con un esempio. Supponiamo di avere il seguente codice di una carta di credito in cui manca la cifra in settima posizione che abbiamo indicato con una X.

1234 56X8 9012 3452

Applichiamo l'algoritmo di Luhn alla sequenza. Consideriamo quindi le quantità doppie delle cifre in posizione dispari e sommiamo le cifre qualora il risultato della moltiplicazione sia maggiore di 9.

$$\begin{aligned} 1 \times 2 = 2 \\ 3 \times 2 = 6 \\ 5 \times 2 = 10 &\rightarrow 1 \\ X \times 2 = 2X \\ 9 \times 2 = 18 &\rightarrow 9 \\ 1 \times 2 = 2 \\ 3 \times 2 = 6 \\ 5 \times 2 = 10 &\rightarrow 1 \end{aligned}$$

Si noti come la cifra X sia trattata allo stesso modo.

Sommiamo adesso le cifre ottenute con le cifre in posizione pari:

$$\begin{aligned} 2 + 4 + 6 + 8 + 0 + 2 + 4 + 2 + \\ 2 + 6 + 1 + 2X + 9 + 2 + 6 + 1 = \\ 55 + 2X \end{aligned}$$

Il risultato sappiamo che dovrà essere un numero multiplo di 10.

Se la cifra X fosse compresa tra 5 e 9, avremmo che  $2X$  sarebbe compreso tra 10 e 18. Il valore di  $2X$  ridotto ad una sola cifra sarebbe allora  $2X-9$ , per cui la somma precedente sarebbe  $55 + 2X - 9 = 46 + 2X$ . L'unico valore di X, il cui risultato è multiplo di 10 è 7, infatti:

- se X fosse 5, allora  $46 + 2 \times 5 = 56$
- se X fosse 6, allora  $46 + 2 \times 6 = 58$
- se X fosse 7, allora  $46 + 2 \times 7 = 60$
- se X fosse 8, allora  $46 + 2 \times 8 = 62$
- se X fosse 9, allora  $46 + 2 \times 9 = 64$

Se al contrario, X fosse inferiore o uguale a 4, si osserva che non c'è alcun valore per cui  $55 + 2X$  è multiplo di 10, infatti:

- se X fosse 0, allora  $55 + 2 \times 0 = 55$
- se X fosse 1, allora  $55 + 2 \times 1 = 57$
- se X fosse 2, allora  $55 + 2 \times 2 = 59$
- se X fosse 3, allora  $55 + 2 \times 3 = 61$
- se X fosse 4, allora  $55 + 2 \times 4 = 63$

Di conseguenza la cifra persa è 7 ed il numero completo della carta di credito è

1234 5678 9012 3452

I due listati che trovate allegati all'articolo implementano su Commodore 64 l'algoritmo di Luhn per validare l'identificativo della carta di credito e per individuare una cifra mancante (indicata con la lettera "X"). La variabile C\$ contiene la stringa di 16 cifre con il numero della carta.

```
10 REM -----
20 REM ALGORITHM DI LUHN
30 REM CHECK ID CARTA
40 REM -----
100 REM STRINGA ID
110 C$="1234567890123452"
500 DEF FN M2(X) = X-2*INT(X/2)
510 DEF FN M10(X)=X-10*INT(X/10)
1000 C=0
1010 FOR I=1 TO 16
1020 : V=VAL(MID$(C$,I,1))
1030 : IF (FN M2(I) > 0) THEN V=2*V
1040 : IF V>9 THEN V=V-9
1050 : C=C+V
1060 NEXT I
1070 IF (FN M10(C) = 0) THEN PRINT
"ID CORRETTO!"
1080 IF (FN M10(C) > 0) THEN PRINT
"ID ERRATO!"
```

```
10 REM -----
20 REM ALGORITHM DI LUHN
30 REM CHECK CIFRA MANCANTE X
40 REM -----
100 REM STRINGA ID
110 C$="123456X890123452"
500 DEF FN M2(X) = X-2*INT(X/2)
510 DEF FN M10(X)=X-10*INT(X/10)
1000 C=0 : K=1
1010 FOR I=1 TO 16
1020 : IF (MID$(C$,I,1)="X") THEN
GOTO 2000
1030 : V=VAL(MID$(C$,I,1))
1040 : IF (FN M2(I) > 0) THEN V=2*V
1050 : IF V>9 THEN V=V-9
1060 : C=C+V
1070 NEXT I
1080 REM C+K*X DEVE ESSERE
1085 REM MULTIPLIO DI 10
1090 FOR X=0 TO 9
1100 : CC=C+K*X
1110 : IF (X>4 AND K=2) THEN CC=CC-9
1120 : IF (FN M10(CC) = 0) THEN
PRINT X
1130 NEXT X
1140 GOTO 3000
2000 REM TROVATA CIFRA MANCANTE
2010 IF (FN M2(I) > 0) THEN K=2
2020 GOTO 1070
3000 REM FINE
```

# LABEN 70, una eccellenza italiana dimenticata

di Alberto Apostolo

## Antefatto

Erano i mitici anni Ottanta quando vidi per la prima volta un LABEN 70. All'epoca ero un ragazzino che frequentava Informatica presso l'I.T.I.S. di Foligno (PG), oggi denominato I.T.T. Leonardo Da Vinci. Il computer si trovava nell'aula-laboratorio di Informatica della Scuola e mi sono sempre chiesto come avesse fatto a entrarne in possesso. Al confronto con i Commodore VIC 20 disponibili nella stessa aula, appariva come qualcosa di ingombrante e irrimediabilmente sorpassato.

L'accensione del LABEN 70 assomigliava a una specie di rito sacerdotale. Gli assistenti di laboratorio (un paio di ragazzi diplomati poco più grandi di noi studenti) accendevano e spegnevano gli interruttori sul pannello frontale del computer, per inserire i numeri in ottale che formavano le istruzioni in linguaggio macchina del caricatore bootstrap. Poi prendevano un nastro perforato di colore grigio, lungo circa un metro, e lo appoggiavano al lettore ottico affinché si potesse caricare il vero e proprio programma di bootstrap necessario per attivare l'unità a nastro, la quale alloggiava una grossa bobina contenente il Sistema Operativo M.T.O.S. e l'Interprete Basic. Con il Basic finalmente residente in memoria, si poteva digitare un piccolo programma (di circa 20-30 righe) dalla consolle, consistente in un monitor a fosfori verdi e una robusta tastiera. Il listato del programma e il suo output si stampavano tramite una piccola stampante ad aghi da 80 colonne posta accanto al monitor (sullo stesso tavolo). Il grande divertimento era salvare il programma su nastro perforato mediante il comando diretto PUNCH. La perforatrice emetteva un rumore assordante e un lungo serpente grigio fatto di carta perforata si agitava per aria come fosse stato vivo per davvero. Una volta terminata la perforazione si strappava via il nastro perforato per arrotolarlo coscienziosamente dentro una piccola scatola cilindrica di plastica trasparente, che in precedenza aveva contenuto i formaggini di una nota marca. Trascorsi tanti anni, è ritornata la curiosità di saperne di più su quel computer e ho

cominciato a cercare informazioni e manuali in Rete (trovando purtroppo poche notizie). Molto del materiale è stato trovato sul sito [www.computerhistory.it](http://www.computerhistory.it) [Ser79] con aggiunte ottenute dal cortese scambio di messaggi via e-mail con l'Ing. Giancarlo Magnaghi avvenuto nel settembre 2014 (al quale voglio porgere i miei ringraziamenti).

In Appendice si trova un esempio di programma in Basic (secondo la versione caricata sul LABEN 70 della Scuola) e la procedura per attivare il caricatore bootstrap (alcune pagine dei manuali venivano fotocopiate e date agli studenti).

## Un mini-computer a 16 (sedici!) bit

L'azienda LABEN (Laboratori Elettronici e Nucleari), faceva parte del gruppo Montedel (Montedison elettronica), a cui appartenevano anche altre aziende come IME (calcolatrici gestionali), OTE (medicali), CS Italia (circuiti stampati) e altre aziende.

La sede si trovava a Milano e aveva tre divisioni: mini-computer, strumentazione nucleare e telemetria spaziale di bordo e di terra. Nel 1968 fu intrapresa la progettazione del LABEN 70, un mini-computer per applicazioni scientifiche ed industriali, poi presentato nel 1970 (Figura 1, Figura 2). Il costo di un sistema con 4K di memoria era di 12.220\$.



Figura 1 - Il LABEN 70 e le altre unità

Già durante la fase di progettazione del computer iniziò lo sviluppo software (il sistema operativo, i linguaggi Assembler e Fortran). Per ovviare alla mancanza della macchina si utilizzò l'elaboratore Univac del Politecnico di Milano su cui era implementato

un LABEN 70 virtuale. Il LABEN 70 virtuale fu realizzato attraverso un emulatore in Fortran V che "girava" in time-sharing sull'elaboratore Univac 1108 del Politecnico di Milano. Il software di base fu progettato in parallelo all'hardware, utilizzando appunto l'emulatore.



Figura 2 - Il pannello frontale

L'area software era guidata dal prof. Galimberti e consisteva in un gruppo per i S.O. (Norma Litmaier e Agostino Belli), per i linguaggi (Giovanni Rosci) e per le applicazioni (Alessandro Osnaghi).

La macchina era "ispirata" ad alcune delle migliori macchine a 12 e 16 bit dell'epoca, come il mini Varian, l'HP 21100, l'IBM 1800 e il DEC PDP8. Infatti era una macchina concepita per il tempo reale con interrupt, ecc...

La CPU era a 16 bit con 4 accumulatori, memoria da 8 a 64 Kbyte, tecnologia MSI TTL (chip con 6 porte NAND, UART per interfacce seriali, ecc.). Le prime memorie erano ancora a nuclei magnetici ("infilati" a mano). Il sistema iniziale prevedeva il controllo attraverso una telescrivente TE300 della Olivetti. Il lettore e perforatore di banda della telescrivente era utilizzato come dispositivo di archiviazione. Al computer potevano essere collegati una stampante parallela, le unità a nastro magnetico ed il disco. Poi si aggiunsero altre periferiche supportate: telescriventi di I/O Teletype ASR 3 e Olivetti TE318, paper tape reader, paper tape punch, card reader, magnetic tape 800 e 1600 Bpi, stampanti ad aghi Centronics e line printer Data Product 300 lpm, Plotter a tamburo Calcomp e plotter analogici XY, periferiche di

processo (convertitori A/D e D/A, multiplexer digitali e analogici, registri con rele').

Le caratteristiche principali del LABEN 70 erano:

- Lunghezza di parola: 16 bit
- Memoria da 4096 a 32.768 parole (memoria a nuclei di ferrite)
- Ciclo di macchina: 1,35 microsecondi
- 8 modi di indirizzamento ed indirizzamento lungo
- Estesa gamma di istruzioni
- Istruzioni speciali per il trasferimento rapido dei dati ad interruzione
- di programma, da unita periferiche standard, convertitori analogico-digitali, ecc.

Come optional era possibile ottenere:

- Moltiplicazione e divisione cablata
- Canali di accesso diretto in memoria (fino ad un massimo di 32)
- Sistema di protezione della memoria
- Sistema di protezione per mancanza di rete
- Orologio in tempo reale

I sistemi operativi disponibili erano:

- POS (Paper Tape Operating System)
- MTOS (Mag Tape Operating System)
- SOTER (Sistema Operativo in Tempo Reale)

Il linguaggio di programmazione disponibili erano:

- Assembler
- Macro Assembler
- Basic
- Fortan IV

Il primo sistema operativo fu il POS (Paper OS) e sull'unita' di 4K parole erano disponibili l'Assembler, le librerie (matematiche, ecc.) e il Binary Loader che produceva programmi "oggetto" direttamente eseguibili. Con le unita' da 8K parole furono disponibili il MacroAssembler, il compilatore Fortran e l'interprete Basic. In seguito fu sviluppato il M.T.O.S. (Magnetic Tape O.S.) che comprendeva tutto il software di base e consentiva alle applicazioni di registrare in coda i dati di interesse. Infine fu sviluppato il D.O.S. (Disc O.S.).

#### Alcune applicazioni del LABEN 70

Le principali applicazioni erano nel campo della ricerca e del controllo di processo (acquisizione dati da gas-cromatografi, da strumentazione medica e nucleare, laboratori di ricerca di societa' farmaceutiche, controllo laminati e forni per la raffinazione dell'alluminio, ecc.). Tra i principali clienti figuravano l'Euratom di Ispra, il CNEN (ora Enea), il CNR, il Policlinico Gemelli (per elettro-encefalografia).

Il primo cliente del LABEN 70 fu il C.N.R. di Fiascherino (La Spezia) che intendeva utilizzarlo per elaborazioni marittime ed oceanografiche. Fu una installazione molto critica perche' non venne preso abbastanza in considerazione l'ambiente marino: venivano corrosi i piedini dei transistori ed altri componenti con loro distacco al contatto manuale. Il trattamento con vernice protettiva risolse il problema ma sorse quello del surriscaldamento. Alla fine venne trovata la soluzione e il LABEN 70 svolse il suo lavoro.

Il Politecnico di Milano utilizzo' il LABEN 70 per il controllo in tempo reale del SUPERSIGMA, un robot di assemblaggio sviluppato dal Politecnico sulla base del robot SIGMA di Olivetti. L'applicazione del LABEN 70 fu progettata dal Prof. Riccardo Cassinis. Inoltre il Politecnico era impegnato nello sviluppo di un sistema di compressione del segnale televisivo e a partire dal 1975 fece uso di un LABEN 70 [Fal11].

Presso l'Universita' di Pavia fu utilizzato da Virginio Cantoni per funzioni di elaborazione delle immagini.

Nei primi anni Settanta il LABEN 70 fu adottato in una ricerca per la costruzione in Europa di una rete di telecomunicazioni simile ad ARPA, la rete che generera' l'attuale Internet. L'uso del LABEN 70 fu consigliato da Luigi Dadda, che partecipava al progetto europeo.

#### Conclusioni

LABEN interruppe la propria attivita' nei computer nel 1974, nell'ambito di una ristrutturazione che determino' il taglio anche del settore dell'automazione industriale, oltre che di 160 dei 360 dipendenti.

Con lo smantellamento del ramo mini-computer di LABEN, molti tecnici (Galimberti,

Belli, Rosci e altri) andarono in Italtel dove svilupparono con altri il Leone, un elaboratore con parola da 24 bit, con un suo "doppio" per aumentarne l'affidabilita' e memory resident per risposte veloci: esso era destinato alle centrali telefoniche di transito (teleselezione e dati). Osnaghi e altri si trasferirono in Olivetti e consociate mentre Litmaier ando' all'Univerita' di Pisa (per occuparsi di Unix e altro). LABEN continuera' la sua attivita' nel settore della strumentazione nucleare e scientifica e nel settore aereo-spaziale dagli anni Ottanta fino alla fusione con Alenia Spazio (2004) dalla sua holding Finmeccanica (acquistandola) e poi integrata come sede milanese nella joint venture franco-italiana del settore spaziale Alcatel Alenia Space (67% Alcatel, 33% Finmeccanica). Divenne poi Thales Alenia Space dal 2006 con l'acquisizione della quota Alcatel da parte del gruppo francese della difesa e dell'elettronica Thales.

#### Appendice

Un esempio di programma in Basic  
La versione di Basic caricata sul LABEN 70 in dotazione all'I.T.I.S. di Foligno, non disponeva della gestione delle variabili stringa. Tuttavia aveva la gestione dell'algebra matriciale. Il seguente programma risolve il sistema lineare di 3 equazioni e 3 incognite:

$$\begin{cases} 3x - y + z = 3 \\ \quad y + z = 2 \\ x \quad - z = 0 \end{cases}$$

```
10 DIM A(3,3),B(3,1),C(3,3),X(3,1)
20 MAT READ A
30 MAT C = INV(A)
40 MAT READ B
50 MAT X=C*B
60 MAT PRINT X
70 END
81 DATA 3,-1,1
82 DATA 0,1,1
83 DATA 1,0,-1
91 DATA 3,2,0
```

#### Bibliografia

- [CC87] R. Curnow, S. Curran, Il primo libro di Informatica, Boringhieri, 1987.  
[Fal11] G. Falciasacca, Storia delle telecomunicazioni, vol. 1, Firenze University Press, 2011.  
[Ser79] L. Serrantoni, LABEN 70, [www.computerhistory.it](http://www.computerhistory.it).



# Programmazione Atari 2600 - DPC+Kernel

di Giorgio Balestrieri

## bAtari Basic e DPC+ Kernel

Continua con questo articolo il nostro viaggio nel mondo della programmazione dell'Atari 2600. Al previsto ciclo di tre articoli, per mutate esigenze editoriali ed a seguito della considerazione che per lo sviluppo di un gioco a chiusura di questa breve immersione nel bAtari Basic si utilizzerà il kernel DPC+, praticamente solo accennato nei precedenti numeri, abbiamo preferito spostare più avanti la pubblicazione dell'articolo del gioco ed inserirne uno in cui viene analizzato il DPC+, così da far acquisire al lettore una maggior familiarità con questo kernel, prima di affrontarlo in un listato complesso come quello di un software completo.

Come di consueto, anche qui seguiremo l'approccio "più pratica e meno grammatica", ovvero procederemo per esempi nell'analisi di questo kernel, utilizzando NTSC come standard video. Poiché i listati degli esempi, per quanto semplici, sono piuttosto lunghi a causa della definizione degli elementi grafici, abbiamo preferito non pubblicarli qui ma di renderli disponibili per il download, al link indicato nel paragrafo delle conclusioni.

Prima di proseguire oltre nella lettura, vi consigliamo di tenere a portata di mano i due articoli precedenti; potrebbero tornare utili come riferimento rapido per i concetti comuni di **bb** e dei kernel utilizzati e perché no, per un veloce ripasso.

## Il DPC+ kernel

Come visto nell'articolo precedente, il bAtari Basic (da qui in poi **bb**) è in grado di utilizzare diversi kernel per lo sviluppo di giochi e software, che aggiungono funzionalità avanzate al già ottimo standard kernel, permettendo al VCS di compiere dei veri miracoli, eseguendo software di una qualità nemmeno lontanamente immaginabile dai progettisti, secondo cui questa piattaforma avrebbe dovuto far girare solo semplici giochi sulla scia di quelli dell'epoca e che disegnarono il 2600 con il preciso scopo di rendere il processo costruttivo più economico possibile.

Il DPC+ è il kernel più avanzato tra quelli disponibili per **bb**, ricostruito a partire dalle funzionalità fornite dal chip DPC. Progettato da David Crane e battezzato con le sue iniziali

(David Patrick Crane), il DPC fu utilizzato per la prima (ed unica) volta nello sviluppo di "Pitfall II – The lost caverns", dove fanno bella mostra di se le migliori grafiche e l'audio a tre voci (più una batteria) ottenuti grazie all'impiego di questo chip.

