



# RetroMagazine

semplicemente retro

Numero 9 - Anno 2 - Settembre 2018 - WWW.RETROMAGAZINE.NET - Pubblicazione gratuita

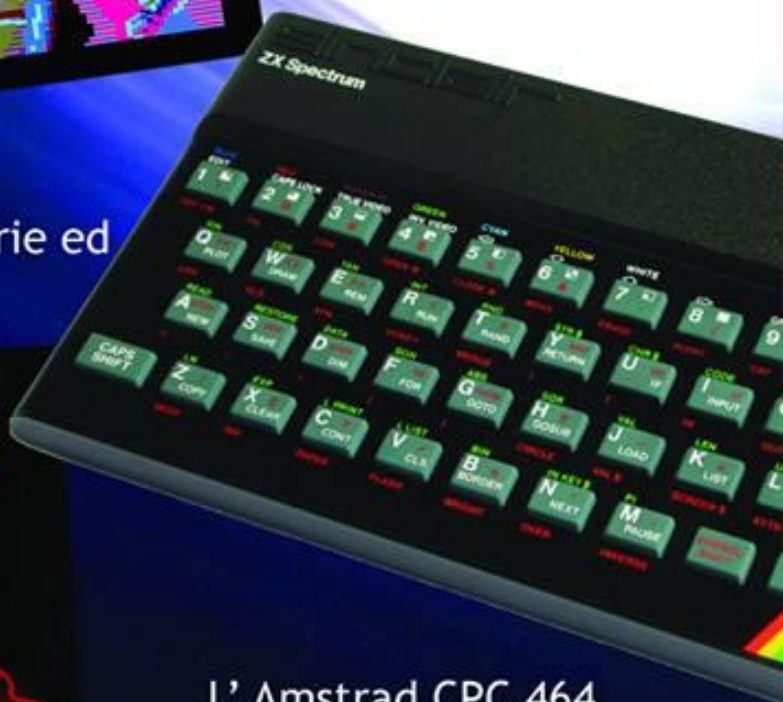


Intervista a Fabrizio Radica

Esplorando L'Amiga  
(seconda parte)

Il Basic dello Spectrum :  
Storia e particolarita'

RetroMath: Pianificare traiettorie ed evitare ostacoli



## IMPOSSIBLE MISSION



L' Amstrad CPC 464



EDITORIALE: L'estate sta finendo e un anno se ne va...

- L'Amstrad CPC 464
- Il BASIC dello Spectrum: storia e particolarità
- Old Programming Games - Da Crobots (1985) a Jrobots (1995)
- B.B.S. - Ritorno al passato
- Fantasy console: TIC 80
- RetroMath: Pianificare traiettorie ed evitare ostacoli
- Il mistero del primo compilatore nella storia dell'informatica
- Esplorando l'Amiga - parte 2
- Le stravaganti calcolatrici di Clive Sinclair
- Console 8bit: Atari 7800

#### GIOCHI

- Xevious (Atari 7800)
- The Dawn of Kernel (Amstrad CPC 464)
- The Tower of Evil (Commodore 16)
- Angolo oscurita': Doctor Hauzer (3DO)
- Rimrunner (Commodore 64)
- Psycho Nics Oscar (Arcade)
- RetroGiochiAmo: Impossible Mission
- Impossible Dennis Caswell Mission
- USCITA DAL RETRO

Intervista a Fabrizio Radica

RM sei tu... chi puo' darti di piu'?

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

# RetroMagazine

Anno 2 - Numero 9

Hanno collaborato a questo numero:

- Alberto Apostolo
- Giuseppe Fedele
- Alessandro Grusso
- Dante Profeta
- Marco Pistorio
- Daniele Brahimi
- Leonardo Boselli
- Leonardo Giordani
- Starfox Mulder
- Michele Ugolini
- Adriano Avecone
- Querino Ialongo
- Flavio Soldani
- Francesco Fiorentini

IN EVIDENZA IN QUESTO NUMERO

## L'estate sta finendo e un anno se ne va...

di Francesco Fiorentini

Anno del Signore 1985, il Commodore 64 fa bella mostra di sé in camera mia accanto ad un desolante televisore in bianco e nero; in sala giochi ho l'imbarazzo della scelta tra Paperboy, Yie Ar Kung-Fu, Green Beret, Ghosts 'n Goblins ed un interessante gioco del calcio chiamato Tehkan World Cup (ma questa è un'altra storia che vi racconterò prossimamente); contemporaneamente i Rigueira se ne escono con una canzone che diventerà un tormentone estivo indimenticabile... Perdonate quindi la citazione della canzone nel titolo, ma si adattava perfettamente alla circostanza di cui voglio parlare in questo editoriale.

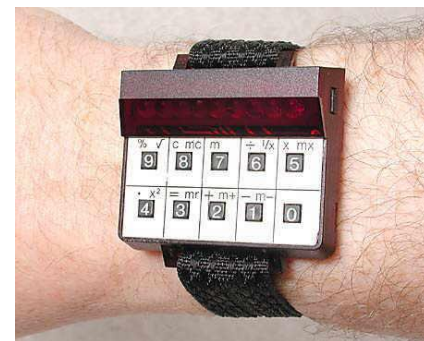
Esattamente un anno fa a Settembre, quando ormai in terra olandese l'estate aveva salutato tutti, prendevo il coraggio a due mani e mi decidevo a pubblicare una richiesta di supporto per avviare il progetto di una rivista a metà tra il tecnico ed il ludico che parlasse di RetroComputing. L'appello fu accolto un po' freddamente perché probabilmente altri ne furono fatti nel tempo senza alla fine produrre un risultato continuativo... Non mi persi d'animo ed il mese successivo postai il primo numero autoprodotta di RM ed i collaboratori a quel punto non tardarono ad arrivare.

Non vi preoccupate, non voglio fare un editoriale autocelebrativo, quanto più una

semplice considerazione su quello che è avvenuto successivamente.

Durante questo anno diversi collaboratori si sono succeduti nella redazione virtuale, alcuni li ho conosciuti personalmente, altri non li ho mai visti di persona, ma con ognuno di essi ho instaurato un rapporto di amicizia. Con qualcuno ci sentiamo regolarmente, con altri un po' meno, ma la passione che ci accomuna fa sì che ogni uscita della rivista sia un piccolo evento da celebrare.

È passato un anno e quello che tenete virtualmente tra le mani è il nono numero. La voglia di scrivere e gli argomenti da trattare sono talmente tanti che ad ogni numero non c'è che l'imbarazzo della scelta per formare la scaletta. Dove arriveremo non lo so, molte riviste ben più famose e prestigiose della nostra hanno chiuso i battenti a causa di svariati motivi, ma principalmente per problemi di budget. Noi fortunatamente non dobbiamo rendere conto a nessuno in termini economici, dobbiamo soltanto assicurarci di fare del nostro meglio per accontentare chi dedica parte del proprio tempo a leggerci. Una cosa però la so e voglio condividerla. Comunque andrà a finire per me sarà un successo, perché nel frattempo ho conosciuto nuove persone, nuovi amici che condividono la mia stessa passione e per questo mi sento immensamente fortunato!



### Le stravaganti calcolatrici di Clive Sinclair

Clive Sinclair non ha prodotto soltanto epici computer, ma anche calcolatrici che il nostro Alberto Apostolo ha definito stravaganti; scopriamo insieme perché.

Articolo a pagina 28



### The Dawn of Kernel

Rilasciato il 23 Agosto 2018 da Juan J. Martinez si tratta di un flip-screen shoot'em up per Amstrad CPC. Correte a leggere la recensione per saperne di più!

Articolo a pagina 33

# L'Amstrad CPC 464

di Francesco Fiorentini

Nel numero scorso, recensendo il sito [cpcrulez.fr](http://cpcrulez.fr), vi avevo promesso di continuare a parlare di una serie di macchine molto particolari che in Italia non ha avuto il successo che forse avrebbe meritato; sto parlando dell'Amstrad CPC.

Come già accennato nel numero precedente la sigla CPC è l'acronimo di **Colour Personal Computer** che sta ad indicare la capacità di queste macchine di gestire grafica a colori. Non tanto perché gli altri computer dell'epoca non avessero questa possibilità, tutt'altro, quanto piuttosto che i computer della serie CPC fossero venduti come sistemi completi con tanto di monitor (a dire il vero anche con monitor monocromatico ma BW&CPC sarebbe stato evidentemente troppo ☺). La strategia di vendere un sistema completo era già stata perseguita in ambito business da altri produttori hardware ed anche dalla Commodore con il PET, ma per il segmento home computer si trattava di una novità che avrebbe preceduto l'Apple Macintosh di almeno un anno.

Dal 1984, anno di presentazione del primo CPC, il 464, al 1990, fine vendita, la famiglia Amstrad CPC si compose di questi modelli:

<b>CPC464</b>	Processore ZX, 64 kB, 42 kB utente e registratore cassette integrato.
<b>CPC472</b>	Identico al 464 ma con 72 kB per il mercato spagnolo.
<b>CPC664</b>	Identico al 464 ma con il lettore floppy integrato.
<b>CPC6128</b>	Identico al 664 ma con 128 kB, utente 42 kB, 64 kB per RAM disk.
<b>CPC464 Plus</b>	Alcune modifiche estetiche e miglioramenti HW.
<b>CPC6128 Plus</b>	Alcune modifiche estetiche e miglioramenti HW.

L'intera gamma CPC ha riscosso un discreto successo di vendite registrando in totale circa

3 milioni di pezzi venduti, di cui 2 milioni soltanto del 464, e quindi, per omaggiarne il successo, ho deciso di parlare in questo numero proprio di questo modello.

## Il computer Amstrad CPC464



Il CPC 464 è stato il primo computer prodotto da Amstrad: la storia vuole che sia stato Alan Sugar, il presidente di Amstrad in persona, a volere fortemente che l'azienda entrasse nel business degli home computer fiutando l'affare dopo aver visto i dati positivi delle vendite dello ZX Spectrum.

La macchina nacque quindi per imporsi come un diretto concorrente dello ZX Spectrum e del Commodore 64 e da entrambi prese qualcosa: montava infatti un processore Z80 come lo Spectrum e nella quantità di memoria e nel nome ricordava molto il famoso computer di casa Commodore.

Il computer fu presentato a Londra il 12 Aprile 1984 con l'altisonante nome di CPC464: il numero 464 nasceva dall'unione della frequenza della CPU (4MHz) con la quantità di RAM installata (64kB) mentre CPC lo abbiamo già spiegato sopra. ;-)

## Caratteristiche tecniche

Come accennato precedentemente la macchina montava una CPU **Zilog Z80A** a 4 Megahertz, 64 kB di RAM e 32 kB ROM suddivise in due banchi da 16 kB contenenti il **Firmware** ed il **Locomotive BASIC**.