Il DPC+ permette di utilizzare fino a 10 sprite giocatore ed un playfield asimmetrico e di avere sprite, playfield e background multicolore, contro i due due sprite giocatore ad un solo colore ed un playfield monocromatico e "simmetrico", ottenuto replicando nella parte destra dello schermo quanto disegnato in quella sinistra, del TIA (e dello standard kernel). A causa della sua complessità, il DPC+ funziona solo su cartucce da 32k perciò, una volta attivato, il **bb** imposta automaticamente la dimensione della cartuccia a 32k, suddivisi in 8 banchi da 4k. A disposizione per il nostro codice però avremo solo 5 banchi (20k), poiché uno viene quasi completamente utilizzato dal codice di bootstrap necessario a **bb** per inizializzare il DPC+, un altro è riservato allo stoccaggio dei dati per gli elementi grafici ed un ultimo utilizzato per il codice ARM necessario per il funzionamento sulle cartucce Harmony/Melody.

Purtroppo il DPC+ non offre solo vantaggi; il rovescio della medaglia è rappresentato dalla disponibilità di sole 35 variabili, 2398 cicli macchina per l'esecuzione del codice di gioco (contro i 2710 dello standard kernel) e nessuna possibilità pratica di sfruttare la fase di vblank per guadagnare qualche ciclo in più. Se però non ci sono necessari tutti e 10 gli sprite per il nostro gioco, possiamo ottenere qualche variabile extra forzando il kernel a ridurre il numero massimo di sprite gestiti.

## Struttura di un gioco con DPC+

Scrivere software con il supporto del DPC+ è un po' più complicato rispetto all'uso dello standard kernel. Benché resti valida la maggior parte dei concetti visti nell'articolo "Programmazione dell'Atari 2600 – bAtari Basic" pubblicato sul numero 5 di RetroMagazine, con questo kernel dovremo utilizzare dei particolari accorgimenti e tecniche per poterlo usare con successo.

Il file "playfield.bas" realizza un piccolo demo che mostra uno sfondo multicolore ed un playfield con un pattern che si estende per 176

righe, ottenuto impiegando le caratteristiche avanzate del DPC+.

Leggendo il sorgente, possiamo notare le istruzioni **bank** necessarie per selezionare uno dei 5 banchi disponibili per ospitare il nostro codice. Se non viene effettuata alcuna scelta, il compilatore mette il codice nel banco 1, in cui però è presente anche il codice di bootstrap aggiunto automaticamente per l'utilizzo del DPC+, che lo riempie quasi del tutto e non lascia molto spazio per il nostro codice. Di fatto, nel banco 1 possiamo solo "infilare" poche istruzioni di inizializzazione ed un **goto** per passare ad un banco vuoto, dove risiederà il nostro programma.

La prima istruzione del sorgente indica al compilatore di attivare il kernel DPC+, le successive dichiarano due alias per le variabili **a** e **b**, prima di passare il controllo al codice nel banco 2. Per posizionare codice in un determinato banco, è sufficiente utilizzare l'istruzione **bank** seguita dal numero del banco desiderato e scrivere codice dopo di essa. Vi preghiamo di porre grande attenzione all'istruzione alla riga 7: il **goto** presente è di estrema importanza. Dichiarare l'inizio di un banco **non** implica che il flusso sequenziale del programma proceda automaticamente da un banco all'altro, quando le istruzioni presenti nel sorgente ed appartenenti ad un certo banco sono terminate. Un banco rappresenta un'area di memoria di 4k, che il compilatore riempie con l'equivalente in codice macchina delle istruzioni **bb** dichiarate in quello spazio. Dopo di queste, a meno di non aver riempito esattamente i 4k, si trovano valori casuali che il processore tenta di eseguire come istruzioni ma che, ovviamente, non hanno logica alcuna. Per questo motivo **è necessario** che il programmatore indichi esplicitamente di saltare al codice di un altro banco, anche se l'abitudine vi suggerirebbe che l'esecuzione sequenziale delle istruzioni raggiunga lo scopo.

Notare l'inizializzazione dei banchi utilizzabili tramite l'assegnazione **temp1 = temp1**, presente in testa ad ogni banco (non è necessario che sia eseguita, basta che la "veda" il compilatore). Ciò è necessario per il corretto funzionamento dei binari generati dal compilatore con le cartucce Harmony, per cui è buona abitudine includerla in ogni programma basato su questo kernel, anche se

non si utilizzano tutti i banchi. Per passare al codice di un banco ad un altro, come già detto, è sufficiente utilizzare l'istruzione **goto**, specificando dopo la label di destinazione il banco desiderato.

Compilando ed eseguendo il sorgente, si nota immediatamente un'altra differenza rispetto allo standard kernel, ovvero la mancanza delle fastidiose righe nere ogni n scanline del playfield, che ora appare compatto e senza soluzione di continuità. In realtà, questo comportamento può essere ottenuto anche con il kernel standard, utilizzando la direttiva per il compilatore **set kernel\_options no\_blank\_lines**, che presenta però l'inconveniente di dover rinunciare allo sprite **missile0**.

Ultimo punto, non certo per importanza, su cui porre l'attenzione è l'inizializzazione delle variabili disponibili, realizzata dalle istruzioni dalle righe 18 alla 25, dove ci premuriamo anche di azzerare il volume dei canali audio della console (riga 18). In un gioco vero, dove è necessario prevedere la pressione dello switch di reset o in generale, di un meccanismo per ricominciare il gioco, questa sezione è vitale per assicurarsi che il software venga riportato alle condizioni iniziali e che le variabili, la cui visibilità è globale, contengano un valore noto ed utile ad i nostri scopi.

Nel prossimo paragrafo, dedicato alla gestione del playfield operata dal DPC+, analizzeremo il resto del codice per scoprire ulteriori caratteristiche messe a disposizione da questo ottimo kernel.

### Playfield

Le capacità di base del playfield, già espanso dallo standard kernel, vengono ulteriormente ampliate dal DPC+ che ci libera dal vincolo della simmetria ed implementa un playfield multicolore di 32 color clock per 176 righe (in NTSC). La possibilità di utilizzare più colori viene data anche al background e combinando le capacità multicolore di playfield e sfondo, possiamo ottenere dei giochi davvero coloratissimi. Il numero di colori utilizzabili vengono decisi dalle variabili di sistema **DF6FRACINC** (background) e **DF4FRACINC** (playfield) e definiti dai blocchi **pfcolors** e **bkcolors**. Contrariamente a ciò che ci suggerisce l'istinto, la scelta del numero di colori utilizzabili non è lineare, cioè non possiamo indicare ad es. valori come 1, 2, 3 e così via a **DF6FRACINC** e **DF4FRACINC** per decidere (e di conseguenza definire) di utilizzare 1, 2 o 3 colori ma siamo obbligati a seguire uno schema ben preciso, mostrato in *figura 2*. I 32 color clock del playfield sono raggruppati dal DPC+ in quattro sezioni (colonne) da 8 pixel, che presentano una

singolare particolarità, quella cioè di poter avere una "risoluzione" diversa l'una dall'altra. Attraverso le variabili riservate **DF0FRACINC...DF3FRACINC**, possiamo decidere quante righe è "alto" il playfield a video, ed è possibile variarne in corsa il valore durante l'esecuzione del programma. O almeno, questa è una buona rappresentazione di quello che accade; in realtà queste variabili decidono quante scanline debbono essere disegnate prima di passare alla prossima riga, ma possiamo ragionevolmente considerarle come la rappresentazione dell'altezza delle quattro colonne del playfield. Per indicare la composizione del playfield, possiamo utilizzare il blocco **playfield**: esattamente come nello standard kernel. Variando dinamicamente la risoluzione delle quattro sezioni, è possibile creare effetti grafici d'effetto, come quello mostrato nel sorgente "playfield.bas" dal codice presente tra le righe 317 e 324, che varia la risoluzione dopo ogni frame ed aspetta 120 quadri video dopo aver visualizzato il playfield completo prima di ricominciare il loop.

Come per gli altri kernel, anche qui abbiamo la capacità di far scorrere (scrolling) il playfield ma, a differenza degli altri, possiamo farlo solo in senso verticale. In compenso, possiamo far scrollare anche i colori del background il che torna utile, ad esempio, per creare degli interessanti effetti di parallasse.

Per una discussione più ampia su questo argomento e per ulteriori esempi, è possibile consultare l'ottima documentazione di Random Terrain all'indirizzo: [http://www.randomterrain.com/atari-2600-memories-batari-basic-commands.html#dpc\\_playfield\\_resolution](http://www.randomterrain.com/atari-2600-memories-batari-basic-commands.html#dpc_playfield_resolution)

### Sprites

Le migliorie apportate dal DPC+ si coprono anche la gestione degli sprite. Invece dei due sprite **player0** e **player1** previsti dal TIA, con il DPC+ possiamo avere fino a 10 sprite, multicolore e con una risoluzione maggiore rispetto a quelli di altri kernel, i cui pixel sono alti due scanline contro una sola degli sprite gestiti dal DPC+. La brutta notizia è che, a causa della modalità di generazione degli sprite aggiuntivi, che di fatto sono copie di **player1**, quando più sprite si trovano nella stessa fascia verticale o più precisamente, quando non c'è abbastanza spazio tra loro sull'asse delle ordinate, questi inizieranno ad esibire un fastidioso sfarfallio, eliminabile solo con un'accorta programmazione che tenda ad evitare le condizioni esposte poc'anzi.

I dieci sprite vengono utilizzati in maniera analoga a quanto avviene con lo standard

kernel, poiché il DPC+ provvede ad estendere il numero delle variabili di sistema ad essi dedicate. Ad esempio, per posizionare il terzo sprite, potremo assegnare un valore alle variabili **player2x** e **player2y** (il primo sprite è il numero zero) e così per tutte le altre variabili riservate agli sprite. Esistono alcune eccezioni a questa regola. La prima è che il registro deputato alla gestione delle proprietà degli sprite, come la dimensione ed il numero di copie aggiuntive, **NUSIZ1**, non è utilizzabile ed al suo posto deve essere usata la variabile **\_NUSIZ1** che lo mappa. Ciò è necessario poiché la gestione del registro corrispondente del TIA viene operata dal DPC+ per il multiplexing dello sprite **player1**.

La seconda eccezione, è la mancanza di variabili REFPn in aggiunta a **REFP0** e **REFP1**, per la riflessione speculare degli sprite sull'asse orizzontale. Al loro posto dobbiamo utilizzare il terzo bit di **\_NUSIZ1** e delle variabili **NUSIZ2-9**; ponendolo ad 1, lo sprite corrispondente viene riflesso specularmente e, com'è facile intuire, impostandolo a zero lo sprite riprende l'orientamento originale.

Per definire i dati degli sprite, vanno utilizzati i blocchi **player0...9** ma, a differenza dello standard kernel, i dati vanno (finalmente) inseriti nell'ordine naturale e non in quello inverso. L'assegnazione dei colori avviene tramite i blocchi **player0...9color**, dove possiamo indicare il colore di ogni riga di pixel. Nel caso in cui si voglia assegnare a più sprite lo stesso aspetto o pattern di colori, si può utilizzare la forma **playerN-M** per la definizione delle forme e **playerN-Mcolor** per i colori, dove N ed M vanno da 1 a 9.

Il file "sprite.bas" contiene un esempio d'uso dei 10 sprite in azione. Il demo mostra tutti gli sprite gestibili dal DPC+, definiti in forma e colori dalle righe dalla 61 alla 85, che percorrono lo schermo da destra a sinistra e viceversa in un loop infinito.

### Conclusioni

Anche se non prevista, speriamo che questa panoramica sul DPC+ sia stata interessante e abbia mostrato con sufficiente chiarezza le potenzialità di questo kernel. Come avevamo premesso già nella seconda puntata di questa mini serie sulla programmazione dell'Atari 2600, scrivere software per il VCS non è affatto un compito banale nemmeno con l'aiuto del bAtari Basic, ma a questo punto dovrete avere abbastanza familiarità con il linguaggio per poter sperimentare con successo lo sviluppo di giochi su questa piattaforma.

Pur mancando ancora un articolo, quello in cui presenteremo un gioco completo scritto in **bb**

(non aspettatevi miracoli però!), per chiudere ufficialmente questo ciclo, di fatto il nostro viaggio all'interno di bAtari Basic termina qui.

Ancora molto ci sarebbe da dire e scrivere sul **bB**, tanto da riempire almeno un libro, ci auguriamo però di avervi trasmesso sufficienti informazioni per muovere i primi passi con questo tool di sviluppo e di aver stimolato a sufficienza la vostra curiosità per proseguire il viaggio da soli.

#### Per approfondire:

Istruzioni per l'uso del DPC+ kernel ad opera di Random Terrain:

<http://www.randomterrain.com/atari-2600-memories-batari-basic-commands.html#dpcplus>

Download sorgenti e binari dei demo presentati in questo articolo:

[http://www.retro magazine.net/download/RetroMagazine6\\_DPC\\_plus\\_Kernel.zip](http://www.retro magazine.net/download/RetroMagazine6_DPC_plus_Kernel.zip)

Numeri 4 e 5 di RetroMagazine

- Numero 4 - <http://bit.ly/RetroMagazine4>

- Numero 5 - <http://bit.ly/RetroMagazine5>

Oppure reperibili direttamente sul sito:

[www.retro magazine.net](http://www.retro magazine.net)

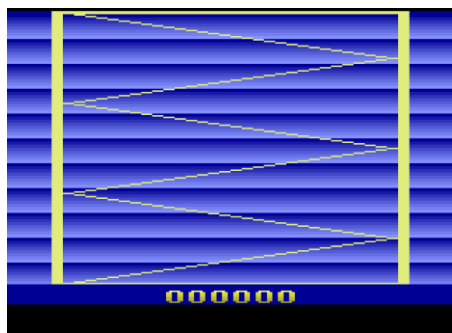


Figura 1 – DPC+ PlayField

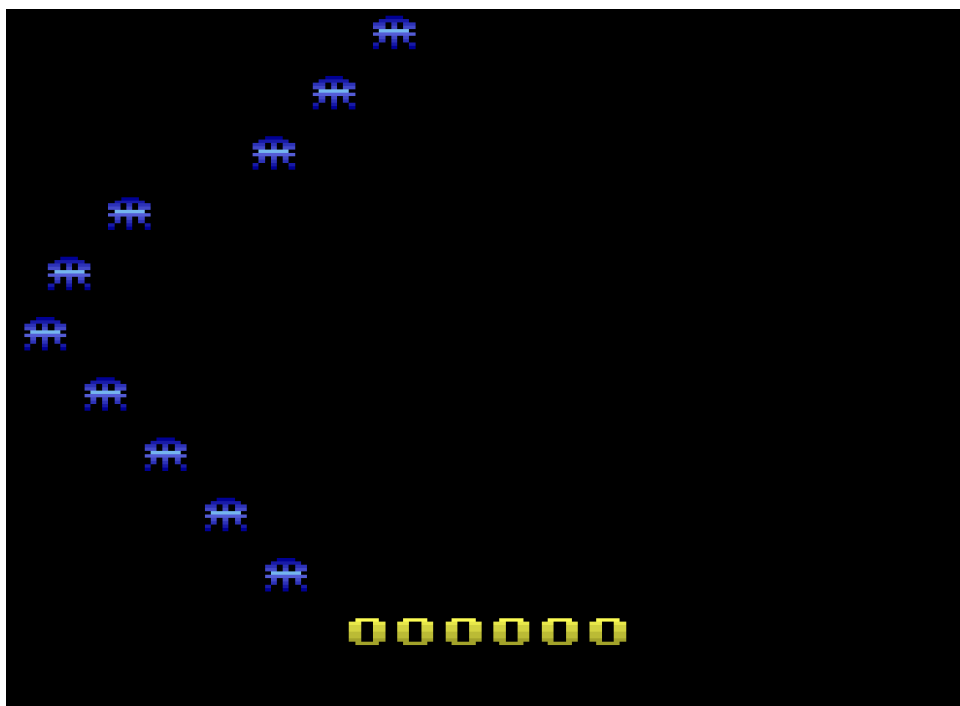


Figura 2 – DPC+ sprite demo

Scanlines	Resolution	DF(0-3)FRACINC	DF4FRACINC
1	176	255	-
2	88	128	255
3	59 *	86	-
4	44	64	128
5	36 *	52	-
6	30 *	43	86
7	26 *	37	-
8	22	32	64
9	20 *	29	-
10	18 *	26	52
11	16	24	-
12	15 *	22	44
13	14 *	20	-
14	13 *	19	38
15	12 *	18	-
16	11	16	32
18	10 *	15	30
19	10 *	14	-
20	9 *	13	26
22	8	12	24
24	8 *	11	22
26	7 *	10	20
29	7 *	9	-
32	6 *	8	16
37	5 *	7	-
44	4	6	12
52	4 *	5	10
64	3 *	4	8
86	3 *	3	6
128	2 *	2	4
176	1	1 or 0	1 or 0

Fonte: randomterrain.com

\* La riga più in basso non si sviluppa per l'intera sua altezza.

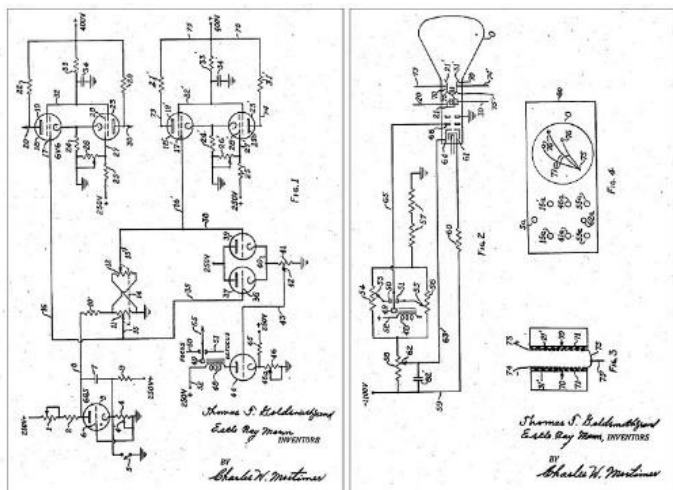
Tabella scanline/risoluzione PlayField



# Una lunga storia iniziata nel 1947

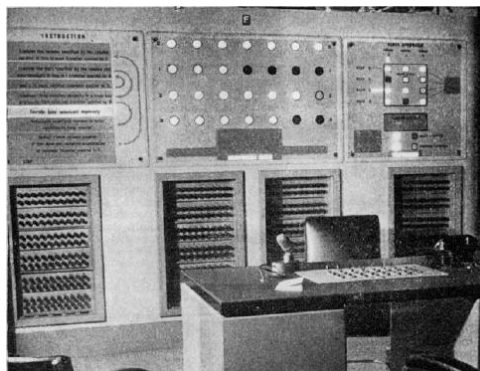
di Rita Diliddo

Per giungere a ciò che attualmente chiamiamo "videogioco" il progresso ha dovuto attraversare numerose tappe, costellate da pionieri della scienza e della tecnologia. Il lungo cammino che ha portato alla nascita del videogioco inizia nel lontano 1947 quando due ingegneri, Thomas T. Goldsmith Jr. e il suo collega Estle Ray Mann misero a punto un sistema di trasmissione di immagini su CRT, sfruttando otto valvole termoelettriche. Le immagini trasmesse consistevano in un punto, rappresentante il missile, guidato dal videogiocatore attraverso due manopole che permettevano di regolare la traiettoria e la velocità con cui il missile avrebbe colpito i bersagli, rappresentati sullo schermo attraverso dei fogli di pellicola trasparente incollati sullo schermo. Un artificio del genere al giorno d'oggi potrebbe lasciar scappare un sorriso divertito, ma bisogna tenere conto di diversi fattori, tra cui l'assenza del concetto di componentistica complessa e di software. La richiesta di registrazione dell'invenzione fu presentata presso l'Ufficio Brevetti nel Gennaio del 1947 e venne registrato nel Dicembre del 1948 col brevetto n. 2 455 992. L'invenzione di Goldsmith e Ray Mann fu solo brevettata ma mai commercializzata a causa degli alti costi di produzione.