Il chip grafico era un **Hitachi HD6845 CRT**, clone del Motorola MC6845, capace di passare dalla modalità NTSC a quella PAL via software grazie al fatto che i tempi di

sincronizzazione del segnale video erano programmabili. Il chip poteva inoltre gestire le seguenti 4 modalità grafiche:

– **Mode 0**: 160x200 pixel con 16 colori selezionabili da una palette di 27

– **Mode 1**: 320x200 pixel con 4 colori senza vincoli

– **Mode 2**: 640x200 pixel con 2 colori; l'alta risoluzione per il CP/M

– **Mode 3**: 160x200 pixel con 4 colori: accessibile solo tramite programmazione in Assembly e non gestibile tramite BASIC

Il sonoro era garantito da un chip **AY-3-8912** capace di generare 3 voci ed 8 ottave; un chip di tutto rispetto per lo standard dell'epoca, ma non all'altezza del ben più famoso SID del Commodore 64.

Il progetto iniziale dell'Amstrad CPC464 prevedeva l'utilizzo di un microprocessore 6502, ma sfortunatamente tale progetto naufragò dopo che uno dei due progettisti iniziali dovette ricorrere a cure riabilitative a causa dello stress a cui si sentì sottoposto e, grazie anche al fatto che il BASIC della Locomotive Software era stato sviluppato per essere utilizzato con lo Z80, si decise di modificare il progetto hardware e passare dal 6502 allo Z80.

Il passaggio al processore Z80 non fu indolore, ma portò in dote alla macchina di Sugar la possibilità di utilizzare il sistema operativo CP/M, distribuito con la versione 2.2, e di accedere di conseguenza ad un vasto catalogo software da ufficio che collocò il CPC464 non solo nel settore casalingo ma anche in quello professionale per piccole imprese o per imprese a gestione familiare.

## Firmware

Il Firmware dell'Amstrad CPC, il suo sistema operativo, occupava 16kB, un intero chip ROM dei due disponibili. A differenza di altri computer dell'epoca, nei quali molte delle funzionalità del Firmware rimasero

```

Amstrad 64K Microcomputer (v1)
©1984 Amstrad Consumer Electronics plc
and Locomotive Software Ltd.
BASIC 1.0
Ready

```

inizialmente sconosciute in quanto non ufficialmente documentate, Amstrad decise di pubblicarne tutte le specifiche all'uscita del computer in un manuale intitolato Firmware Guide ed inserito a catalogo come **SOFT 968**. Questa mossa si rivelò decisamente vincente, infatti all'uscita dell'Amstrad CPC464 il computer era già dotato di un database, un word processor ed un paio di giochi piuttosto buoni. Il perché è presto detto, i programmatori non dovettero perdere tempo ad analizzare l'hardware del computer perché le funzionalità del Firmware erano già documentate, potendo così concentrarsi soltanto sullo sviluppo del software.

Vale inoltre ricordare che il Firmware del CPC era piuttosto avanzato con funzionalità messe a disposizione anche del Locomotive BASIC. Alcuni dei comandi dell'interprete, come WINDOW, PLOT, DRAW... non erano infatti comandi basic nativi, ma semplici chiamate a funzioni del firmware. In pratica il SO dell'Amstrad faceva tutto il lavoro sporco. Un bel vantaggio!

### Locomotive BASIC

Il BASIC a corredo della linea CPC fu sviluppato dalla Locomotive Software. Si tratta essenzialmente di un dialetto BASIC con l'aggiunto di diversi comandi capaci di gestire grafica e sonoro. Il Commodore BASIC, fornito a corredo del C64, per accedere alle primitive grafiche e sonore deve far ricorso ai comandi POKE. Il BASIC dello Spectrum ha dalla sua dei comandi dedicati al comparto grafico e sonoro. Il Locomotive BASIC si spinge addirittura oltre e si distingue per una ancora migliore implementazione del

comando SOUND, capace di gestire quasi tutti i parametri sonori a disposizione del chip AY-3-8912.

Un altro vantaggio del Locomotive BASIC è una migliore gestione degli argomenti passati alle estensioni RSX. Come per altri computer dell'epoca, le funzionalità dei programmi BASIC erano estese ricorrendo a codice

macchina caricato in RAM, ROM o su cartridge. Questa funzionalità, conosciuta sul Locomotive BASIC come RSX (Resident System eXtensions), era particolarmente efficace sull'interprete del CPC. I comandi aggiunti al Locomotive BASIC erano facilmente riconoscibili perché preceduti dal carattere "|" pipe ed avevano il vantaggio di essere facilmente rilocabili in RAM. Alcuni dei più famosi comandi RSX erano mutuati dalla ROM dall'AMSDOS, come |REN, |DISC, |TAPE, mentre altri comandi erano semplicemente aggiunti in RAM e richiamati alla bisogna dal BASIC. Tornando all'affermazione iniziale; a differenza di altri dialetti, dove il passaggio degli argomenti alle estensioni basic avviene tramite concatenazione di stringhe, il Locomotive BASIC prevedeva un passaggio diretto.

### Vendite

Le aspettative iniziali di vendita del CPC 464 fatte dagli analisti si aggiravano intorno alle 100.000 unità, con l'intento di Amstrad di piazzarsi nella top ten dei venditori di home computer, ma ben presto queste stime dovettero essere riviste al rialzo. Si calcolano infatti circa 2 milioni di esemplari venduti in tutta Europa, con ottimi risultati in Inghilterra, ovviamente, in Francia ed in Germania. Anche in Spagna, dove il CPC 464 fu distribuito tramite Indescorp come CPC 472, con 8 kB di RAM aggiuntivi per aggirare una buffa legge che prevedeva una sovrattassa per i computer equipaggiati con 64, o meno, kB RAM.