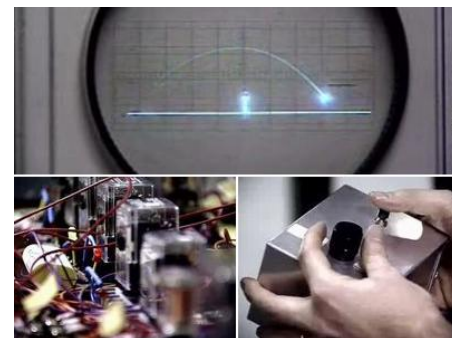
Un nuovo passo in avanti verso l'attuale concetto di videogioco venne compiuto nel 1951, quando la Ferranti International per dimostrare la potenza dei suoi computer mise a disposizione del pubblico della Fiera della Scienza di South Kensington, Nimrod, un dispositivo che riproduceva l'antico gioco di strategia Nim. Scopo del gioco è quello di

spostare dieci oggetti da una stanza all'altra, spostandone uno alla volta; vince chi riesce a spostare per primo tutti gli oggetti. Nel caso di Nimrod, gli oggetti da spostare erano rappresentati da delle lucette riportate sul computer, e queste si spegnevano quando l'oggetto veniva spostato. Seppur in maniera molto semplicistica, Nimrod rappresenta uno dei primi esempi di CPU, in quanto in tale caso il videogiocatore sfidava il pc. Nimrod non fu mai commercializzato, ma rimase a disposizione del pubblico per un periodo di tempo limitato.



La primissima forma in cui il videogioco è fruibile al grande pubblico è attraverso Pong, gioco in cui viene simulata una partita di tennis. Antenato di Pong è Tennis For Two, ideato dal fisico William Higinbotham nel 1958. Alla base della nascita del videogioco vi era l'intenzione di intrattenere i visitatori del Brookhaven National Laboratory durante le lunghe file, in modo tale che essi non si annoiassero; il sistema era dotato di un oscilloscopio che fungeva da schermo la cui visuale del campo da tennis era laterale e non dall'alto come sarà in seguito, rendendo visibile solamente un'asticella che rappresentava la rete, e una pallina. Allo schermo erano accompagnati due scatole di acciaio, i controller, dotati di una manopola per regolare l'angolazione del lancio e di un tasto azione per lanciare la palla nella metà campo avversaria. Partecipò all'assemblaggio

del dispositivo anche Robert Dvorak, collega di Higinbotham. Il computer utilizzato era un Donnel Model 30 e anche in questo caso il prodotto non venne commercializzato, anche se riscosse successo presso i visitatori della fiera.



Il videogioco assume le sue caratteristiche tipiche grazie a Spacewar!, sviluppato ed ideato nel 1961 da tre studenti del MIT, Steve Russel, Martin Graetz e Wayne Wiitanen su un computer PDP-1. Programmato per puro divertimento personale, il videogioco in questione ha come tema centrale le guerre spaziali: compito del videogiocatore è abbattere la navicella avversaria, controllata da un altro videogiocatore, mentre queste vagano nell'iperspazio, schivando eventuali oggetti che entrano nell'area di gioco. Come con Tennis For Two, le navicelle vengono guidate da un controller, dotato di una manopola per controllare l'astronave e due tasti azione, uno utilizzabile per colpire la navicella avversaria e l'altro per permettere alla navicella di muoversi nell'iperspazio. Spacewar ebbe successo negli istituti ove era installato un PDP-1, in cui venne diffuso a titolo completamente gratuito: il primo videogioco era nato non per fini lucrosi, ma per poter intrattenere i pochi appassionati di computeristica dell'epoca.



# ABC della merenda... opsss... del neofita appassionato!

## ...ovvero come arrivare ad usare un emulatore di retro computer al meglio e vivere felici! - Parte 2

di Ermanno Betori

Dopo aver presentato nello scorso numero i vari modelli, ci accingiamo ad esplorare il mondo delle periferiche di questa serie di computer, partendo dalla periferica più importante e diffusa il **Program Record**.



Per molti di noi il registratore a cassetta è stato il primo dispositivo che abbiamo avuto per salvare i programmi che avevamo digitato nella console. Rispetto a un sistema di unità disco, il registratore era molto meno costoso e anche se lento, ha fatto sempre un buon lavoro ed era abbastanza affidabile.

Basti pensare che dopo 30 anni le cassette registrate all'epoca ancora funzionano e si leggono i programmi.

Quando il TI-99/4 fu venduto nel 1979, la TI non rilasciò il proprio registratore di programmi fino all'inizio del 1983. Durante questo periodo gli utenti erano praticamente soli a trovare un lettore di cassette che avrebbe funzionato. C'erano diversi articoli nelle riviste TI del tempo che confrontavano diversi modelli e fornivano elenchi di quelli che avrebbero funzionato e quelli che non funzionavano. La migliore fortuna si poteva avere scegliendo un modello di qualità con un controllo di tono e volume. Il modello che la TI alla fine rilasciò è stato effettivamente prodotto da General Electric e la Texas Instruments semplicemente lo rinominò. I primi erano neri, e i modelli a breve color beige erano la norma. Un problema che era abbastanza comune con i registratori di terze parti era che non potevano essere accesi e

spenti usando il jack remoto. La rivista 99'er produsse ed ebbe una elevata vendita di un piccolo circuito adattatore chiamato "TI-Sette". Ciò permise al jack remoto di funzionare sui registratori sopra menzionati.

Un sacco di software è stato rilasciato su cassetta, e anche con un sistema a disco non è probabilmente una cattiva idea avere anche un registratore a portata di mano!



L'immagine sopra mostra i controlli principali. Una caratteristica molto utile è il contatore. Questo ti permetteva di mettere diversi programmi sullo stesso nastro, e annotare dove ogni programma iniziava, tale contatore divenne parte fondamentale nella scelta di un registratore anche se di altre marche.



L'immagine sopra mostra la vista laterale del registratore, notare i vari Jack in cui collegare il cavo in dotazione al computer. La figura mostra anche i controlli di tono e volume e anche il connettore CA utilizzato con il cavo CA in dotazione. Anche se potevi far funzionare questo registratore con le batterie, non era consigliabile poiché se le batterie si erano esaurite, il registratore rallentava e produceva risultati insoddisfacenti.

### Il Peripheral Expansion System



In cosa consiste questa periferica? La risposta funzionale ideata nel 1981 su come gestire tutte le schede di espansione del TI99/4A senza occupare un oceano di spazio. Infatti per chi non ricordasse, la precedente versione del TI99 completa di tutte le sue espansioni era configurata nel seguente modo:



Perché questa periferica è indispensabile? Lo è in quanto permette l'inserimento in otto slot di schede di espansione le quali completavano e rendevano efficiente al massimo il computer.



Infatti per usare al meglio il TI99, un qualsiasi acquirente degli anni 80, avrebbe dovuto comprare la seguente configurazione:

Console TI99, Speech Synthesizer, Interfaccia seriale/parallela (per collegare modem/stampante), espansione di memoria da 32K e come unità di storage il floppy disk con relativo controller.



Il tutto contornato dalle cartucce applicative USCD Pascal, Forth, Editor Assembler, Extended Basic, Mini Memory che permettevano l'accesso e l'uso di tali periferiche.

Ma per avere tutto questo insieme di accessori, l'utente dell'epoca doveva spendere molto... infatti in Italia causa il super dollaro, le spese di importazione, ricarico del venditore ecc.. fece sì, che il computer completo costava circa 4 milioni di Lire. Inoltre vi era una penuria enorme di software, non perché mancassero i programmi, ma semplicemente mancavano i punti vendita che avrebbero dovuto fare da intermediari tra i vari distributori di software americani o tedeschi che erano i maggiori creatori / distributori.

Il costo, la penuria dei programmi su supporto floppy e delle cartucce, e il fatto che i programmi erano scritti al 95% in americano, causò la convinzione presso gli utenti dell'epoca che il computer aveva scarse capacità grafiche / sonore (vedi giochi) e qualità computazionali per essere usato in ambito didattico o professionale, cosa che decretò la pochissima diffusione in Italia del PE-Box. Ad onore del vero il TI99 fu una delle macchine più fornite di software scolastico o gestionale dell'epoca ma ovviamente non in Italia.

Torniamo ora al nostro PE-Box.

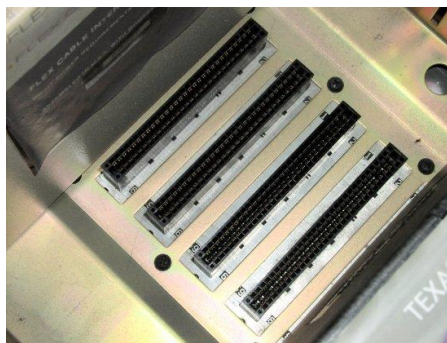


Finalmente ne siamo "felicitemente" entrati in possesso e cominciamo a scoprirne le caratteristiche..

Positive: robustezza estrema, facilità di uso, nessun problema nel inserire le schede di espansione.

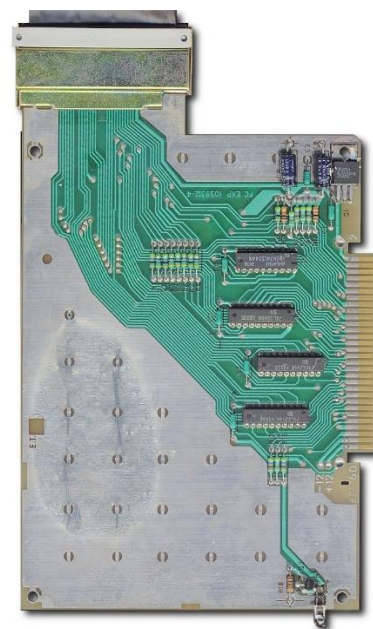
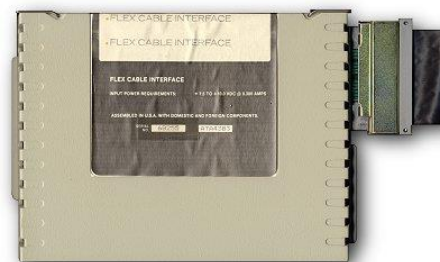
Negative: estremamente pesante, ingombrante, ventola di raffreddamento rumorosa (notare che lavora a 220V, NO 12V

come accade oggi negli alimentatori dei computer moderni).



In foto vediamo in particolare i connettori di espansione dove si inseriscono le schede. Ebbene abbiamo il primo plug and play della storia informatica, infatti come inseriamo le schede di espansione in un qualsiasi slot venivano correttamente riconosciute. Dobbiamo aspettare quasi 30 anni per avere la stessa funzione realmente operativa su computer IBM compatibili che usano il sistema operativo Windows Vista. La tecnologia Plug and Play viene portata a conoscenza del pubblico nella fine degli anni 90, tramite Win98 ma non funzionò mai al 100%, erano pochi i computer che una volta assemblati con tante schede hardware non necessitavano di interventi di correzione di driver di sistema o altro, infatti all'epoca il motto pubblicitario fu cambiato dagli informatici da plug and play in plug and pray, cioè da inserisci e gioca a inserisci e prega che funziona. La Texas Instruments aveva percorso troppo i tempi e solo una élite informatica capì fino in fondo come e con quali idee all'avanguardia la TI sviluppava i computer, ricordiamoci che il TI99 nacque come computer didattico, in teoria doveva essere un sotto prodotto costruito per fare soldi, ma gli riuscì ugualmente quasi professionale. Abbiamo parlato del PE-Box ma da solo è contenitore vuoto che veniva venduto con una scheda dalla quale usciva un connettore che si interfacciava alla consolle. Le schede di espansione venivano vendute in un pesante case di alluminio che fungevano anche da dissipatore di calore. Negli ultimi anni le produssero in una plastica speciale che doveva essere meno soggetta a interferenze radio magnetiche e naturalmente avere un costo minore. Perciò erano le schede inserite che creavano la configurazione finale del computer. Partiamo con le indispensabili:

### Flex Cable Interface Card

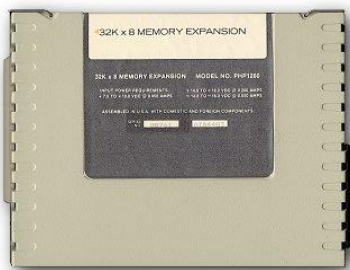


Connettore Fire-Hose

Senza questa scheda è impossibile collegare la consolle al Pe-Box, il collegamento venne chiamato Fire-Hose cioè Manichetta Antincendio in quanto lo ricordava. La foto con la parte finale della scheda chiusa in un chassid nero mostra la parte che si aggancia alla consolle del TI99/4A.

### Espansione di memoria da 32K



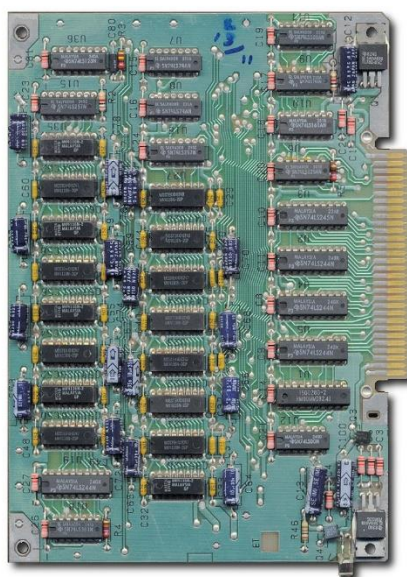


La scheda di espansione della memoria aumenta le capacità di memoria del home computer TI99/4A. La scheda aggiunge 32K byte di memoria ad accesso casuale (RAM) ai 16K byte di RAM disponibili con il computer.

Questa memoria espansa è progettata per essere utilizzata con il TI Extended BASIC, Editor / Assembler, LOGO o qualsiasi altro modulo di comando del software Solid State progettato per utilizzare la memoria aggiuntiva, nonché il sistema p UCSD.

Questo veniva descritto nella scheda tecnica dell'epoca.

Ma in realtà cosa volevano spiegare? Semplicemente che si confondevano tra la memoria del chip video (oggi chiameremmo della scheda grafica) con la memoria del microprocessore, ma come nacque questo pacchiano errore a livello di comunicazione? Fu a causa degli altissimi costi delle memorie Ram che i progettisti del TI99 (parliamo degli anni 1977-78), sfruttarono la Ram insita del processore video come Ram di sistema. Perciò l'interprete del linguaggio TI-basic che era di default, usava solamente tale Ram, e le cartucce con giochi o programmi applicativi avevano al loro interno la circuiteria necessaria per eseguire il programma. Anche il Basic esteso aveva accesso alla memoria video come memoria di sistema e vi erano pochissimi programmi che usavano l'espansione di memoria. Una microscopica Ram del processore esisteva nella consolle TI99 ma era di **256 Byte!!!!**



Il 90% degli utenti Italiani negli anni 80 del TI99 erano possessori di cartucce contenenti giochi e come altro linguaggio di programmazione comprarono solo l'extended basic. Quindi l'espansione di memoria era un oggetto sconosciuto che non si capiva l'utilizzo, dato che i programmi che funzionavano in basic o ext.basic bastava la Ram presa dal chip video. Solamente chi aveva il PE-Box completo di floppy e modulo Editor Assembler aveva scoperto l'essenzialità di questa scheda. In pratica aveva scoperto (come accadrà anni dopo su altri computer) che usando il linguaggio assembler si poteva fare la copia del contenuto delle cartucce e riversarlo su floppy o nastro e usufruire finalmente dei migliori programmi dell'epoca senza spendere una cifra esorbitante.

### Interfaccia RS232

è un adattatore di comunicazione con funzionalità seriali parallele a 8 bit che consente di collegare un'ampia gamma di dispositivi accessori al computer di casa TI. Con la scheda di interfaccia RS232 inserita nel sistema di espansione per periferiche, è possibile elencare i programmi su una stampante, inviare e ricevere dati da un terminale, scambiare i programmi TI BASIC direttamente tra computer domestici TI e molto altro. Aggiungendo l'accoppiatore telefonico (modem) e alcuni moduli di comando software a stato solido, il computer di casa può

comunicare con altri computer e terminali su normali linee telefoniche.

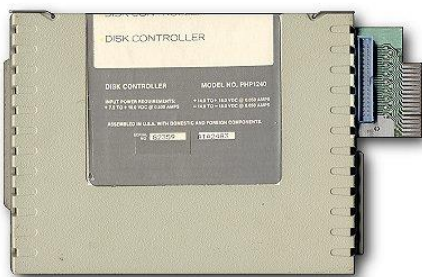
Questa descrizione trovata in una vecchissima rivista americana era quasi a livello di locandina era veritiera negli anni 80 ma solamente negli U.S.A. e per rimanere in Europa in Germania, ma in Italia vuoi i costi, vuoi la mancanza di infrastrutture (la linea telefonica esisteva solo per la fonia, NO dati parliamo sempre di utenze domestiche) era fantascienza pura!!!

Altro esempio di descrizione tecnica del periodo....

È possibile accedere a un computer dell'ufficio o ad un network da casa propria, utilizzando il TI home Computer come terminale remoto per inviare e ricevere dati. E puoi scrivere programmi in TI BASIC che utilizzano dispositivi compatibili RS232C standard EIA (Electronic Industry Association). inclusi stampanti, plotter, terminali video e altri computer. Oltre alla sua funzione di dati seriali, la scheda di interfaccia RS232 dispone anche di un I / O parallelo (Input / Output) che gestisce i dati di input e output in un formato a 8 bit. La porta I / O parallela Interfaccia direttamente con le stampanti che accettano i dati in un formato parallelo, risparmiando la spesa di avere un'interfaccia RS232 sulla stampante.



## TI Disk Controller Card & Disk Drive



Come veniva descritta una memoria di massa negli anni 80? Didatticamente si definiva un qualsiasi componente hardware che permetteva il salvataggio permanente dei dati a computer spento. Le più storiche furono la scheda perforata, la memoria a bolle, a nastro magnetico e disco magnetico. Come si definiva una unità floppy negli anni 80? Proviamo a fare una descrizione vintage...

Il sistema di memoria su disco della Texas Instruments è una potente combinazione di hardware e software che consente di archiviare e recuperare i dati in modo rapido e preciso su dischetti da 5¼ pollici. Il sistema è costituito dai seguenti componenti:

### TI DISK MEMORY DRIVE



L'unità disco legge le informazioni e scrive le informazioni sul dischetto. Può individuare rapidamente qualsiasi posizione o file sul dischetto come indicato dal controller dell'unità disco. L'unità disco gira il dischetto a una velocità costante e controlla il movimento della testina magnetica. Potresti avere una delle due unità con aspetti leggermente diversi. Tuttavia, funzionano esattamente allo stesso modo.

Nota: al momento, solo le unità disco a lato singolo (modello PHP1250 e PHP1850) sono disponibili per l'uso con il sistema di memoria su disco TI. Queste unità utilizzano solo dischetti monofaccia 5¼ pollici, a singola densità. e il modulo di comando Gestione

disco include informazioni sui dischetti a doppia faccia e a doppia densità. Questa "locandina" riporta fedelmente la tecnologia del 1981, infatti i più smaliziati noteranno che il disk drive mostrato in figura è full size cioè occupa lo spazio di 2 lettori DVD appaiati, e che usavano floppy monofaccia a singola densità alias avevano al massimo 90Kbyte come salvataggio dati. Stiamo parlando di una tecnologia fine anni 70 infatti nell'arco di pochi anni i floppy disk aumenteranno la capacità in maniera esponenziale, arrivando nel 1986-87 a valori di 720Kb, otto volte la capacità iniziale.

### TI DISK DRIVE CONTROLLER



Il disk controller originale della TI, può controllare fino a tre unità disco, comunica alle unità disco il modo cui posizionare la testina magnetica per leggere o scrivere correttamente le informazioni. Il controller mette anche un indice sul disco, rendendo facile da localizzare i dati che sono stati scritti. Questa è una descrizione classica di un qualsiasi interfaccia floppy, ma cosa aveva in più rispetto agli altri Home Computer dell'epoca? La Texas Instruments cosa fece di importante come progresso rispetto agli altri sistemi? La TI scrisse il software di controllo, (oggi chiamato firmware che è insito nel BIOS dei computer), necessario per il sistema di controllo del disco in una ROM permanente sia nella cartuccia modulo di comando Disk Manager che nella scheda drive controller, i quali potevano sia lavorare insieme che separatamente. Inoltre avevano già implementato la possibilità di usare floppy a doppia faccia per un totale di 180K che all'epoca sia il disk drive che i floppy drive che

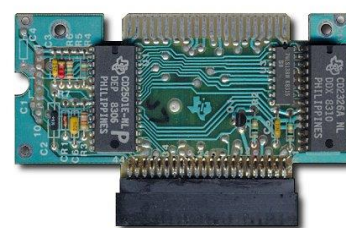
la supportavano erano costosissimi. Questo per aiutare gli utenti professionisti specialmente nell'uso di applicativi professionali tipo TI-Writer, Microsoft Multiplan, Ti Base che sono i progenitori dei programmi Word, Excel, e Access presenti oggi nel pacchetto office.

### Solid State Speech Synthesizer



Il sintetizzatore vocale è un dispositivo collegato alla porta di espansione posta sul lato destro del 99/4 o 99/4A. Il discorso prodotto dal sintetizzatore vocale è molto buono (rapportato ovviamente all'epoca!!!). Il sintetizzatore vocale contiene 373 parole e frasi diverse, ovviamente in lingua inglese e in Americano sono i fonemi della pronuncia, dunque se uno gli facesse parlare frasi in italiano la parola avrebbe la tonalità della pronuncia di uno straniero. Non eri limitato solo all'uso delle parole o frasi preimpostate, infatti utilizzando alcune cartucce speciali uno poteva creare qualsiasi parola o frase che desiderava. Un modulo è l'editor vocale (speech editor) e un altro è il Terminal Emulator II. Con uno dei due moduli sopra elencati, uno si potrebbe divertire ancora oggi a fare in modo che il sintetizzatore vocale dica qualcosa con accento Italiano!

C'erano molti moduli che hanno approfittato del sintetizzatore vocale, spesso giochi e per questo era un pezzo di hardware posseduto da molte persone.



L'immagine sopra mostra un lato del circuito stampato del sintetizzatore vocale. L'IC a sinistra è il chip di sintesi vocale TMS5220 (è contrassegnato come NL di CD2501E) e gli IC a destra sono due chip ROM saldati con il sistema piggy-back alias uno sopra l'altro.



# TheC64mini Joystick modding: the 2nd renaissance

di Michele Ugolini

Eccoci qua ragazzi, oggi vi propongo un argomento delicato, parleremo del joystick di una console veramente particolare, il neonato TheC64mini, del quale potrete leggere l'articolo "Il pomo della discordia" in RM n.6. In questo numero come promesso dalla recensione di Creatures2 vi racconto di una modifica hardware al joystick. Immagino che tutti, ahinoi, abbiamo assaporato una meccanica non troppo stabile nonchè una rigidità della manopola, soprattutto un "end-feel" cioè a fine corsa, che a volte veniva confusa con una latenza elevata. In realtà dopo alcune ore di gioco il "fine corsa" della manopola si è allentato donando soddisfazioni sufficientemente simili al glorioso e quotatissimo Tac2, un vero cimelio per i collezionisti.

Purtroppo però la stabilità di questo joy, accidenti, proprio non soddisfa nessuno neppure me. Difatti se Creatures1 era abbastanza giocabile, nel secondo capitolo è realmente difficile non perdere le vite in seguito a sbagli di timing nelle azioni o spazialità millimetriche per evitare i nemici. Immagino oltretutto, che riguardo la precisione dei controller, Creatures2 non sia neppure annoverato tra i giochi per Commodore più esigenti.



Quindi che fare? Vi propongo un modding che comporta l'apertura e la manomissione del joy, quindi si invalida immediatamente ed irreversibilmente la garanzia. Ovviamente si declina qualsiasi responsabilità per qualsiasi tipo di futuri problemi: sia che il modding riesca, sia che non riesca, sia che il risultato non sia personalmente soddisfacente.

Ricordo anche, che l'ovvia declinazione di responsabilità del mod include sia me (come autore dell'articolo), sia tutta quanta la redazione della rivista, pertanto si sconsiglia vivamente di attuarla. Per la serie... avete presente quando al cinema proiettano l'immagine che "...la pirateria è un reato

gravissimo...", ecco, ci troviamo nel medesimo modus operandi, non dovete farlo e se lo farete forse non dormirete mai più sereni!

Questo mod sicuramente non sarà di gradimento planetario, concetto avvalorato dal fatto che la soluzione della stabilità non raggiunge mai la qualità di mostri sacri di joy del passato e la garanzia viene persa. Però.. però.. se prima leggerete e rileggerete bene tutto l'articolo e vi si accenderà una lampadina oppure "il piccolo IO" della nostra infanzia sta già accennando un sorriso, beh... lo chiaramente l'ho fatto e il sonno, foorse è migliorato di un 1 x 1.000.000!

Finite le golose ed incoraggianti premesse, entriamo nell'habitat videoludico degli anni 90.

Ecco il nostro amato Commodore 64 appoggiato sul tavolino, monitor di fronte, tubo catodico irradiante i nostri teneri occhi e... elemento indispensabile di quegli anni, il tavolino di vetro, oggidi "l'economica formica" ha sostituito il glorioso e costoso "tavolo buono della sala", freddissimo in inverno, di robustissimo vetro, una pericolosissima spada non ferrosa e soprattutto raramente temperata, difficile da pulire, una calamita naturale per improntacce opache di polpastrelli sporchi e gomiti sbucciati.

Passiamo ad un altro oggetto importantissimo cioè il joystick e sotto di esso... le ventose! Ma perché sono andate in disuso? Erano così perfette allo scopo!

Infine elemento indispensabile, noi che giocavamo. Eh si, portavamo veramente all'esasperazione il joy, spesso rompendo tasti, le plastiche della leva, ma per fortuna le ventose e il "tavolo buono" erano sempre lì, sicuri e sereni!

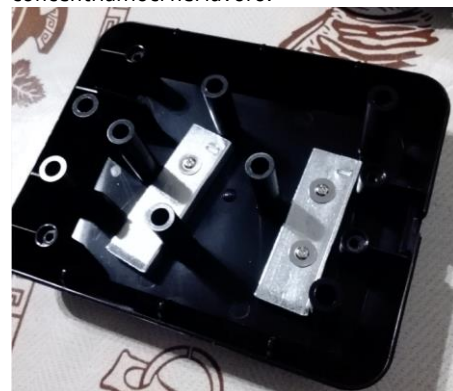
Allora le compriamo queste sante ventose? Io le ho prese da 40mm di diametro "made in Germany" con dado zigrinato, di ottima qualità, facile reperibilità e con circa dieci euro ne ho prese 12, il numero giusto per i miei 3 joystick del TheC64mini.

Dunque.. Apriamo il joystick e salutiamo la garanzia?

Servirà un cacciavite a croce tipo micro che usano gli orologiai, io ho preso un kit di questi cacciaviti a 2€, in un negozio di articoli cinesi. Svitiamo le quattro viti col cerchietto rosso. Notate che sono dinamicamente serrate di buon grado.



Aperto? Sì? Ok, addio garanzia, adesso concentriamoci nel lavoro.



Questo è l'interno, che dolore quei due pesetti.. e pensare che al tempo le ditte litigavano notti intere con i progettisti dei joy per renderli più leggeri..

Ok, togliamo i pesetti, stesso cacciavite, le viti non le useremo mai più, i pesetti li reinseriremo tra un po'.



Prendiamo delle tronchesine di precisione, dobbiamo eliminare i supporti dei pesetti dalla base del joy.



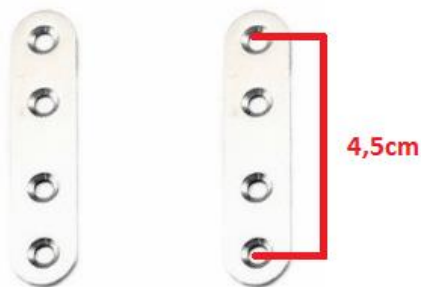
Dopo averle eliminate assicuriamoci con un coltellino di portare in piano uniformemente il tutto.





Ora cerchiamo nella cassetta degli attrezzi di casa e troviamo due comunissime piastre piane preforate, che dimensione? Quanto spazio deve esserci tra i fori? Poco importa, questi pezzetti di ferro ne vedranno delle belle! L'importante è che tra due fori della piastra ci siano 4,5cm.

Le piastre sono infinite in ferramenta, se non si trova quella giusta bisogna armarsi di seghetto per il ferro per accorciarle e trapano con una buona punta Hss per svasare il ferro per ottenere i famosi 4,5cm tra due fori e poter inserire la piastra all'interno della base del joy.



Dopo un pò di lavorazione quella piastra è riuscita ad entrare? Bene! Il meno è fatto (come disse Adriano Celentano). Inseriamo le piastre, si noterà subito un altro intoppo, non possono rimanere perfettamente adagiate alla base, ci sono due supporti di plastica del joy che danno distanza e solidità al joy stesso, non possiamo eliminarli, però possiamo eliminare parte delle piastre, allora armiamoci di pennarello e come in figura tracciamo la parte che dovrà essere eliminata.



Questa volta servirà il nostro fedele trapano con una punta da ferro Hss e un dremel (o simile) con punta per smerigliare il ferro, infine due pinze molto robuste.

Iniziamo col trapano e buchiamo in tre punti:



Poi con le pinze strappiamo il pezzo di ferro interno e con la smerigliatrice eliminiamo le parti appuntite ed affilate rimaste alla piastra.



Ecco il risultato. Adesso inseriamo la piastra per assicurarci che entri e si possa adagiare alla base interna, se ok allora passiamo a tracciare col pennarello la distanza dei fori, circa 4,5cm sulla faccia esterna della base.

Se tutto è stato fatto correttamente, questi quattro puntini dovrebbero trovarsi in diagonale di 45° rispetto l'asse dell'etichetta.



Prendiamo il dremel e con una semplice punta sottile per plastica perforiamo le zone contraddistinte dal pennarello.



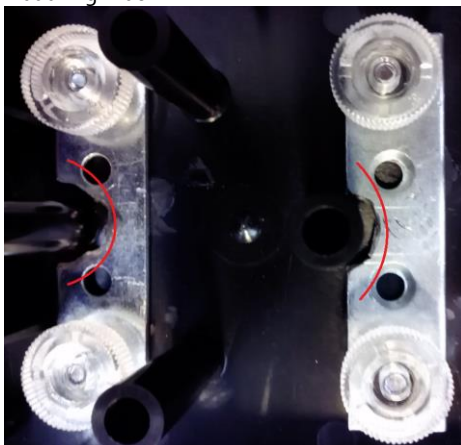
Bene, ora siamo a numerosi parsec dalla garanzia!



Ora magicamente prendiamo le ventose e le montiamo dall'esterno.



Internamente inseriamo le 2 piastre tagliate. Infine nella parte interna della base avvitiemo i dadi zigrinati.



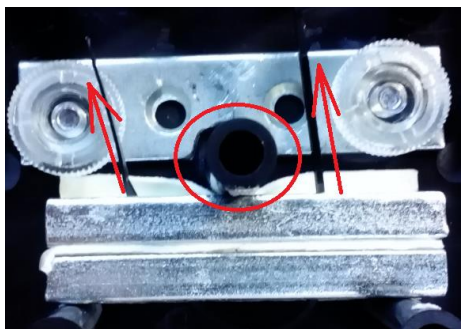
Prendiamo un poco di biadesivo, dovremo fare un sandwich (A) con i due pesetti originali.



Ora ne mettiamo un poco anche alla faccia perpendicolare (B) perché diventerà un sandwich posizionato perpendicolarmente nella base del joy con la propria base costituita da un bel biadesivo che lo tratterrà in posizione.

Ora prendiamo una fascetta e la facciamo passare attraverso i fori dei pesetti (i famosi fori che servivano al fissaggio originale della base del joy)

Inseriamo il tutto nel mezzo della base e la appiccichiamo bene alla stessa. Le terminazioni della fascetta saranno direzionate al cilindro di plastica centrale del joy.



Serriamo la fascetta al cilindro e saremo sicuri che il biadesivo alla base insieme alla fascetta, stabilizzeranno questi pesetti perfettamente.

Chiudiamo il joystick con le sue quattro viti e proviamolo!

#### Conclusioni:

Sono rimasto veramente sorpreso dalla stabilità di questa modifica, invito la casa produttrice a modificare in tal senso e con sistemi sicuramente più efficaci del mio, i futuri joystick che produrrà.

La piastra metallica che ho elaborato crea una maggior faccia d'interazione tra ventose esterne e plastica (leggera) interna della base. Le ventose sono così stabilissime e si relazionano sia internamente (dado zigrinato) che esternamente (base gommosa della ventosa) con un ottimale scarico proporzionale delle forze. Tutto sembrerebbe discretamente bilanciato.

#### Turbamenti:

Solo una cosa mi turba, nelle varie ore di gioco, le ventose sono rimaste appiccicate al tavolino di vetro per ben quattro giorni senza accusare il minimo affaticamento, il joystick non si è spostato di un millimetro, il tavolino ha resistito, il mio braccio ha lavorato ma... rimane un ma.. le famose quattro viti di

apertura della base, avete visto come sono piccole ed avvitate per così pochi millimetri alle plasticine (leggere) delle rimanenti parti del joy?

Si romperanno? La filettatura di plastica si allenterà fino a cedere e perdere l'unione tra parte sopra e base del joystick? Servirà una futura modifica in tal senso per dare più solidità alla parte sopra del joystick? Ai posteri l'ardua sentenza!



#### Fotografie sopra:

A sinistra joystick standard, a destra quello moddato. Ultima foto, il joystick modificato.

#### Esonero di responsabilita'

E' gia' stato ribadito piu' volte all'interno dell'articolo stesso ma *repetita iuvant*.

L'autore e la redazione di RetroMagazine declinano ogni responsabilita' per eventuali conseguenze dannose che possano derivare agli utenti che decideranno di effettuare la modifica illustrata nell'articolo. In pratica chi decidera' di eseguire la modifica lo fara' a proprio rischio: sia per la propria incolumita' (utilizzo di utensili) che per la funzionalita' del joystick.



# RetroTool: datatool (C64)

di Francesco Fiorentini

#1 LIST	#3 SEARCH	#5 LOAD	#7 PRINT		
#2 RUN	#4 SORT	#6 SAVE	#8 SYSTEM		
Headline		Vendite			
Comment					
Column	A	B	C	D	E
Title	Region	Vend.A	Vend.B	Totale	
Printable	yes	yes	yes	yes	yes
Type	text	number	number	number	text
Layout	left	right	right	left	left
Width	6	6	6	6	12
Variable		x	y	z	
BASIC					
001 Data	USA	836	544		
002 Data	EU	1331	980		
003 Data	APAC	1321	350		
004 Data	SA	1111	159		
005 Data	ROU	878	642		
006 Data					
007 Data					
008 Data					
009 Data					
010 Data					
011 Data					
012 Data					

Figura 1 - Foglio di calcolo – da notare la variabili x,y,z ed il codice BASIC

In questa rubrica proveremo ad analizzare alcuni software di produttività che girano sulle nostre amate macchine con lo scopo di capire se possono essere utilizzati ancora oggi come strumenti di lavoro. Nel numero 4 di RetroMagazine abbiamo parlato del GEOS un S.O. che ha rivoluzionato il modo di lavorare sul Commodore 64, introducendo tra gli altri tool un editor di testo ed un foglio di calcolo, ma non c'è sempre bisogno di scomodare il GEOS per accedere a strumenti di lavoro evoluti, almeno sul C64.

Il software che ci apprestiamo a presentarvi si chiama **datatool** ed è uno Spreadsheet per il Commodore 64.

Il tool può essere scaricato gratuitamente dal link: <http://www.webnet.at/c64/datatool.htm> dove tra l'altro è presente anche un'esaustiva pagina html che descrive tutte le funzionalità del prodotto e che guida l'utente passo dopo passo in alcune delle operazioni più complesse.

Già dalla schermata iniziale ci rendiamo immediatamente conto che ci troviamo di fronte ad un ottimo prodotto, curato nei dettagli e con un sapiente utilizzo dall'area di lavoro. Il programmatore, Franz Kottira, ha fatto un ottimo lavoro dedicando la parte superiore dello schermo al menù ad accesso rapido che consente di controllare tutte le

azioni, accessibili premendo i tasti funzione da F1 ad F8.

Per accedere all'area di lavoro basta premere **F1 LIST** ed avremo a disposizione una griglia composta da 26 colonne, identificate dalle lettere dalla A alla Z, e 999 righe numerate da 001 a 999. Ovviamente data la limitata memoria del Commodore 64 non tutte le righe e colonne potranno essere utilizzate senza espansione di memoria (a detta del programmatore 3 colonne numeriche e 999 righe potrebbero essere a disposizione sul C64 standard). Abituati ai fogli di calcolo Excel queste limitazioni ci faranno senz'altro sorridere, ma su un computer con meno di 64 KB a disposizione (tra cui far girare anche il software) sono un discreto risultato.

La prima cosa che salta all'occhio nell'area di lavoro è la possibilità di definire diversi parametri delle colonne quali il titolo della colonna, la natura del dato immesso (testo o numerico), l'allineamento, la larghezza della

#1 LIST	#3 SEARCH	#5 LOAD	#7 PRINT		
#2 RUN	#4 SORT	#6 SAVE	#8 SYSTEM		
Headline		Vendite			
Comment					
Column	A	B	C	D	E
Title	Region	Vend.A	Vend.B	Totale	
Printable	yes	yes	yes	yes	yes
Type	text	number	number	number	text
Layout	left	right	right	left	left
Width	6	6	6	6	12
Variable		x	y	z	
BASIC					
001 Data	USA	836	544	1380	
002 Data	EU	1331	980	2311	
003 Data	APAC	1321	350	1671	
004 Data	SA	1111	159	1270	
005 Data	ROU	878	642	1520	
006 Data					
007 Data					
008 Data					
009 Data					
010 Data					
011 Data					
012 Data					

Figura 2 – Calcolo del totale delle vendite eseguito tramite funzione F2.

colonna e due parametri che meritano sicuramente di essere approfonditi (Variable e BASIC).