In Italia purtroppo non sono stati raggiunti gli stessi risultati, forse a causa della massiccia

presenza della concorrenza, Commodore 64 e ZX Spectrum su tutti, ma molto più probabilmente per una minor disponibilità di software pirata dedicato a questo computer. Volenti o nolenti la pirateria "autorizzata" era parte integrante del tessuto sociale in cui si muovevano in Italia gli home computer degli anni '80. Per rendersene conto, basta dare un'occhiata all'ottimo sito **Edicola 8 bit** - <http://specialprogramsipe.altavista.org/default.php>, che contiene centinaia di cassette dedicate a C64, Spectrum ed MSX, ma nessuna specifica per l'Amstrad. Eppure mi ricordo di averne vista qualcuna quando ero ragazzo ☺...

Concludendo possiamo affermare che, nonostante questa l'idea sia nata come una mera operazione commerciale a fronte dei risultati ottenuti da altri brand ben più affermati, il progetto CPC si è rivelato tutt'altro che fallimentare ritagliandosi una parte da protagonista nella storia degli home computer ad 8 bit. Personalmente non possiedo nessun modello CPC nella mia collezione, ma non mancherò sicuramente di testare giochi, software e codice con gli emulatori, arricchendo così la mia personale esperienza di amante delle macchine 8/16 bit.

Nei prossimi numeri vedremo insieme quale emulatore utilizzare e come configurarlo per sfruttare al meglio il tanto software prodotto per la famiglia CPC. Ah, esistono anche degli ottimi emulatori online, ma per il momento non vi svelo altro...

Continuate a seguirci!

### Bibliografia

Wikipedia:

[https://it.wikipedia.org/wiki/Amstrad\\_CPC](https://it.wikipedia.org/wiki/Amstrad_CPC)  
[https://en.wikipedia.org/wiki/Amstrad\\_CPC\\_464](https://en.wikipedia.org/wiki/Amstrad_CPC_464)

CPCWiki: <http://www.cpcwiki.eu>

Per una lista dei comandi RSX mutuati dall'AMSDOS:

<http://www.cpcwiki.eu/index.php/AMSDOS>

Per una lista esaustiva dei comandi del Locomotive BASIC:

[http://www.cpcwiki.eu/index.php/Locomotive\\_BASIC](http://www.cpcwiki.eu/index.php/Locomotive_BASIC)

# Il BASIC dello Spectrum:

## storia e particolarità

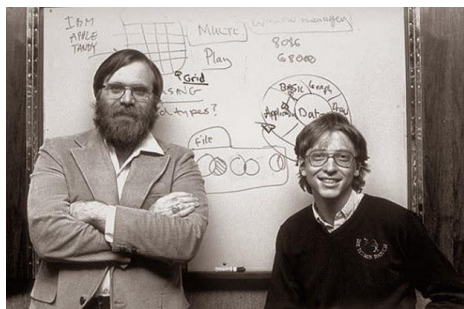
di Alessandro Grussu

BASIC è l'acronimo di *Beginner's All-purpose Symbolic Instruction Code*, "Linguaggio di istruzioni simboliche di uso generale per principianti", un linguaggio di programmazione realizzato nel 1964 all'Università di Dartmouth (New Hampshire, USA) dai proff. John G. Kemeny e Thomas E. Kurtz con l'obiettivo preciso di fornire a coloro che si avvicinano allo studio dell'informatica una piattaforma di sviluppo semplice da imparare, caratterizzato da un complesso di istruzioni preciso e ben definito, orientato quindi alla facilità d'uso piuttosto che alla potenza di esecuzione.



T.E. Kurtz e J.G. Kemeny

Malgrado i due autori avessero dato vita a una vasta opera di divulgazione, propagando la loro creazione su riviste specializzate e di settore, le quali davano spesso spazio ai lettori per la pubblicazione dei loro listati, la vera fortuna del BASIC cominciò negli anni Settanta, grazie al successo di uno dei primissimi microcomputer della storia, l'Altair 8800, prodotto dalla MITS (Micro Instrumentation and Telemetry Systems) a partire dal 1975. Nello stesso anno, due giovani programmatori, Bill Gates e Paul Allen, proposero alla MITS l'acquisto in



Paul Allen e Bill Gates

esclusiva di un interprete BASIC da far girare sull'Altair.

Né Gates né Allen avevano elaborato un interprete BASIC per l'Altair al momento di proporlo alla MITS; l'idea fu di Gates, e Allen adattò all'uopo un interprete BASIC da lui in precedenza realizzato per il *mainframe* PDP-10, aiutandosi col manuale tecnico dell'Altair e sperimentandolo su una piattaforma in *timesharing* a Boston. Fu proprio per commercializzare il loro nuovo prodotto che Gates e Allen, dopo aver siglato l'accordo con la MITS, fondarono la *Micro-Soft*, in seguito Microsoft. L'accordo andò in porto e il BASIC, nella versione di Gates e Allen – occupante solo 4 Kb di memoria e denominata appunto *Altair BASIC* – iniziò a diffondersi sui primi elaboratori "casalinghi".



Clive Sinclair

Tra la fine degli anni '70 e l'inizio degli anni '80, la comparsa sul mercato dei primi home computer per uso generico continuò a favorire la conoscenza e l'utilizzo del BASIC. Differenziati in vari "dialetti" a seconda delle macchine su cui giravano, gli interpreti BASIC portarono il linguaggio ideato da Kemeny e Kurtz nelle case di milioni di utenti. Tra le piattaforme più importanti di questo periodo dotate di interprete BASIC ricordiamo l'Apple II della Macintosh, la famiglia ZX (ZX80, ZX81 e ZX Spectrum) della Sinclair, la serie PET e i VIC-20 e CBM-64 della Commodore, tutte le

generazioni dello standard MSX e il Tandy TRS-80 della Radio Shack.