**Variable** - è possibile associare una variabile ai valori inseriti nella colonna.

**BASIC** - è possibile eseguire operazioni utilizzando espressioni BASIC e le variabili appena menzionate.

Nell'esempio in figura 1 abbiamo definito le variabili x, y e z rispettivamente associandole alle colonne B, C e D. Nella colonna D nel tab BASIC abbiamo aggiunto una semplice somma tra i valori x ed y associandoli alla

#1 LIST	#3 SEARCH	#5 LOAD	#7 PRINT		
#2 RUN	#4 SORT	#6 SAVE	#8 SYSTEM		
Headline		Vendite			
Comment					
Column	A	B	C	D	E
Title	Region	Vend.A	Vend.B	Totale	
Printable	yes	yes	yes	yes	yes
Type	text	number	number	number	text
Layout	left	right	right	left	left
Width	6	6	6	6	12
Variable		x	y	z	
BASIC					
001 Data	APAC	1321	350	1671	
002 Data	EU	1331	980	2311	
003 Data	LATAH	1111	159	1270	
004 Data	ROU	878	642	1520	
005 Data	USA	836	544	1380	
006 Data					
007 Data					
008 Data					
009 Data					
010 Data					
011 Data					
012 Data					

Figura 3 – Il risultato dell'ordinamento F3 e della sostituzione F4.



f1 LIST	f3 SEARCH	f5 LOAD	f7 PRINT
f2 RUN	f4 SORT	f6 SAVE	f8 SYSTEM
BASIC MEMORY		6400	
ENTER BASIC			
LOAD EXTENSION		bar.extn	
RUN EXTENSION			
SET DRIVE NUMBER		8	
VIEW DIRECTORY			
SEND DISK COMMAND			
CLEAR DATA			
CLEAR ALL			
EXIT PROGRAM			

Figura 4 – Come caricare l'estensione 'bar.extn' ed eseguirla tramite il menu' F8.

variabile z ( $z=x+y$ ). Il risultato sarà quello di indicare la somma delle vendite dei prodotti nella colonna D. Non aspettatevi di vedere il valore calcolato in tempo reale come in Excel, per eseguire l'operazione occorre eseguire il comando **F2 RUN**. Dopodichè, come vedete dall'immagine in figura 2, il valore del totale delle vendite è stato correttamente calcolato dal programma.

Con la funzione **F3 SEARCH** è possibile cercare un testo nel foglio di calcolo e persino sostituirlo con altro... Nel nostro esempio sostituiamo il valore SA (South America) con il valore LATAM (LATin AMERICA). Una volta avviata l'operazione il cursore si fermerà sul valore SA di USA che però non vogliamo sostituire, basterà premere la freccia a dx per andare alla ricorrenza successiva (mentre con la freccia a sx si torna indietro) e poi premere return per sostituire SA. Una volta terminate tutte le sostituzioni premiamo F1 per tornare nuovamente al foglio di calcolo.

Premendo **F4 SORT** invece è possibile ordinare il foglio di calcolo in base ai valori inseriti in una determinata colonna (in modo ascendente o discendente). Nel nostro caso abbiamo provato ad ordinare in modo

f1 LIST	f3 SEARCH	f5 LOAD	f7 PRINT
f2 RUN	f4 SORT	f6 SAVE	f8 SYSTEM
Axis x	Column	A	
	From row	1	
	to row	199	
	Scale step	5	
Axis y	Red Column	B	
	Green Column	C	
	Blue Column	D	
	Yell. Column		
	Purp. Column		
	Grey Column		
	Red 2 Column		
	Blue2 Column		
	Scale step		
	Scale	automatic	
	Min.	0	
	Max.	2311	
	Base	0	
	Value step	462.2	
Chart	Title	Totale Vendite	
	Start	OKAY	

Figura 5 – Il menu' di configurazione del grafico a barre. Intuitivo ed immediato.

ascendente la colonna A. Anche in questo caso il tool ha risposto in maniera eccellente (vedere il risultato in figura 3).

Un discorso a parte lo merita il menù **SYSTEM** richiamabile

con il tasto funzione **F8**. Da questo menù è possibile fare diverse cose interessanti. Accedendo alla voce di menù **LOAD EXTENSION** è possibile infatti caricare le estensioni esterne del programma:

- **PIVOT.EXTN** è un'estensione che permette di creare tabelle pivot.
- **PLOT.EXTN** è un'estensione che permette di creare grafici x/y a punti.
- **BAR.EXTN** è un'estensione che permette di creare grafici a barre.
- **CHART2GEOS.EXTN** è un'estensione che permette di esportare i grafici su **GeoPaint**.

Nel nostro esempio proveremo a caricare l'estensione **BAR.EXTN** ed ad eseguirla per creare un grafico a barre con i valori inseriti nel foglio di calcolo.

Una volta digitato il nome dell'estensione 'bar.extn' e premuto **LOAD EXTENSION** non ci resta altro che eseguirla tramite **RUN EXTENSION** (figura 4) e ci troveremo di fronte ad un semplice menù di configurazione per disegnare il grafico (figura 5).

Configurare il grafico è abbastanza intuitivo ed il risultato è di tutto rispetto (figura 6).

Nella nostra prova abbiamo anche testato la funzionalità di salvataggio e recupero dei dati. Ovviamente tutto ha funzionato a meraviglia in modo semplice e veloce.

Che dire di **datatool**? E' sicuramente un ottimo prodotto, un tool che fa tutto quello che promette e forse anche qualcosa di più. Potrebbe essere utilizzato in ambito professionale? Sicuramente no, le limitazioni imposte dalla poca memoria a disposizione sul Commodore 64 ne limitano molto l'utilizzo, ma in ambito casalingo potrebbe essere usato tranquillamente per gestire un semplice bilancio familiare oppure per una piccola rubrica telefonica. Per un utilizzo didattico o per pura curiosità consiglio invece a tutti di darci un'occhiata.

Vorrei fare i miei personali complimenti al programmatore, il software interamente scritto in Assembly è veloce, intuitivo e con una GUI (Graphic User Interface) di tutto rispetto! Impressionante per essere stato realizzato da una persona sola e soltanto come un mero esercizio di stile, senza nessun fine di lucro!

#### Per chi vuole passare subito all'azione...

Il tool può essere scaricato gratuitamente dal link: <http://www.webnet.at/c64/datatool.htm> dove peraltro è disponibile un esauritivo manuale curato direttamente dal programmatore che vi aiuterà a scoprire tutte le funzionalità di **datatool**. E non sono poche. Noi ci siamo imitati alle funzioni basilari, ma il tool è incredibilmente versatile!

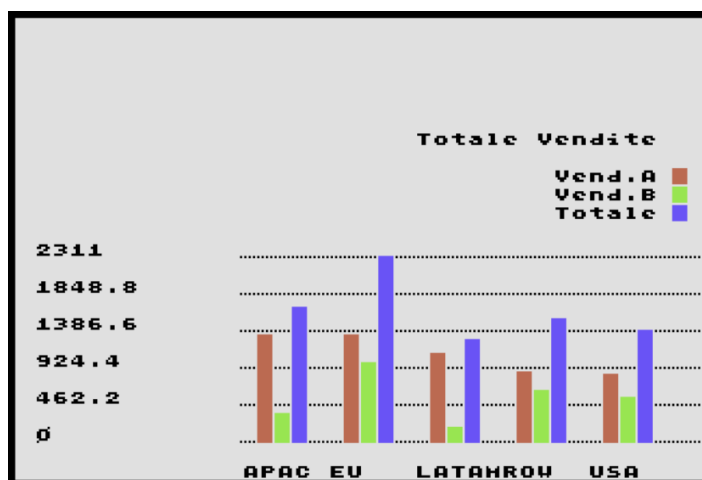


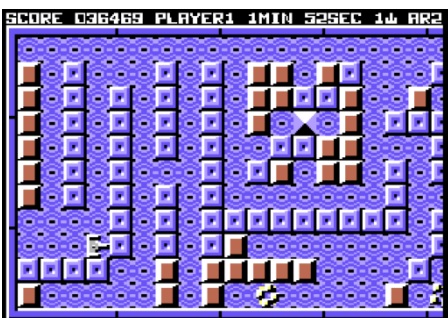
Figura 6 – Il grafico a barra in tutto il suo splendore. Excel chi?! ©

## ANARCHY



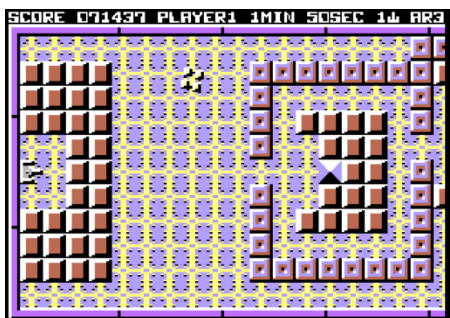
## Livello 1

Appena partiti e non e' ben chiaro cosa dobbiamo fare. Inoltre i colori dello sfondo del primo livello non aiutano molto a rendere il gioco "user friendly".



## Livello 2

La scelta cromatica del secondo livello invece e' gia' piu' azzeccata.



## Livello 3

Ormai abbiamo capito come comportarci...

## GIUDIZIO SUL GIOCO

## GIOCABILITA'

80%

La prime partite servono a capire la dinamica del gioco e come affrontarlo.

## LONGEVITA'

90%

Ha retto e regge tuttora la prova del tempo!

## Anarchy

Hewson - Anno 1987 - Piattaforma Commodore 64

*"E' scoppiata l'anarchia sul pianeta Sentinel 4, i ribelli hanno preso il sopravvento ed il mondo e' nel caos.*

*Sei stato scelto per una missione, una missione che se avra' successo fara' cadere i ribelli e riporterà la pace su Sentinel 4"...*



Questo recita l'introduzione di Anarchy, caricadonci di una pressione non indifferente per il futuro del pianeta Sentinel 4; saremo in grado di portare a termine la missione e riportare la pace laddove e' venuta meno? Scopriamolo insieme.

In Anarchy vestiamo i panni di un piccolo carrarmato, l'A.C.E. MK2 Interceptor, che infiltrandosi nel complesso dei ribelli, deve distruggere le casse di munizioni di ogni edificio prima di poter passare al successivo.

Per distruggere le casse di munizioni dobbiamo semplicemente sparargli contro; ma con una limitazione. Le casse infatti non possono essere distrutte a bruciapelo, cioe' a distanza ravvicinata, per farlo dobbiamo restare ad almeno un blocco di distanza dalla cassa per fare si' che questa possa essere distrutta dal nostro cannoncino. Ovviamente alcune delle casse sono disposte in modo tale da non rendere immediato il fatto che possano essere distrutte, costringendoci quindi a ragionare sul percorso da intraprendere per poterle disintegrare tutte nel minor tempo possibile. Eh si', perche' anche il fattore tempo e' fondamentale in Anarchy. Ogni edificio deve essere ripulito in meno di 2 minuti di tempo, pena la perdita di una vita.

Due minuti sembrano un lasso di tempo molto largo per completare ogni schermo, ma vi assicuro che il tutto e' ben calibrato. La grandezza di ogni edificio, che si sviluppa solo in orizzontale, unita all'elevato numero di casse da distruggere ed alla presenza di guardiani che al solo contatto possono distruggerci, faranno in modo di farci

letteralmente correre per riuscire a passare al livello successivo.

Per difenderci dai guardiani possiamo ovviamente sparare, ma anche in questo caso i programmatori hanno deciso di renderci la vita difficile. I guardiani una volta colpiti, invece di venire distrutti, si trasformano temporaneamente in un blocco speciale che nuovamente colpito puo' essere spostato di una posizione. Dobbiamo fare quindi attenzione a non chiuderci la strada pena la perdita di altro tempo prezioso.

Una volta distrutte tutte le casse di munizioni, si aprira' un passaggio identificato dalla scritta Exit che, sempre entro il tempo limite dei due minuti, dovremo raggiungere per passare finalmente al livello successivo.

Il passaggio si apre generalmente al centro dell'edificio e dovremo fare in modo di trovarci nei paraggi per poterlo raggiungere sani e salvi il piu' in fretta possibile. Anche perche' non ve l'ho ancora detto, ma quando il passaggio e' aperto, perderemo la facolta' di sparare e con i guardiani a caccia, l'unico modo per sfuggirgli e' cercare di evitarli.

**La chiave per completare il gioco quindi sta tutta nell'imparare il percorso migliore per distruggere tutte le casse velocemente ed al tempo stesso trovarsi nella posizione ideale quando si apre l'uscita.**

Come avrete sicuramente intuito dalla descrizione e dalle immagini allegate alla recensione, Anarchy e' sostanzialmente un puzzle game. Un puzzle game sui generis, con una buona dose di azione pero'!

Tutto sommato ci troviamo di fronte ad un buon gioco, che fu ben accolto dalla stampa nel 1987 e che ha mantenuto un certo appeal anche 30 anni dopo. I puzzle game infatti, se ben congegnati, non necessitano di chissà quale grafica per risultare divertenti e longevi ed Anarchy, pur nella sua semplicita', non sfugge a questa regola.

Se siete possessori del THEC64 Mini, potete trovare Anarchy come uno dei giochi inclusi nella 'console', se invece non lo siete, procuratevi una copia e passerete qualche ora divertendovi in compagnia dell'A.C.E. MK2 Interceptor.

di Francesco Fiorentini

## LUCKY LUKE



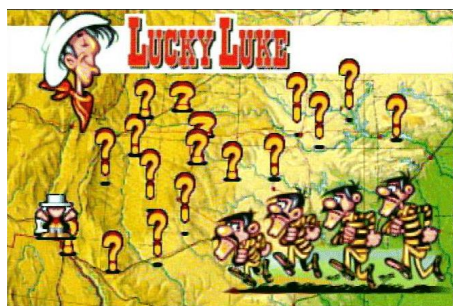
## Rantanplan

Ad aiutarci nella nostra missione ci saranno vari alleati, tra cui il fido cane Rantanplan.



## Lucky Luke

Lucky Luke è il pistolero protagonista dell'omonimo fumetto di Morris.



## I fratelli Dalton

Dobbiamo riacciuffare i fratelli Dalton, scappati di nuovo di prigione.

GIUDIZIO SUL GIOCO

GIOCABILITA'

75%

Il titolo è comunque molto tosto, soprattutto per le dimensioni generose del protagonista.

LONGEVITA'

85%

Il gioco ha 24 livelli, più sezioni segrete e stage bonus!

## Angolo Oscurita': Lucky Luke

The Vision Factory - Anno 1996 - Piattaforma CD-i

Benvenuti in *Angolo Oscurità*, una nuova rubrica dedicata a recensioni di giochi e sistemi "particolari", rimasti sconosciuti al grande pubblico.

Iniziamo con un titolo uscito esclusivamente su Philips CD-i, famigerato sistema multimediale da sempre ignorato e deriso dalla comunità degli appassionati.

Lucky Luke: The Videogame è un platform con una massiccia presenza di scontri a fuoco.

Sviluppato da The Vision Factory, un team olandese che col CD-i ci sapeva davvero fare (come dimostra *The Apprentice*), il gioco è uscito come esclusiva (e solo in Europa) nel 1996.

Lucky Luke (per chi non lo conoscesse) è il pistolero protagonista dell'omonimo fumetto di Morris (nome d'arte di Maurice de Bèvere), nato sul settimanale belga *Spirou* nel 1946. Da allora conosce un grandissimo successo, tanto da aver ispirato: film, serie televisive, cartoni animati e videogiochi. Questo non è né il primo né l'ultimo adattamento videoludico del personaggio, ma guai a sottovalutare la sua uscita su CD-i.

Nel gioco impersoneremo proprio il buon Lucky Luke, intento a riacciuffare per l'ennesima volta i fratelli Dalton scappati di prigione.

Graficamente il titolo è uno dei più belli in assoluto mai usciti su CD-i. Le capacità bidimensionali della macchina sono portate ai limiti ed il risultato è sbalorditivo. I personaggi sono enormi, disegnati ed animati molto bene; i fondali sono curatissimi, pieni di dettagli e presentano persino diversi layer di profondità. Il tutto è arricchito da uno scrolling piuttosto fluido e da occasionali effetti atmosferici come pioggia o nebbia, nonché il passaggio da giorno a notte. Ci sono anche livelli a scrolling forzato, dove dovremo farci largo a suon di proiettili in sella a Jolly Jumper o sul carrello di una miniera. Anche questi stage sono molto belli da vedere, anche se qui lo scrolling presenta rallentamenti notevoli (ma anche un sacco di personaggi ed elementi in movimento).

Il gioco richiede obbligatoriamente la scheda d'espansione, ma direi che il risultato vale ampiamente la pena.

Dal punto di vista sonoro non si possono fare rimostranze: la colonna sonora è di altissima qualità, come tutte quelle marchiate The Vision Factory (si può anche ascoltare in un lettore cd) e gli effetti sono ottimi. Anche la voce di Lucky Luke è ben campionata, anche se tende a ripetersi un po' troppo.

Il gioco pone l'enfasi su platform e sparatorie e va giocato obbligatoriamente col gamepad.

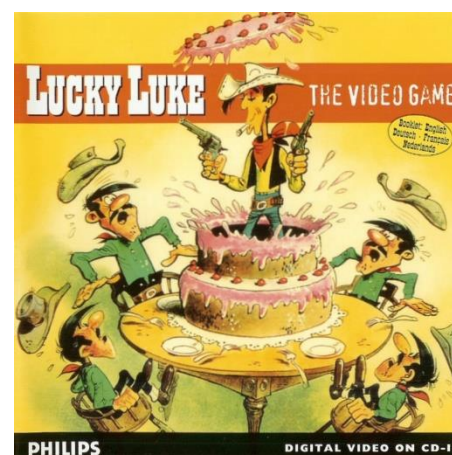
Pur risultando più facile di *The Apprentice* (Luke ha una barra d'energia) il titolo è comunque molto tosto, soprattutto per le dimensioni generose del protagonista. Ad aiutarci nella nostra missione però ci saranno vari alleati, tra cui il fido cane Rantanplan, che attaccheranno i nemici al nostro posto. Oltre a questo troveremo svariate armi sempre più potenti e la dinamite, che ci aiuterà ad uscire dalle situazioni peggiori.

Il gioco ha 24 livelli, più sezioni segrete e stage bonus. Ogni tre livelli i nostri progressi saranno salvati nella memoria del CD-i, garantendo una sfida equa ed una discreta longevità.

In definitiva Lucky Luke è un buon platform ed una grande prova di programmazione. Il fatto che sia un'esclusiva lo rende ancora più desiderabile. Consigliato a tutti gli amanti del genere e del personaggio creato da Morris.

Questa prima recensione finisce qui e, sperando vi sia piaciuta, vi do appuntamento ai prossimi numeri.

di Federico Gori





## AMOS NEWTON - MENTE ALIENA



## L'avventura ha inizio.

Una semplice schermata a caratteri caratterizza le avventure scritte da BdB.

## GIUDIZIO SUL GIOCO

## GIOCABILITA'

83%

La trama, la buona qualità del parser e dei testi, sempre chiari nel fornire le indicazioni sull'ambiente al giocatore ed il ritmo mantenuto sul "vivace andante" sono senz'altro i punti di forza di questo gioco e fanno guadagnare punti alla giocabilità. L'eccessiva facilità con cui si incorre nel game over e alcune forzature nel costringere il giocatore a mosse francamente evitabili, gliene fanno perdere un po', ma resta comunque un bel titolo molto godibile.

## LONGEVITA'

50%

Come quasi tutte le avventure, difficilmente lo rigiocherete dopo averlo finito, se non per nostalgia anni dopo. Assegniamo un punteggio insolitamente alto per questo tipo di giochi solo perché finire la prima parte di "Mente Alienata" vi spingerà a giocare anche la seconda, ma in effetti in questo caso non si può parlare di longevità in senso stretto.



## Amos Newton - Mente Alienata, parte uno

Brainstorm Enterprise - Anno 1986 - Piattaforma Commodore 64

Negli anni '80, nel periodo d'oro delle avventure testuali, un nostro connazionale, Bonaventura Di Bello, universalmente noto come BdB, in giovane età fondò insieme alla moglie (ed al fratello) una società per la produzione e distribuzione di giochi d'avventura, seguendo senza saperlo le orme di Scott Adams (v. RetroMagazine n. 2). Durante la sua carriera, analoga anche in questo a quella di Adams, Bonaventura produsse una quantità notevole di giochi, che ancora oggi vengono ricordati e giocati da chi li scoprì da ragazzo in quegli anni. Continuando l'incredibile successione di coincidenze con la vita di Adams, anche Bonaventura chiuse l'azienda quando il trend delle avventure testuali cessò e si separò da sua moglie.

Chi ha letto il numero precedente di questa rivista sa che BdB è tornato in pista ed ha scritto un nuovo gioco a quattro mani con Marco Vallarino, di cui recensiremo un'avventura in un prossimo numero. La notizia ha dato nuova linfa ad una passione mai sopita nei redattori di RM amanti del genere, in chi vi scrive in primis, il cui epilogo è l'articolo che state per leggere.

## Amos Newton - Mente Alienata, parte uno.

Pubblicato per Commodore 64 e distribuito tramite edicola nella collana "Explorer" nel 1986, l'avventura di stampo fantascientifico "Mente Alienata" vi mette nei panni di un fuggitivo perso su un pianeta sconosciuto, inseguito da soldati di cui non viene rivelata l'origine e la razza, che tenta una fuga disperata per tornare al proprio pianeta natio. Prodotto utilizzando in coppia i toolkit "The Quill" e "The Illustrator" per la produzione di immagini statiche, in "Mente Alienata" si apprezza subito la qualità del parser, in grado di capire non solo i semplici comandi formati da verbo + complemento ma anche frasi complesse, con tanto di articoli ed azioni multiple. Ad esempio, digitando "PRENDI IL MARTELLO ED IL QUADRO", il parser non avrà nessuna difficoltà a comprendere le nostre intenzioni e metterà nell'inventario il martello ed il quadro presenti nella locazione. Non in questo gioco però, dove non esistono né martelli né quadri. L'effetto "caccia alla parola" che caratterizza questo tipo di opere, tristemente noto nelle avventure di bassa qualità, qui è davvero poco sentito e difficilmente vi bloccherete perché il gioco rifiuterà di eseguire un vostro comando fino a che non gli fornirete ESATTAMENTE l'unica parola che riconosce per completare un'azione. Questo comportamento, purtroppo anche troppo diffuso, porta la giocabilità a crollare vertiginosamente e spinge l'avventuriero di turno a lanciare il tutto dalla finestra... ma come detto, "Mente Alienata" è sufficientemente al riparo da questo rischio.

Nella migliore tradizione dei giochi di BdB, in questa avventura ci ritroveremo con un buon numero di locazioni da esplorare, di cui molte assolutamente ininfluenti ai fini del completamento del gioco ma

che contribuiscono egregiamente nel creare un'atmosfera immersiva ed ad aumentare la godibilità del titolo.

L'avventura è costellata di trappole e tranelli che spesso e volentieri ci condurranno a vivere l'esperienza di una morte prematura e richiederanno un buon sollecito delle cellule grigie e una discreta apertura verso il pensiero laterale per superarli. A differenza di quanto avviene nella grande maggioranza dei titoli odierni, dove il percorso è invariabilmente guidato con il focus spostato sulla narrazione piuttosto che sull'iniziativa del giocatore, qui ogni progresso va guadagnato facendo lavorare la mente e nonostante la limitatezza del mondo di gioco (rispetto a quelli modellati oggi), potrebbero esservi necessarie un buon numero di ore per portarlo a termine.

Oltre ai puzzle "logici", cioè risolvibili tramite una mediamente intesa attività elucubratoria, ve ne sono alcuni dove la soluzione è completamente affidata al processo "a tentoni", in cui non è possibile applicare alcun ragionamento e si deve procedere per tentativi (ad esempio, scegliere tra due pulsanti quale premere). Naturalmente, un tentativo errato provocherà quasi certamente la morte del protagonista ma tant'è, BdB è sempre stato piuttosto cattivello in questo.

Tenendo presente anche la tensione provocata dal fatto di non avere troppe mosse da sprecare a causa del costante inseguimento da parte dei soldati avversari, avrete chiaro il quadro del ritmo a cui sarete sottoposti. In pratica, vi sentirete infilati in una pentola a pressione costantemente sul filo della cottura...

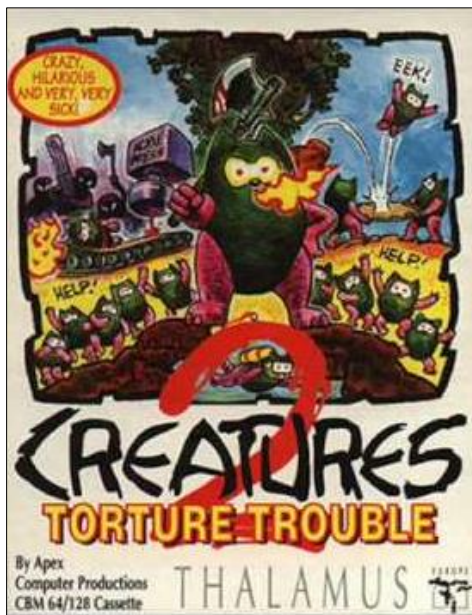
## Conclusioni

"Mente Alienata" era un buon gioco all'epoca come lo è oggi. La predilezione di BdB per la fantascienza traspare chiaramente in questo titolo, al punto che sembrerà di ritrovarsi dinanzi un racconto della collana "Urania", con la differenza che qui si è coinvolti in prima persona. Seppur non sia paragonabile a titoli maggiori, come quelli della Infocom o della Beam Software (quella di "The Hobbit"), è un dignitosissimo rappresentante del genere, impegnativo e divertente da giocare, con tocchi di humor leggero qui e lì e scritto in uno stile che contribuisce molto a creare un legame tra il giocatore ed il gioco.

Ultimo punto a favore, di certo non per importanza, l'autore ha gentilmente concesso a RetroMagazine di ospitare e distribuire l'immagine nastro del programma a beneficio dei lettori. Chi volesse giocarlo, può usare un emulatore di Commodore 64 come il Vice ed il file recuperabile all'indirizzo: <http://www.retromagazine.net/download/menteealiena1.zip>

di Giorgio Balestrieri

## CREATURES 2



## Acme+BugMatic: pessimo connubio

Dopo questi due capitoli pubblicati dalla Thalamus i fratelli Rowlands hanno creato Mayhem in Monsterland solo per C64 nel 1993.

## GIUDIZIO SUL GIOCO

## GIOCABILITA'

95%

Le stanze delle torture sono di più e molto più cruento.

## LONGEVITA'

97%

E' molto gratificante sapere che il secondo capitolo di un gioco riscuote punteggi più alti del primo, è poco gratificante perdere dei Fuzzy per via di atroci torture! Ovviamente la fusione di questi due concetti ne ha decretato il successo

## Creatures 2

Thalamus - Anno 1992 - Piattaforma: Commodore 64

Oh no, ancora?! Aiuto! Un demonio di mostriciattolo sta cuocendo mio figlio sul girarrosto. Che fare? Se indugio troppo a lungo, verrà cotto, se rallento un demone, altri demoni uccideranno i miei amici, se studio bene gli elementi di questo enigma, uhhh, pensiamo, forse forse ...

Cari amici, è giunto il momento di recensire il seguito di Creatures, parlerò di Creatures 2 cioè Torture trouble 2, un videogioco che sulla prestigiosa rivista "Zzap!" ha ricevuto il privilegio di essere titolato con un memorabile punteggio di 97%, non male per essere il secondo episodio di un videogioco.

La cassetta era venduta a 10.99€ mentre il dischetto a 15.99€, stiamo parlando del 1992, in questa data di uscita, erano cifre importanti.

Ovviamente la distribuzione era sempre in mano alla, ormai diventata potentissima, Thalamus Europe.

Questa volta la piattaforma di lavoro, vista la grandiosa giocabilità del primo capitolo, è rimasta fedele unicamente al Commodore 64. Sviluppato sempre dalla Apex Computer Productions, la famosa Apex dei geniali fratelli Rowlands.

Anche questa volta, squadra che vince non deve cambiare nulla, la formazione di gioco permane con le mansioni di programmazione in mano a John, mentre musica e grafica saranno trattate da Steve. La musica è realmente accattivante e degna di essere elevata tra le migliori Sid della storia del C64. Per coloro che non avessero già letto la recensione del primo capitolo, vi invito fortemente sia alla lettura (RM n.6) che all'immersione ludica di questo bellissimo videogioco, per due motivi, il primo sicuramente è di godere di un capolavoro che ha spremuto (a fondo!) i chip del C64 e il secondo è che dovrete salvare numerosi amici spremuti (e non solo!) da infiniti demoni.

La storia di Clyde si era conclusa con la salvezza dei Fuzzy, la decimazione dei demoni ed esilio dei mostri sopravvissuti nelle isolette vicine, infine il protagonista si era sposato con una giovane "Fuzzetta" e con lei aveva avuto tanti adorabili Fuzzy, ben nove. Nove, un numero non casuale perché purtroppo i demoni rapiscono i figli di Clyde. Verranno portati nelle tre isole, quindi tre figli per ciascuna isola, per essere torturati ed uccisi, a meno che il nostro eroe raggiunga le isole in tempo per salvarli tutti quanti.

Anche qui ci saranno enigmi da risolvere entro un certo tempo, inoltre ci saranno nuove stanze delle torture. Anche questo platform a scorrimento orizzontale abbraccia la logica

dei puzzle composti da diabolici enigmi, immersi in un countdown angosciante.

Creatures 1 aveva introdotto una logica psicologica riguardo le torture di questi adorabili animaletti alieni.

Ne aveva fatto il proprio cavallo di battaglia per assicurare più interesse negli acquirenti del secondo capitolo, quindi...

In Creatures 2 le torture saranno più sanguinose, le tipologie di morte dei Fuzzy saranno molto più colorite e tragiche rispetto al primo capitolo.

Non solo, in questo capitolo si aggiunge un altro stadio psicologico, la decisione di: o salvare d'acchito gli amici, oppure modificare i movimenti dei demoni che portano alla morte dei Fuzzy. Questo stress evocerà numerosi dilemmi nel giocatore, in maniera molto simile ai Lemmings: un Lemming scoppia, infiniti Lemmings vivono...

Sempre parlando di cinismo dei fratelli Rowlands, l'interazione tra i demoni nello schermo delle torture assume valori e animazioni più complessi, gli oggetti utilizzati per interrompere le torture sono spesso impiegati in maniera originale, non conforme a ciò che a colpo d'occhio potrebbe essere adoperato in maniera logica.

Che dire, si potrebbe parlare di dilemma, black humor, ma in questo caso la rivista Commodore Format lo ha definito con "wild humor", come dare torto al recensore?

Questa volta ci saranno numerose inversioni di tendenza, siccome nel primo capitolo i nemici di Clyde ridevano sempre, mentre i Fuzzy avevano ben poco da ridere, questa volta ci saranno rapide variazioni di umore per tutti quanti.

Le stanze delle torture saranno sei e tra di esse saranno aggiunti schermi dove potremo guadagnare delle vite.

La composizione delle schermate è abbastanza complessa:

in sintesi Clyde deve superare un percorso composto da 3 isole, divise ciascuna da 6 parti distinte: 2 stanze delle torture, 2 intermezzi per far passare a furor di balzi e rimbaldi i Fuzzy tra le isole, 1 difficile scontro con un demone maggiore, infine 1 schermo dove dovrà trasportare i figli tra le isole.

Anche gli sprite saranno molto più ricchi, per esempio vi ricordate il divertentissimo Game&Watch dove gli omini che si lanciavano nel vuoto venivano salvati da due barellieri dotati di portantina e, di rimbaldi in rimbaldi, dalla barella gli omini atterravano nell'ambulanza?

Ecco qua che i Fuzzy saranno salvati allo stesso modo, ma no tranquilli, non è un ovvio



## APPROFONDIMENTO



In quegli anni un videogioco con scene di tortura così ben miscelate da orrore, humour, sadismo, cinismo, divertimento, hanno scatenato diversi vignettisti, allego un link di una irriverente, divertente nonché recentissima vignetta del 2014.

<http://blowthecartridge.com/2016/02/17/creatures-2-torture-trouble/>

Un divertente clone di Creatures2 è Critters2, un comico mix tra Creatures e Lemmings dove il puzzle game e il platform a scorrimento orizzontale trovano un singolare e complicato connubio. E' stato sviluppato nel 2000 da Karl-Johan Nilsson, è freeware e gira in ambiente Dos, lo potrete reperire tramite questo link:

[https://archive.org/details/swizzle\\_demo\\_C2](https://archive.org/details/swizzle_demo_C2)

## CURIOSITA'

Il divertente diario di bordo della faticosa programmazione dei fratelli Rowlands può essere letto in questo sito:

[http://www.gamestone.co.uk/zzap\\_world64/apex\\_creatures\\_2\\_torture\\_trouble.php](http://www.gamestone.co.uk/zzap_world64/apex_creatures_2_torture_trouble.php)



plagio, ricordiamoci l'ambientazione, qui siamo nel nulla, in una isoletta sconosciuta del Pacifico, i Fuzzy in questo secondo episodio devono saltare da un'isola ad un'altra per essere messi in salvo, inoltre in aria raccolgono una cosa importantissima per quell'isola piena zeppa di demoni : tokens volanti letteralmente "denaro sospeso a mezz'aria". Eppoi, un bel gufo cicciotto che quasi cattura i Fuzzy rimbalzanti ci sta proprio a pennello. Un tritico di gufi, Fuzzy e soldi.

Il tutto farcito da abbondanti macchine delle torture, ovviamente della famosa ed integerrima ACME.

Anche questa volta i vari ingranaggi ACME si muovono in situazioni meccanicamente complicatissime per assolvere a scopi semplicissimi, seguendo la filosofia del famoso fumettista William Heath Robinson che disegnava meccaniche impossibili per concludersi in buffi scopi basilari. Che dire, sig.Robinson e fratelli Rowlands, grazie di cuore, siete stati geniali.

Idem le animazioni si aggiungono e gli sprite si sommano rispetto al capitolo precedente, oltre che rimbalzare, Clyde calcia i demoni a terra, un Fuzzy gira come un pollo allo spiedo sul fuoco, un altro rimane appeso a testa in giù per una zampetta, un altro ancora viene trasportato a pelo d'acqua mentre Clyde sott'acqua nella boccia di vetro lo sospinge evitando i nemici, vari Fuzzy saltellano e rimbalzano, si spiaccicano a terra, si sciogliono nell'acido, etc..

Un grande applauso va come sempre a tutti gli sfondi del game e, come nel primo capitolo, lo scrolling è impeccabile nonché dotato di ancor più profondità visiva e dettagli particolareggiati all'inverosimile.

Addirittura un gruppetto di pixel è dedicato ad una scopa volante in lontananza che solca il cielo oppure al fumo che esce da un vulcano, infine il sole con gli occhiali, non vogliamo citarlo?

Anche le animazioni dei demoni hanno acquisito più maturità, la prevedibilità e il carattere dei mostriattoli, soprattutto acquatici, mostrano reiterazione di turni e coincidenze nei passaggi attraverso i nemici, molto più complesse. Un esempio possono essere le molteplici meduse che si gonfiano e si strizzano intersecando medesimi quadranti dello schermo oppure animaletti sferici che si gonfiano e si sospingono in salita o discesa nei fondali.

Nel livello dei demoni volanti più voluminosi dobbiamo notare che questi bombardano il povero Clyde con liquami tossici. Questi mostri svolazzanti hanno animazioni incredibili: sono stati programmati con una gestione degli sprite che obbliga il protagonista a movimenti chirurgici per evitare di perdere la vita. Da questo discorso

mi sento in dovere di una piccola precisazione riguardo l'hardware.

Creatures forse non sarà stato il gioco che ha richiesto più precisione nella storia del gameplay della Commodore ma sicuramente l'evoluzione e la maturità di nozioni insite in questo secondo capitolo ha creato non pochi problemi ai giocatori di quel tempo.

**Attenzione spoiler:** L'acquisto di un Joystick stabile e preciso era consigliata da tutte le riviste dell'epoca, addirittura era stato definito "un gioco difficile per giocatori novizi".

Creatures 2 purtroppo non è presente nella recente mini console TheC64mini, ma chiaramente i detentori del gioco possono sperimentarne la giocabilità in emulazione tramite la medesima console, io ci ho provato e vi assicuro che già dal primo capitolo (presente nella console) alcuni difetti del "joystick TheC64" sono realmente stati sottolineati in Creatures 2.

Per fortuna a tutto c'è rimedio e in questa rivista c'è un altro mio articolo inerente ad una possibile parziale soluzione di questo problema.

Ritornando a noi, concludo rammentando che oltre ai soliti personaggi del primo capitolo ci sono altri demoni, gufi, gufoni, uccellacci, grandi mostri volanti, pesci, anfibi surreali e addirittura squali neri come la pece. Tra gli sprite più sorprendenti per l'eccellente gestione della fluidità dobbiamo annoverare il mostro finale, una sorta di cervellone volante che a differenza di tutti gli altri demoni precedenti non ha il famoso sorriso stampato sulla faccia.

Una volta sconfitto il demone finale il tutto si chiude con i ringraziamenti da parte di Clyde a noi player, nonché una divertentissima conclusione dove l'ultimissimo mostro sopravvissuto mostra un viso arrabbiato, poi sorpreso e poi, beh, è una scena veramente divertente e sarei felice se la vedrete coi vostri occhi.

Vale realmente la pena assaporare questo finale soprattutto se, come me, avete adorato la saga Creatures e la Thalamus.

A proposito, pochi mesi fa, la neonata Thalamus Digital ha promesso il remake dei due capitoli e addirittura la creazione del terzo.