Il BASIC dello Spectrum è una variante detta appunto BASIC Sinclair. Le sue origini risalgono al BASIC per lo ZX80 sviluppato nel 1979 da John Grant della Nine Tiles Network. Clive Sinclair desiderava mantenere più basso possibile il prezzo dello ZX80 per l'utente finale, per cui il pagamento di una licenza alla Microsoft era fuori discussione.



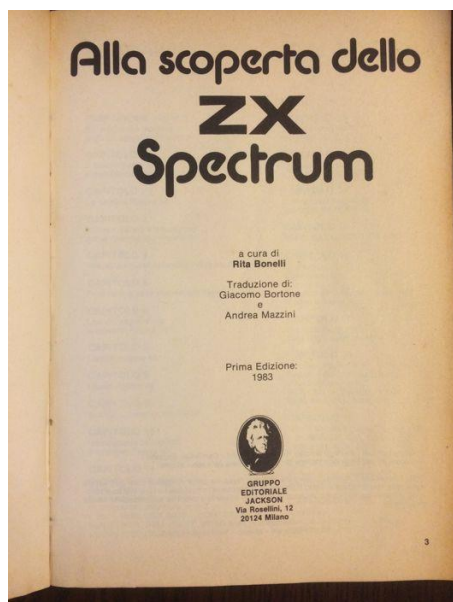
John Grant

Di qui l'idea di rivolgersi a Grant, che realizzò una versione del BASIC assai funzionale, date le risorse estremamente limitate del microcomputer di Sinclair. Si deve a lui, tra l'altro, l'indicazione degli errori di sintassi nelle linee di programma prima che vengano inserite, il che facilita enormemente l'apprendimento e l'uso del BASIC. Lo ZX81 fu poi equipaggiato con un BASIC largamente rielaborato da parte di Steve Vickers, in forza alla Nine Tiles dal gennaio 1980.



Steve Vickers e Richard Altwasser

Quando fu la volta del BASIC dello Spectrum non fu però possibile né a Vickers né a Richard Altwasser, l'ingegnere a cui Sinclair aveva affidato il disegno dello hardware dello Spectrum, realizzare tutte le loro idee, sia per l'ostinazione di Sinclair di far rientrare tutto in un quadro già stabilito sulla base dei costi, sia perché il tempo per le rifiniture non era sufficiente. A ciò si aggiunsero le divergenze di natura finanziaria tra Sinclair e la Nine Tiles sul pagamento delle royalty sulla ROM dello Spectrum, e il fatto che Vickers e Altwasser si erano licenziati dalla Sinclair Research per fondare la loro compagnia Cantab, la quale realizzerà il Jupiter Ace, un clone dello ZX80 dotato del linguaggio Forth al posto del BASIC. Lo Spectrum pertanto uscì nell'aprile 1982 con un BASIC non interamente completato, sul quale la Nine Tiles lavorò persino dopo il lancio del computer, ma senza approdare a risultati di rilievo.



Le peculiarità del BASIC Sinclair rispetto agli altri "dialetti" non sono tantissime e rendono la conversione di un programma un compito non eccessivamente complicato. Ciononostante esistono e di seguito ne indicheremo le più notevoli.

La maggior parte dei BASIC usa l'aritmetica a virgola mobile, con variabili analoghe a quelle dello Spectrum, ma alcuni permettono anche di specificare variabili intere aggiungendo al nome della variabile il simbolo %. Di conseguenza A indica una variabile a virgola mobile, mentre A% una variabile intera. Talvolta il programma può ricorrere

all'aritmetica intera, specialmente nella divisione tra numeri il cui risultato debba essere ricondotto all'intero. In tal caso sullo Spectrum si adopera la funzione INT, la quale tralascia la parte decimale di un numero non intero, per cui abbiamo

```
LET X = INT (Y/100)
```

che in altri "dialetti" BASIC verrebbe scritto

```
X% = Y/100
```

Come si può notare da questi esempi, nell'assegnazione di un valore a una variabile il LET non può essere omissso. Lo Spectrum inoltre non fa differenza tra maiuscole e minuscole nei nomi delle variabili, per cui A e a, K\$ e k\$, PIPPO e plpPo non vengono considerate diverse.

Nella gestione delle condizioni, IF...THEN manca di ELSE, per cui casi diversi devono essere esplicitati. THEN inoltre non può essere omissso. Per esempio:

```
10 IF A = B PRINT "A e' uguale a B" ELSE PRINT "A e' diverso da B"
```

sullo Spectrum diventa:

```
10 IF A = B THEN PRINT "A e' uguale a B"
```

```
20 IF A <> B THEN PRINT "A e' diverso da B"
```

Si noti che se le condizioni sono solamente due, nella riga 20 si potrebbe inserire semplicemente PRINT "A e' diverso da B" in quanto lo Spectrum, vista la falsità della condizione precedente, salta alla riga successiva.

Un modo alternativo per giungere allo stesso risultato implica l'uso dell'operatore logico AND:

```
10 PRINT "A e' "+("uguale a B" AND A=B)+("diverso da B" AND A<>B)
```

Il BASIC dello Spectrum non dispone dei cicli condizionati DO/REPEAT...UNTIL. Per ottenere lo stesso risultato è necessario utilizzare una struttura del tipo IF...THEN GO TO. Per esempio:

```
10 REPEAT
20 INPUT "Nome?"; a$
30 UNTIL a$ = "fine"
```

diventa:

```
10 REM
20 INPUT "Nome?"; a$
30 IF a$ <> "fine" THEN GO TO 10
```

Una struttura simile, o una combinazione di operatori AND e di salti calcolati, si usa sullo Spectrum al posto della condizione ON...GOTO (o GOSUB). Se per esempio avessimo

```
ON X GOTO 200, 300, 400, 500
```

il programma andrebbe alla linea 200 se X = 1, 300 se X = 2 e così via. Sullo Spectrum lo stesso compito può essere eseguito in diversi modi. Si potrebbe innanzitutto indicare una serie di salti condizionati:

```
10 IF X = 1 THEN GO TO 200
20 IF X = 2 THEN GO TO 300
```

e così via; oppure, dal momento che lo Spectrum ammette l'uso di un'espressione numerica per indicare il numero di riga a cui saltare o di un sottoprogramma, es. GO TO N\*10, GO SUB A+1000 ecc., si può impiegare una serie di condizioni stabilite attraverso l'uso di AND:

```
GO TO (200 AND X = 1) + (300 AND X = 2) + (400 AND X = 3) + (500 AND X = 4)
```

Possiamo anche semplificare il tutto così:

```
GO TO 100 + X*100
```

Le operazioni di divisione delle stringhe (o slicing), che in molte versioni di BASIC avvengono tramite le funzioni LEFT\$, MID\$ e RIGHT\$, sullo Spectrum si compiono tramite l'operatore TO.

LEFT\$(X\$, Y) darà i primi Y caratteri di X\$; il suo equivalente nello Spectrum è X\$(TO Y).

MID\$(X\$, Y, Z) fornisce il numero Z di caratteri della stringa X\$ che parte dal

carattere numero Y. Ciò equivale a  $X\$ (Y \text{ TO } Y+Z-1)$ .

`RIGHT$(X$,Y)` fornisce il numero Y di caratteri che si trovano più a destra nella stringa X\$. La sua traduzione per lo Spectrum è  $X\$ (LEN X\$-Y+1 \text{ TO})$ . `LEN`, come in altri BASIC, è la funzione che restituisce il numero di caratteri di una variabile alfanumerica.

Inoltre, se vogliamo assegnare a una variabile alfanumerica una parte dei caratteri di un'altra, useremo la sintassi `LET A$ = X$(A TO B)`, dove A rappresenta il carattere numero A di X\$ e B il carattere numero B. A e B possono essere omessi se A coincide con il primo carattere di X\$ o se B coincide con l'ultimo.

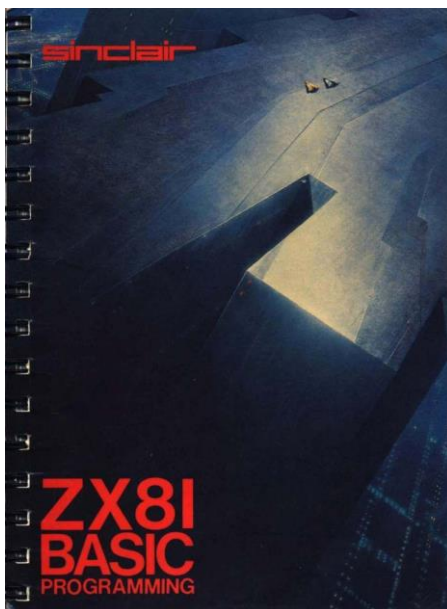
La maggior parte dei BASIC impiega, per le matrici (o *array*), indici che iniziano da 0 e non da 1. Di conseguenza, `DIM A(3)` definisce di solito una matrice di quattro elementi, da A(0) ad A(3). Per lo Spectrum, invece, questa sarebbe una matrice di tre elementi, da A(1) ad A(3).

Gli array di alcuni BASIC usano nomi di variabili costituiti da più caratteri. Sullo Spectrum questi nomi dovranno essere convertiti in caratteri singoli.

Gli array di stringhe dello Spectrum sono molto simili a quelli della maggior parte degli altri BASIC (tranne per il fatto che, anche in questo caso, gli altri "dialetti" fanno partire la numerazione degli indici da 0 anziché da 1). Esistono, però, differenze nel modo in cui concepiscono la lunghezza delle stringhe in una matrice. L'istruzione `DIM`, oltre a definire in numero di stringhe in una matrice, definisce anche la lunghezza massima di ciascuna stringa, in quanto, se queste sono troppo lunghe, viene emesso un segnale di errore. Lo Spectrum regola la lunghezza di una stringa in modo da adattarla esattamente alla lunghezza dimensionata per la stringa stessa. Questo può, talvolta, creare qualche problema quando viene eseguito il confronto tra due stringhe, di cui una compresa in un array. Per esempio, l'istruzione `DIM` di stringa

```
DIM A$(2,2) [10]
```

per definire un array di stringhe 2x2 nella quale ciascun elemento ha una lunghezza



massima di 10 caratteri, sullo Spectrum andrebbe scritta nella forma

```
DIM A$(2,2,10)
```

Alcuni BASIC permettono poi il dimensionamento di più matrici con una sola istruzione `DIM`. Questo non è possibile sullo Spectrum, per cui

```
10 DIM A(2,2), B(50)
```

diventa

```
10 DIM A(2,2): DIM B(50)
```

La funzione `GET` o `GET$`, che in alcuni BASIC serve ad attendere la pressione di un tasto da parte dell'utente e ritorna il codice numerico, o il carattere, corrispondente al tasto premuto, è resa sullo Spectrum da `INKEY$`, la quale però non arresta il computer. Un semplice salto condizionato è necessario per far ciò, ad esempio

```
10 LET A$ = INKEY$: IF INKEY$ = "" THEN GO TO 10
```

La funzione `VAL` ammette solo stringhe composte da cifre. In caso contrario, altri BASIC le assegnano il valore 0, mentre sullo Spectrum ciò causa l'arresto del programma con un messaggio di errore 2 *Variable not found* (variabile non trovata) se la stringa comincia per lettera e contiene un nome non in precedenza assegnato a una variabile numerica, o *C Nonsense in BASIC* (privo di significato in BASIC) negli altri casi.