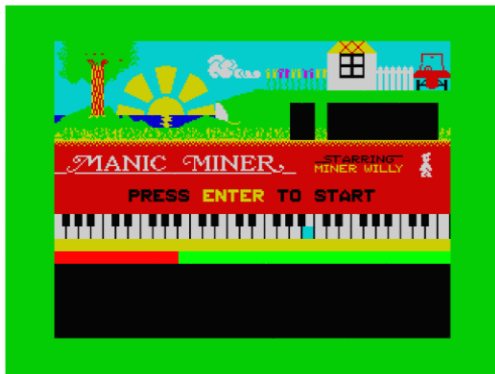
Sarebbe fantastico, sarei veramente onorato di recensire in futuro il loro terzo capolavoro, creato dopo circa un quarto di secolo.

Quindi cari amici, a presto!

*di Michele Ugolini*



## MANIC MINER



## Colonna sonora in-game 1...

Manic Miner è stato il primo videogame ad avere una colonna sonora in game su zx spectrum, esattamente: Nell'antro del re della montagna di Edvard Grieg.



## Colonna sonora in-game 2...

Durante i titoli invece, il giocatore potrà godere del classico Sul bel Danubio Blu di Johann Strauss II.



Manic Miner è un platform a schermata fissa. Venti livelli da affrontare con tre sole vite ed una quantità vergognosa di modi per morire. Solo per terminare il primo livello ricomincerete il gioco un buon numero di volte, figuriamoci per i successivi. Pensare che le dinamiche sono semplicissimi: due tasti per spostarsi ed uno per saltare. In ogni stage ci sarà una porta da aprire e per poterlo fare dovremo raccogliere tutte le chiavi (o altri oggetti variabili di livello in livello) evitando di cadere vittima di nemici e trappole disseminate lungo il percorso. Trattandosi il nostro di un minatore e trovandosi in delle caverne, quello a cui dovrà far fronte però sarà anche lo scorrere del temp...ehm, l'esaurirsi

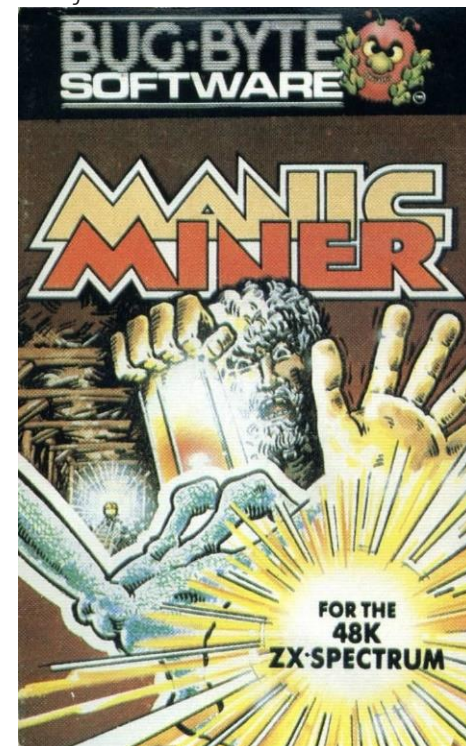
## Manic Miner

Bug-Byte – Anno 1983 – Piattaforma ZX Spectrum

dell'ossigeno respirabile, pena la prematura dipartita. Come poi le opere di Smith ci abitueranno, sin da subito fu chiaro che la difficoltà non era un elemento lasciato lì per caso. Il gioco è punitivo e divertentissimo al tempo stesso. Per trionfare sarà necessario coordinazione, riflessi, memoria ed esperienza. E' facile trovarsi a giocare per ore a Manic Miner, collezionando un numero incalcolabile di morti, eppure ogni volta si sarà riusciti a far un po' più di terreno. Ogni nuovo livello ci saprà mettere di fronte ad una nuova sfida da capire (in primis) ed in seguito superare, nella speranza di non vedere ancora una volta quell'enorme piedone calare dall'alto sulla nostra testa fino a schiacciarsi con la sua scritta Game Over.

La trama di Jet Set Willy, tornato a casa dopo l'impresa e pieno di ricchezze ma con più caos nella magione di quanto ne abbia trovato nelle caverne, vuole il nostro eroe effettivamente trionfante da questa prima impresa, sebbene riuscire a condurlo incolume per tutto il gioco ha dell'epico più dei viaggi di Giasone e gli argonauti. Non so dirvi se Manic Miner sia uno dei migliori giochi in assoluto per Zx Spectrum, ma so per certo che, almeno per me, possedere uno degli home pc di Clive Sinclair senza averlo giocato sarebbe come non aver mai provato Donkey Kong in sala giochi!

di Starfox Mulder



## GIUDIZIO SUL GIOCO

## GIOCABILITA'

90%

Immediato da apprendere, difficile da dominare. Vi basteranno davvero poche partite per uscirne conquistati.

## LONGEVITA'

95%

Venti livelli da portare a termine uno di fila all'altro con tre sole vite (a meno che non bariate con emulazione e save stats) sono una sfida degna di essere affrontata. Fatemi sapere se riuscirete nell'impresa, probabilmente non vi crederò!

# Intervista a Kevin Toms, il creatore di Football Manager

di David La Monaca (Cercamon)

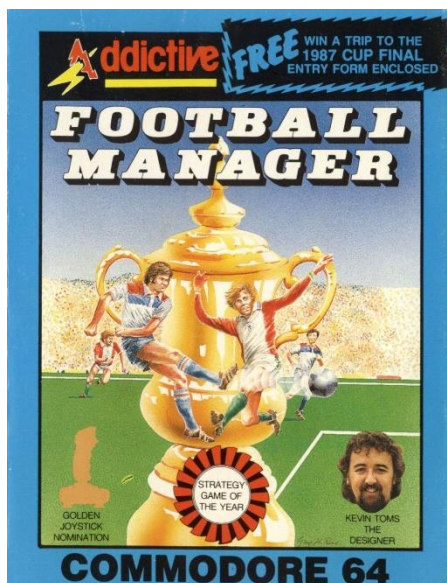
## Introduzione

Negli anni '80 gli home computer a 8/16 bit più diffusi in Italia e in Europa potevano contare su centinaia di giochi ma davvero pochi di questi erano capaci di tenere alta l'attenzione ed un elevato livello di giocabilità nel tempo. Pochissimi poi erano in grado di tenere incollati gli utenti allo schermo per interi pomeriggi o settimane, di diventare un classico e addirittura di creare un genere. È questo il caso del leggendario Football Manager, della Addictive Games, uscito per la prima volta nel gennaio del 1982 per Sinclair ZX81 (e per Video Genie, un clone del Tandy TRS-80) ed in seguito portato su ZX Spectrum, Commodore 64 e su molte altre piattaforme. Le ragioni del successo di Football Manager e della serie legata al suo autore si devono a vari fattori, fra cui spicca una notevole longevità fornita principalmente da un ottimo mix di realismo e imprevedibilità. Proprio l'aderenza alla realtà calcistica del campionato inglese di allora rendeva il gioco, pur nella sua relativa semplicità in confronto ai moderni titoli manageriali calcistici pieni zeppi di opzioni, sempre attraente e stimolante. L'immediatezza della parte strategica concentrata nel controllo di un numero limitato di variabili ed alcuni elementi visuali come gli highlights delle partite disputate e i risultati di ciascuna giornata, concorrevano per l'epoca (e concorrono ancora oggi, perché no?) a regalare al videogiocatore un'esperienza del tutto innovativa e completa. L'atmosfera tipica del calcio inglese veniva resa in modo davvero sorprendente e coinvolgente e la sfida lanciata all'utente di partire dalla Quarta Divisione e giungere fino alla Prima Divisione, passando attraverso la roulette della leggendaria F.A. Cup, garantiva divertimento ed emozioni degne di un allenatore/manager dell'epoca.

## L'intervista

L'ideatore e realizzatore di tutto questo, Kevin Toms, è da sempre una sorta di leggenda vivente per aver di fatto inventato il genere del calcio manageriale, in un

momento in cui i programmatori con le loro idee creative potevano avere tanto successo da guadagnare molto ed assurgere al ruolo di rock star. Lo abbiamo contattato e Kevin è stato così gentile da accettare un'intervista per RetroMagazine.



**RM:** Salve Kevin. Grazie mille per aver accettato il nostro invito per un'intervista tutta dedicata alla tua carriera di progettista e di programmatore di uno dei titoli più famosi di sempre: Football Manager. Molti di noi, da ragazzini, passavano ore ed ore a giocare sui loro home computer, soprattutto su ZX Spectrum e C64 in Italia. Eravamo praticamente ipnotizzati dallo schema di gioco, semplice ma impegnativo, dagli highlights delle partite (vera novità) e dai risultati della giornata di campionato, per controllare la posizione in classifica della propria squadra e giungere alla promozione, evitare la retrocessione o addirittura ambire al titolo nazionale o vincere la F.A. Cup. Ma cominciamo dall'inizio e dai tuoi esordi.

**Kevin,** suppongo che tu sia sempre stato un tifoso di calcio sin da bambino. Dove sei cresciuto e qual era la tua squadra del cuore?

**KT:** Grazie a voi per l'invito. Sono cresciuto a Paignton, nel Devon, Regno Unito e sono ancora oggi un grande tifoso del Torquay United (n.d.a. - il Torquay United attualmente

milita nella National League, la quinta divisione inglese. Nato nel 1899, pur avendo una gloriosa storia, non è mai andato oltre la terza serie).

**RM:** Molti ragazzi negli anni '80 raccoglievano dati e statistiche sulle loro squadre preferite e sui giocatori più famosi per giocare poi campionati di fantasia con giochi da tavolo come il Subbuteo. Sei mai stato un fan delle statistiche calcistiche?

**KT:** Tenere traccia dei risultati è stato molto importante per me, all'epoca. E giocavo anch'io a Subbuteo con gli amici. Con una delle versioni di Football Manager ho anche fornito un libretto per registrare i risultati delle partite e le statistiche principali del campionato.

**RM:** Cosa ti ha spinto sulla strada della programmazione e qual è stata la tua prima esperienza con un computer?

**KT:** Dopo aver lasciato la scuola ho lavorato in un ufficio di amministrazione e poi ho fatto un esame da Programmatore di Computer. Ricordo di aver preso il massimo nella parte di astrazione. :-). Così, diventai un programmatore di sistemi mainframe e lavoravo allo sviluppo di software gestionale.



**RM:** Cosa ti ha maggiormente ispirato a sviluppare un gioco come "Football Manager"?

**KT:** Ho sempre inventato giochi fin da bambino. Nuovi giochi da tavolo, nuovi giochi con un pallone con gli amici. E poi ho cominciato ad unire la mia innata attitudine

nel creare giochi alle mie abilità professionali di programmatore.

**RM: Il lavoro di programmatore e di progettista di giochi è sempre stato quello che avresti voluto fare nella vita? Quando hai deciso che volevi diventare un programmatore e hai cominciato a sviluppare programmi e giochi?**

KT: In un certo senso, senza essere davvero cosciente di che cosa si trattasse, ho sempre voluto diventare un programmatore di computer, ma per lavoro una volta ho anche chiesto di essere assunto come progettista di giochi da tavolo.

**RM: Qual era il sistema hardware che hai utilizzato per sviluppare la primissima versione di Football Manager?**

Direttamente sulle macchine che avevo a disposizione. Prima il Video Genie [n.d.a. - un clone del Tandy TRS-80], poi lo ZX81. :-)

**RM: Hai mai frequentato un corso di programmazione e se sì, quali sono stati e quanto tempo hai impiegato? Oppure come molti programmatori dei primi anni Ottanta, sei stato un autodidatta?**

No, niente corsi. Ai tempi del primo sviluppo di Football Manager ero già un programmatore professionista. Ed ero pagato per sviluppare codice! ;-)



**RM: E' vero che hai programmato la prima versione di Football Manager con il Basic Sinclair? Qual è stata la parte più difficile durante l'implementazione nelle versioni per i computer della serie ZX?**

Sì, è vero. Sarebbe stata una follia utilizzare il linguaggio assembly per un gioco di strategia che prevedeva molte routine matematiche e logiche. E poi, a quel tempo, la pirateria dei giochi non rappresentava alcun problema. Impiegando il Basic non avevo veri problemi

durante lo sviluppo del codice. Si trattava solo di buttare giù un buon progetto e di metterlo a punto. Un migliore algoritmo di ordinamento per la generazione della classifica sarebbe stato utile, ma soltanto la prima parte del calcolo era lenta, perché utilizzai un algoritmo bubble sort.

**RM: Quali strumenti hai utilizzato o hai programmato tu stesso durante il processo di sviluppo (grafica, suono, font, compilazione, ecc.)?**

KT: Come detto, ho fatto tutto utilizzando la sola macchina che avevo a disposizione. Nessuno strumento o tool separato. Qualche tempo dopo era tutto diverso ed ho fatto uso di un cross-compiler per aumentare la portabilità del codice e velocizzare lo sviluppo per i vari sistemi di allora.

**RM: Com'era organizzato il tuo metodo di sviluppo? Ad esempio, disegnavi prima una bozza dei concetti base, per poi passare a progettare le schermate, la grafica e le animazioni e le altre parti del gioco?**

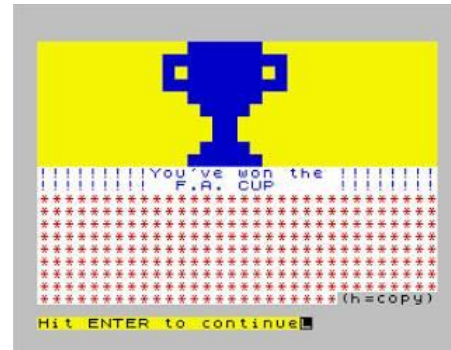
KT: Sia allora che oggi, tendo a mantenere un atteggiamento minimale durante le prime fasi di un progetto. Ma mi piace molto poter riutilizzare ciò che sviluppo. Faccio funzionare piccole parti insieme pur mantenendo l'immagine complessiva in mente. Mi piace sapere di poter contare su quello che ho appena creato.

**RM: Football Manager, persino nella sua prima versione, non è un gioco facile. Come hai progettato il programma per simulare un intero campionato? Hai cominciato da alcune tecniche di programmazione conosciute? Puoi dirci qualcosa sull'algoritmo AI (se previsto) che hai usato per mantenere la simulazione realistica?**

KT: È la mia specialità saper riprodurre la realtà e la sensazione di reale attorno ad un gioco. È il mio tocco da designer di giochi in generale. Lo metto sull'intero progetto, dagli algoritmi principali al numero di variabili di controllo che si hanno. Mi sento a mio agio con tutto questo. E lavoro moltissimo sulla messa a punto per assicurarmi che il risultato sia insieme realistico e imprevedibile. Queste due cose sono in conflitto: se è prevedibile risulterà noioso, se è troppo casuale non sarà realistico.

**RM: Dopo il grande successo di Football Manager per ZX Spectrum, ti sei preso personalmente cura del porting verso le altre piattaforme a 8 bit?**

KT: No, sarebbe stato troppo noioso per me. Per lo sviluppo di Football Manager 2 avevo automatizzato tutto il processo ed il mio codice girava su tutte le macchine target. Con la versione attuale, il remake "Kevin Tom's Football Star\* Manager", ovviamente ho portato tutta la parte di realizzazione ad un altro livello.



**RM: Ti sei mai ispirato ad altri giochi manageriali o agli eventi reali del calcio inglese per le versioni successive di Football Manager?**

KT: Sono sempre stato (e lo sono tuttora) in stretta sintonia con il calcio reale, ma sono anche fortunato perché sono sempre prolifico e pieno di idee. Il mio limite più grande è il tempo necessario per l'implementazione di tutte queste idee.

**RM: Quale genere e quali videogiochi hai giocato nel tempo?**

KT: La maggior parte del tempo la passo scrivendo e programmando. Quando utilizzo i videogiochi mi piacciono quelli in cui non devo impegnarmi troppo per imparare come si gioca.

**RM: Qual era il tuo ruolo nella Addictive Games che ha pubblicato le tue prime versioni di Football Manager?**

KT: Ho fondato e diretto la Addictive Games fino all'avvento di Football Manager 2. [La Addictive Games fu acquistata dalla Prism Leisure Corporation, una software house che, insieme a Kevin, si dedicò a portare la seconda versione di FM su molte delle piattaforme più diffuse nella seconda metà degli anni '80.



Kevin lasciò di lì a poco la Prism in disaccordo con la nuova proprietà. Per la cronaca FM 2 riscosse un discreto successo così come FM World Cup Edition, dedicata ai Mondiali di Calcio di Italia '90. La terza versione di FM fu invece sostanzialmente un mezzo fiasco, il gioco aveva ormai perso il tocco di Kevin ed era stato inzeppato di opzioni tali da togliere l'immediatezza e l'atmosfera del calcio inglese].



**RM:** *Se ci fosse qualcosa che potessi cambiare nella serie di giochi FM e potessi usare la DeLorean di "Ritorno al Futuro", cosa faresti?*

Una sola cosa?! Ce ne sarebbero almeno un centinaio! Quella principale, che sto sperimentando ora con il rinnovo del gioco sulle piattaforme moderne, è sapere quanto fosse importante per gli utenti. A quel tempo non ne avevo idea.

**RM:** *C'è stato un ostacolo che ricordi di aver dovuto superare per far funzionare le prime versioni di FM?*

KT: La prima versione è stata per ZX81, per cui l'ostacolo più grande è stato il limite di memoria disponibile. Per la versione ZX Spectrum la difficoltà maggiore è stata quella di far funzionare bene gli highlights delle partite sulla base della logica booleana.

**RM:** *Di quale versione di FM nelle sue diverse incarnazioni sei più orgoglioso e perché?*

KT: Senza dubbio, la versione per ZX Spectrum. Era molto innovativa per l'epoca. Sono passati molti anni prima che qualsiasi altro programma di questo genere mostrasse le azioni salienti delle partite in forma animata.

**RM:** *Il tuo sito web ufficiale riporta del recente sviluppo delle nuove versioni di FM per i dispositivi mobili (Android e Apple iOS) e per PC Windows. Puoi dirci di più su come*

*hai progettato questa nuova versione che però mantiene un sapore ed uno stile "vecchia maniera"?*

KT: Tutto è nato dai commenti online di alcune persone che mi dicevano di sentire la mancanza dei miei giochi diretti ed intuitivi e che i titoli simili moderni sono enormi e troppo impegnativi da giocare. In quel momento stavo ancora programmando per il mondo mobile ed ho chiesto: "Che ne pensate se rinnovassi FM per smartphone o tablet?". Ho ottenuto una reazione molto positiva alla mia domanda e così ho cominciato a lavorarci nel 2015. La nuova versione ha visto la luce alla fine del 2016. Le animazioni grafiche delle azioni più importanti delle partite le ho aggiunte in seguito, proprio come avvenne con l'originale. Il design interno, la giocabilità ed il codice sono tutti opera mia, quindi ha un'atmosfera molto simile a quella del gioco originale. Ed è molto coinvolgente, gli utenti ci stanno giocando centinaia di stagioni. Ma adesso è anche portatile e si carica istantaneamente. L'ho realizzato prima per Android e iOS, poi per Mac OSX, Kindle Fire e per ultimo Windows 10.

**RM:** *Che mi dici di "The Origin of Football Manager", il libro che, stando a quanto riportato sul tuo sito, stai scrivendo per raccontare tutta la storia su come, nel 1981, hai in realtà creato il genere del calcio manageriale? Come hai pensato di trovare i fondi per il processo di redazione del libro e quando credi che sarà pubblicato?*

KT: Sarà certamente pubblicato nei prossimi mesi. Pensavo di poterlo finire ed averlo pronto da pubblicare più velocemente, ma non ho mai scritto un libro prima d'ora! In parte sarà finanziato grazie al crowdfunding sulla piattaforma Indiegogo.