Nei cicli `FOR...NEXT` sullo Spectrum, `NEXT` deve sempre essere seguito dalla variabile cui si riferisce il relativo `FOR`, laddove in vari altri BASIC non è richiesto.

Non è necessario concludere il programma con `END`. Se occorre uscire dal programma prima della sua completa esecuzione, ad esempio al verificarsi di una certa condizione, si può usare `STOP`.

## Bibliografia

Bonelli, Rita (a cura di), **Alla scoperta dello ZX Spectrum**, Gruppo Editoriale Jackson, Milano 1983.

Dale, Rodney, **The Sinclair Story**, Duckworth, London 1985.

Grussu, Alessandro, **Spectrumedia**, Universitalia, Roma 2012.

James, Mike, **An Expert Guide to the Spectrum**, Granada, London 1983.

Jones, Dilwyn, **Beyond simple BASIC – Delving deeper into your ZX Spectrum**, Interface, London 1983.

Lord, Mike, **Alla scoperta del BASIC Spectrum**, Gruppo Editoriale Jackson, Milano 1985.

Owen, Andrew, **The History of Sinclair BASIC**, s. d.



# Old Programming Games

## Da Crobots (1985) a Jrobots (1995)

di Leonardo Boselli

**OldGamesItalia**

Come anticipato nei numeri precedenti, **RetroMagazine** ha stretto un sodalizio con **OldGamesItalia** collaborando ad uno "scambio culturale" tra le due redazioni con lo scopo di arricchire l'offerta per i nostri comuni lettori. Questo che state leggendo è soltanto il primo degli articoli scritti dalla redazione di **OGL** che **RM** ospiterà nelle pagine della sua rivista. Anche se non si tratta di un articolo creato ad-hoc, si sposa perfettamente con l'argomento trattato in questo stesso numero nella rubrica **RetroMath**. Chissà che qualcuno non decida di applicare le strategie suggerite da **Giuseppe Fedele** ad uno dei giochi qui presentati da **Leonardo Boselli**.

### Programming Game... che roba è?

No, non avete letto male. Il titolo di questo articolo cita proprio i "programming games" e non il "game programming".

Immaginate un'area di un chilometro quadrato su cui scorrazzano, a folle velocità, robot cingolati che si individuano mediante impulsi radar e si sparano potenti cannonate. Da ciò che vi siete figurati mentalmente, togliete la moderna grafica tridimensionale di una PS4, sostituitemela con una vista planimetrica a caratteri, magari visualizzata da un bel monitor a fosfori verdi, e avrete **Crobots**, il primo "programming game" a girare su DOS.

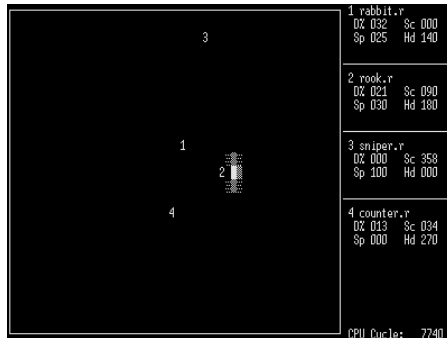
A questo punto molti si chiederanno che cos'è un "programming game"?

Si tratta di un gioco, come dice il nome, ma molto particolare. Un gioco per programmatori. Nel caso di **Crobots**, i giocatori dovevano scrivere, in un linguaggio molto simile al C, gli algoritmi che gestivano il comportamento di questi "robot" stilizzati, cercando di realizzare la miglior strategia per sopravvivere nell'arena, in cui vigeva la logica del "ne rimarrà soltanto uno".

Un gioco, quindi, in cui non si agisce in prima persona: si crea un'intelligenza artificiale e si osserva il suo comportamento contro altre "intelligenze" programmate da altri giocatori.

Descritto in questo modo, sembra un gioco astratto e poco coinvolgente, tuttavia chi avesse provato sa che ci si affeziona alle proprie creature, si gioisce quando sopravvivono e si soffre quando soccombono, ma il dolore passa presto e si comincia subito a escogitare migliorie da apportare al codice per ottenere azioni difensive e offensive più efficaci.

Di giochi di questo genere, pur non essendo molto popolari, è piena la rete. **Crobots** è stato uno degli antesignani e ha generato una quantità incredibile di cloni, alcuni dei quali estinti da anni. A sua volta, però, era una sorta di clone di **RobotWar**, un gioco per il PLATO computer system degli anni '70 (che consisteva in migliaia di terminali grafici connessi a una dozzina di mainframe), culminato in una versione per Apple II del 1981.



### Crobots... a programming game for programmers, or aspiring ones

**Crobots** vide la luce nel dicembre del 1985 grazie a Tom Poindexter che lo distribuì come shareware. In quella lontana epoca ante-internet, quando imperversava la grafica a 16 colori del Commodore 64 e le "avventure testuali" commerciali stavano già andando incontro al loro inesorabile declino, non passò del tutto inosservato.