**RM:** *Quali sono i tuoi progetti per i prossimi due anni riguardo a Football Star\* Manager e non?*

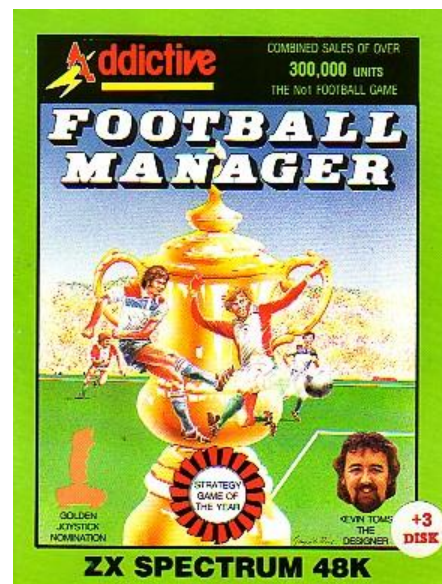
KT: Ho qualcosa di nuovo in serbo per FSM che uscirà in aprile. Mi chiedono di continuo di rinnovare anche i miei altri titoli, per cui sto riflettendo su come organizzarmi per cominciarne eventualmente lo sviluppo.

**RM:** *Ripensando a quando tutto è cominciato, c'è qualcosa che avresti fatto*

*diversamente riguardo al design o ad altri dettagli?*

KT: Sicuramente strutturerei il business in maniera differente. Ma questa è una lunga storia... Avevo solo 24 anni quando ho avviato tutto quanto. Diciamo che ho imparato col tempo. ;-) Gran bella esperienza però!!

Ringraziamo Kevin per la sua gentile disponibilità e passiamo ora a parlare del gioco cercando di capire cosa lo rese così popolare e amato in tutta Europa.



## Il gioco

Abbiamo provato Football Manager nella versione per ZX Spectrum per ragioni storiche (è stata la prima versione ad essere sviluppata con il contributo delle animazioni grafiche) e in quella per C64 per ragioni squisitamente sentimentali (quella con cui il sottoscritto ha nutrito la propria miopia). Le due versioni sono naturalmente molto simili, ma la versione per C64 offre qualcosa in più a livello grafico durante le partite (gli highlights mostrano sprite "blocchettosi", utili soprattutto quando si desidera inveire contro i nostri attaccanti che sbagliano gol sotto porta) mentre il commento sonoro è sostanzialmente piatto e si limita ad un rumore bianco per simulare il tifo e ad un suono più accentuato quando si verifica una segnatura da entrambe le parti.

La primissima versione del gioco vide la luce su Video Genie, un clone del TRS-80, ma non ebbe praticamente riscontro commerciale. Fu

interamente sviluppato in Basic e di lì a poco portato su ZX81 (espanso a 16K) e prevedeva solo l'uso di pagine di testo. Grazie al relativo successo delle vendite per ZX81, Kevin Toms poté permettersi di acquistare riquadri pubblicitari sulle riviste specializzate inglesi e sostenere così lo sviluppo della versione per ZX Spectrum che per la prima volta includeva una rappresentazione grafica animata delle migliori azioni di ogni match disputato dalla squadra prescelta dal giocatore. Oggi quelle immagini fanno sorridere o suscitano "tenerezza" se paragonate con i manageriali calcistici di ultima generazione, ma quando apparvero sugli schermi dei videogiocatori di tanti anni fa riuscivano a provocare in loro quasi le stesse emozioni di soddisfazione o delusione di un vero match vissuto alla TV.

Gli ingredienti del gioco sono semplici ma, come detto, il risultato è un titolo che è diventato un classico, inaugurando un vero e proprio genere nel panorama videoludico: il manageriale strategico. Dopo aver scelto la squadra con cui intraprendere la scalata al campionato inglese, si diventa in pratica il manager del club (come da sempre accade in Inghilterra non si è soltanto il tecnico o l'allenatore, ma si rivestono compiti amministrativi e di gestione di tutto ciò che ruota attorno ad un team) e attraverso varie schermate testuali nelle quali impostare la formazione, vendere o comprare giocatori, badare alle finanze con la possibilità di chiedere prestiti, l'obiettivo è quello di partire dalla Quarta Divisione, l'ultima prevista dal gioco, e cercare di salire fino alla Prima (negli anni '80 la Football Association non aveva ancora ristrutturato i campionati in Premiership, Championship, League One, League Two e National League come oggi). Ciascuna serie è composta da 16 squadre e si ha la possibilità di essere promossi arrivando fra le prime tre in ogni stagione e di giocare per la F.A. Cup, la più antica competizione calcistica al mondo, che corrisponde alla nostrana Coppa Italia.

Stagione dopo stagione, l'obiettivo del giocatore/manager cambia passando dalla promozione alla salvezza, mantenendo la squadra competitiva tecnicamente e controllando sempre molto da vicino le finanze del club per evitare la bancarotta, cosa che porrebbe fine al gioco. Accedere alle varie fasi del gioco risulta comodo e

immediato ed è sempre possibile interrompere la sessione di gioco salvando la stagione in corso partita dopo partita. Fra le opzioni disponibili è presente la possibilità di personalizzare i nomi dei club e dei giocatori, in modo da creare una propria lega di fantasia oppure un campionato di una diversa nazione. In Italia, tra le famigerate "cassette da edicola" che proponevano, impunemente date le leggi allora vigenti in materia di copyright sul software, giochi commerciali piratati a cui veniva semplicemente cambiato il nome, venne pubblicato anche Football Manager con l'improprio nome di "Calcio Mercato", con le squadre ed i giocatori dei campionati italiani di allora. Altre versioni personalizzate ma perfettamente legali furono pubblicate nella ex-Jugoslavia, Germania, Spagna, Repubblica Ceca e Olanda.

Complessivamente le vendite della prima versione di Football Manager raggiunsero le 500.000 copie, mentre la serie completa fino alla versione 3 e includendo le varianti uscite in occasione del Mondiale di Italia '90 si avvicina al milione di copie totali. Un vero successo ed una diffusione molto più ampia di quanto dicano i numeri commerciali se si considera anche una notevole porzione di copie non ufficiali. Cifre che pongono Football Manager fra i giochi per home computer più famosi di tutti i tempi.

### Conclusioni e giudizi

Football Manager – Addictive Games - 1982 – Commodore 64 / ZX Spectrum 48K

### GIOCABILITA' 82%

L'interfaccia utente del gioco è tutta improntata sulle schermate di testo che di volta in volta propongono le scelte strategiche, la gestione delle finanze del club, la compravendita dei calciatori, la selezione della formazione da mandare in campo, fino ad arrivare a seguire sullo schermo il match della giornata di campionato o del turno di F.A. Cup tramite le azioni salienti della partita. Dopo qualche stagione disputata il ripetersi delle schermate potrebbe risultare un po' noioso, ma in fondo il gioco è strutturato su un percorso di operazioni necessarie per farci vivere un'esperienza manageriale emozionante lungo tutto il calendario calcistico del nostro amato club.

### LONGEVITA' 89%

Con sette diversi livelli di difficoltà, modificabile in qualsiasi momento, riuscire nell'intento di raggiungere una promozione o evitare la retrocessione, arrivare a vincere la First Division o la F.A. Cup sono obiettivi tutt'altro che semplici, anche se i meccanismi che regolano l'andamento del gioco e, più in generale, le variabili da controllare non sono poi molte. Come nel calcio reale, non basta vincere un campionato per dirsi soddisfatti, occorre ripetersi nella stagione successiva, magari evitando la retrocessione o cercando di rimpolpare le esigue casse del club dopo aver speso un patrimonio in giocatori di classe. E come nella realtà, un ruolo fondamentale e più attuale che mai è quello della gestione delle finanze, vendendo giocatori al momento giusto e comprando giovani promettenti e di sicuro successo per l'annata successiva.

### Riferimenti

#### Kevin Toms' Football Star\* Manager

#### Versione Android:

<https://play.google.com/store/apps/details?id=com.gmail.toms.kevin.ktfsm&hl=en>

#### Versione iOS:

<https://itunes.apple.com/gb/app/kevin-toms-football-star-manager/id1130764879>

Tutte le versioni: <http://bit.ly/playfsm>

#### Facebook page:

<https://www.facebook.com/KevinTomsGames/>

#### Twitter: @kevintoms /

<https://twitter.com/KevinToms>

#### Kevin Toms' mailing list:

<https://blog.kevintoms.com/my-games-mailing-list/>

#### Versione ZX Spectrum:

<http://www.worldofspectrum.org/infoseekid.cgi?id=0001823>

#### Versione C64 (una delle tante versioni con crack disponibili):

<http://csdb.dk/release/?id=129338>

#### Calcio Mercato (recentemente fixata):

[http://ready64.org/download/scheda\\_download.php?id\\_download=1047](http://ready64.org/download/scheda_download.php?id_download=1047)

#### NEW! Kevin Toms World Football Cup (maggio 2018) Android:

<https://play.google.com/store/apps/details?id=com.kevintoms.ktwfc>



# Eventi: Vicoretro' 2018

di Alberto Apostolo

Domenica 20 maggio 2018 a Vicopisano (PI), si è svolta la terza edizione di Vicoretrò organizzata da **Alberto Peri** e **Luca Cusani** con l'aiuto di **Gabriele Lera** e di tutti gli altri componenti dell'associazione culturale Vicoretrò.

Nella sede di Via Moricotti 1F, all'ombra delle torri della Rocca progettata dal Brunelleschi, si sono dati appuntamento da tutta Italia gli appassionati di computer, programmi e videogiochi che hanno fatto la storia dell'informatica.

## 3a Mostra Vintage Computers & Videogames



Naturalmente il grosso degli espositori proveniva dalla Toscana: espositori del gruppo di Vicoretrò (padroni di casa), il gruppo Firenze Vintage Bit Onlus, il gruppo ClassX, CBMitapages del senese Maurizio Maltinti con Fabio Antimi da Città di Castello (PG), il Museo Dagomari di Prato con alcune macchine calcolatrici, Alessio Bianchi con una selezione di videogiochi, Simone Gentilini con esemplari Olivetti, Marco Fanciulli che proponeva la spiegazione del sistema AGC con cimeli originali della NASA e una riproduzione in forex a grandezza naturale del pannello strumenti del Modulo Lunare L.E.M. usato nelle missioni Apollo (figura3).

Tra gli espositori non toscani occorre citare il gruppo torinese RetroForce.eu con i loro computer MSX, il sig. Antonello Giordano giunto dal Molise con alcune repliche autoconstruite di modelli del passato e il

nutrito gruppo di espositori romani: il gruppo Bit Old che presentava anche un portatile Osborne 1 autografato del suo progettista Lee Felsenstein, Fabrizio Picci con macchine Amiga evolute, Edoardo Massari e Marco Gastreghini possessori di rari cloni Sinclair, Stefano Sanna possessore di una macchina Sun Solaris e di una NeXT Station (prodotta dalla NeXT Computer del compianto Steve Jobs).

Grazie anche al bel tempo (durato praticamente tutto il giorno) ed alla concomitanza con la Fiera Florovivaistica "Castello in Fiore" presente nelle vie e nelle piazze di Vicopisano, la partecipazione del pubblico non è mancata.

Si sono presentate persone di ogni età, curiose di andare a vedere le macchine esposte e di ascoltare spiegazioni e aneddoti sul loro funzionamento, mamme e papà che accompagnavano i figli ansiosi di provare i giochi caricati sui diversi modelli presenti.

Durante la manifestazione, è stato promosso un simpatico concorso per eleggere il migliore espositore. Il primo premio (consistente in un Kindle) è andato a Marco Fanciulli per il suo allestimento in grande stile che ha ricevuto il favore pressoché unanime dei visitatori e degli altri espositori intervenuti.

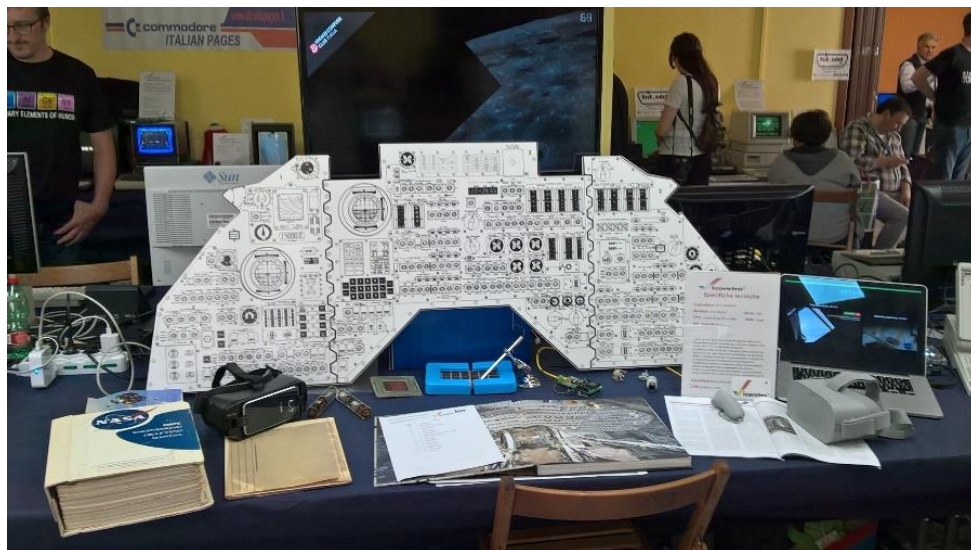


Inoltre è stato dato spazio con una breve intervista (condotta da Luca Cusani) all'ospite d'onore di questa edizione, Loris Massarelli che negli anni '80 fu titolare a Pisa dell'esercizio commerciale Electronic Service, un punto di riferimento per i ragazzi che volevano avvicinarsi all'informatica e un centro di assistenza per la clientela.

Infine si è giunti alla chiusura con i reciproci saluti e l'augurio di ritrovarsi al Firenze Vintage Bit 2018 che si terrà a Lastra a Signa (Firenze) nell'ultima domenica di Novembre.

Si ricorda che quest'anno il Firenze Vintage Bit (organizzato dalla associazione culturale Firenze Vintage Bit Onlus) giungerà alla decima edizione ed è un'altra delle manifestazioni di rilevanza regionale e nazionale inerenti al Retro-Computing e alla divulgazione della storia dell'informatica.

Se volete pubblicizzare i vostri eventi e/o le vostre manifestazioni, oppure inviare il reportage di un evento, non dovete fare altro che contattarci tramite la mailbox: [retromagazine.redazione@gmail.com](mailto:retromagazine.redazione@gmail.com)



*L'allestimento di Marco Fanciulli, vincitore del premio come miglior espositore.*



# Chiusura ed anticipazioni...

di Francesco Fiorentini e Giorgio Balestrieri

## CollaborAZIONE!

Il numero che avete appena terminato di leggere è forse uno dei più completi sinora pubblicati da RetroMagazine. La presenza di numerosi articoli tecnici e recensioni di giochi, l'intervista a Kevin Toms, il reportage dell'evento Vicoretrò e la modifica HW al joystick del C64 Mini contribuiscono a rappresentare le diverse anime del retrò che questa rivista vuole incarnare.

Nel titolo di questa chiusura abbiamo voluto mettere in evidenza, con un gioco di parole, due comportamenti applicandoli al Retrocomputing. **Azione e collaborazione.**

Azione intesa come fare, in contrapposizione all'immobilismo ed all'attesa. Se qualcosa non accade, non aspettiamo che sia sempre qualcun altro a farla succedere, ma rimbocchiamoci le maniche ed attiviamoci per essere noi stessi artefici del cambiamento e fautori di qualcosa.

Mancano eventi? Organizziamoli. Mancano serate a tema? Facciamole. Manca la recensione di un gioco o di un prodotto? Scriviamola! Certo, non è sempre facile farlo così come dirlo, ma di esempi di successo ne abbiamo avuti negli ultimi mesi e spesso basta davvero volerlo ed attivarsi per farlo.

E se l'azione è fondamentale, altrettanto lo è la collaborazione, non a caso una contiene l'altra.

Noi siamo fermamente convinti del valore della collaborazione (...non saremmo certo qui a scrivere se non avessimo trovato

collaboratori per questa iniziativa NdR), perciò l'annuncio di uno "scambio culturale" con la redazione di **OldGamesItalia** ci rende profondamente orgogliosi.

A partire da questo numero, **OldGamesItalia** ci ospiterà sulle sue pagine ed altrettanto faremo noi, dando spazio ai redattori di OGI che vorranno condividere i loro articoli sulle pagine della rivista.

Ma non è finita qui. Benchè restino ancora alcuni dettagli da definire nel momento in cui scriviamo, siamo ben lieti di annunciare anche una collaborazione con **Ed. Master** che distribuirà la nostra rivista come allegato nel **CD di WinMagazine**. Speriamo che questo possa "servire alla causa" e contribuisca a diffondere il verbo del Retrocomputing al di fuori dei canali di nicchia di Facebook e dei siti dedicati.

Per ultimo, ma non certo per importanza, vogliamo anticiparvi l'uscita dell'intervista di **Isacco Luongo**.

Isacco è l'autore del progetto '**Aleph-project: la CPU Z80 diventa a 24 bit di indirizzamento**' e ci spiegherà come, grazie alle ricerche di uno studioso autodidatta, la CPU Z80 può indirizzare fino a 32 megabyte di memoria ed oltre con alcuni accorgimenti di progetto.

Chiudiamo con i **ringraziamenti a tutti i gruppi Facebook** che ci aiutano a condividere la rivista ad ogni uscita e rinnoviamo l'invito a chi volesse dare il suo contributo a farsi avanti.

Al prossimo numero di RetroMagazine!

## Disclaimer

**RetroMagazine** (fanzine aperiodica) è un progetto interamente no profit e fuori da qualsiasi circuito commerciale. Tutto il materiale pubblicato è prodotto dai rispettivi autori e pubblicato grazie alla loro autorizzazione.

**RetroMagazine** viene concesso con licenza: Attribuzione - Non commerciale - Condividi allo stesso modo 3.0 Italia (CC BY-NC-SA 3.0 IT):

<https://creativecommons.org/licenses/by-nc-sa/3.0/it/>

In pratica sei libero di:

**Condividere** - riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato.

**Modificare** - remixare, trasformare il materiale e basarti su di esso per le tue opere.

Alle seguenti condizioni:

**Attribuzione** - Devi riconoscere una menzione di paternità adeguata, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi fare ciò in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o il tuo utilizzo del materiale.

**NonCommerciale** - Non puoi utilizzare il materiale per scopi commerciali.

**StessaLicenza** - Se remixi, trasformi il materiale o ti basi su di esso, devi distribuire i tuoi contributi con la stessa licenza del materiale originario.

**Divieto di restrizioni aggiuntive** - Non puoi applicare termini legali o misure tecnologiche che impongano ad altri soggetti dei vincoli giuridici su quanto la licenza consente loro di fare.

# RetroMagazine

Anno 2 - Numero 7

Direttore Responsabile  
Francesco Fiorentini

Immagine di copertina  
Flavio Soldani

Maggio 2018