I punti di forza di **Crobots** consistevano nel linguaggio ad alto livello, simile al C, e nella VM (virtual machine) che eseguiva il codice macchina ottenuto nella compilazione. Il semplice insieme di funzioni con le quali il giocatore poteva leggere i dati dei sensori (lo

"scanner") e inviare comandi al robot (per regolare la velocità del robot e indirizzare i colpi del "cannone") ne facevano un gioco interessante per chi aveva voglia di imparare a programmare divertendosi.

Io scoprii **Crobots**, come altri ragazzi italiani, grazie agli articoli di Corrado Giustozzi sull'ormai mitica rivista **MC Microcomputer** (non più pubblicata da tempo e [disponibile per la consultazione online](#)).

La rivista si è occupata del gioco dal 1990 al 1998 ([qui potete leggere il primo articolo sull'argomento](#)). Il redattore che se ne occupava organizzò in tutto otto tornei a cadenza annuale.

Ma la storia di **Crobots** non si è esaurita con **MC Microcomputer**. Gli appassionati, soprattutto italiani, hanno continuato a occuparsi del gioco, a produrne versioni che potessero essere eseguite da computer molto più veloci e potenti di quelli del 1985 e su <http://crobots.deepthought.it> si possono tuttora leggere i risultati dei tornei periodici che vengono organizzati.

E si può addirittura provare l'ebbrezza di vedere **Crobots** eseguito sui fosfori verdi di un [emulatore DOS scritto in JavaScript](#).

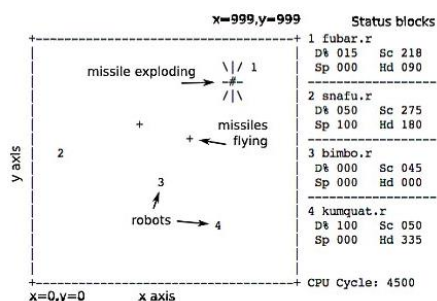
**Crobots**, quindi, è un gioco di nicchia che ha una storia antica (relativamente ai tempi dell'industria videoludica) giunta fino ai giorni nostri grazie ad autentici appassionati e alla felice intuizione di Tom Poindexter che nel 2013, a 28 anni dalla pubblicazione, ha deciso di rendere pubblici i sorgenti (disponibili qui: <http://github.com/tpoindex/crobots/>) un tempo disponibili solo per posta, previo pagamento di 20\$.

A questo punto, per soddisfare la curiosità del lettore, sarebbe interessante illustrare le dinamiche del gioco, le principali tattiche, citare i campioni.

Per quanto riguarda **Crobots**, potete fare riferimento ai link che ho riportato sopra. Io, invece, approfondirò questi aspetti



relativamente a Jrobots, uno dei tanti cloni che sono comparsi sulla scena e che conosco bene perché... l'ho sviluppato io (per un elenco nutrito di cloni più o meno fedeli si può visitare <http://programminggames.org/>).



### Jrobots... a multitasking multiplayer time-sharing programming game

L'ispirazione che ha portato a **Jrobots** è sicuramente nata da Crobots, ma c'è voluta anche l'esplosione del World Wide Web e la comparsa del linguaggio Java per portare a compimento l'opera. Infatti, anche se l'idea era nell'aria fin dal 1995, ho dovuto aspettare il 1999 perché la tecnologia diventasse matura per ciò che avevo in mente: robot programmati in Java, ciascuno eseguito in un suo thread all'interno di un browser.

Jrobots possiede alcune caratteristiche che lo distinguono da altri cloni.

Innanzitutto, i tornei (che si sono svolti con cadenza mensile per un decennio, dal 1999 al 2009) avvenivano letteralmente online. Come ho già detto, i giocatori programmavano il loro robot in Java, utilizzando gli strumenti di sviluppo originali della SUN, gli stessi con cui si realizzavano le prime applet Java, e ne provavano l'efficacia in un'arena offline. Una volta soddisfatti del risultato, si eseguiva l'upload del codice (una normale classe Java) nell'arena online, dopodiché il robot si poteva cimentare contro avversari creati da altri programmatori per un intero mese.

Questo spiega il "multiplayer", ma perché nel titolo si cita il "multitasking"? Perché il codice di ogni robot era eseguito, come ho già accennato, in un thread separato. Su alcuni sistemi operativi erano thread simulati, mentre su altri erano thread nativi, a seconda dell'implementazione della Java Virtual Machine su ciascun sistema.

Resta da spiegare il "time-sharing".

L'idea che ha consentito di implementare in modo semplice il gioco, senza aver bisogno di costosi servizi di hosting, è stata quella di **far eseguire i tornei dai PC dei giocatori che si collegavano al sito per vedere le classifiche parziali del torneo mensile**. Gli incontri, infatti, non venivano eseguiti dal server (e neppure eseguiti sul computer di casa dell'organizzatore, cioè io), ma dai client, cioè nei browser dei visitatori del sito, i quali, per sicurezza, non avevano modo di sapere quali incontri consentivano di disputare col loro "tempo macchina": vedevano solo le classifiche che, man mano, venivano aggiornate.

In un mese si disputavano migliaia di incontri, senza contare i pareggi, nelle tre diverse categorie: singolo (1 vs 1), doppio (2 vs 2) e a squadre (4 squadre composte da 8 robot ciascuna, cioè 32 robots presenti contemporaneamente nella stessa arena).

Naturalmente una simile realizzazione ha comportato l'adozione di numerosi accorgimenti tecnici sviluppati nel corso degli anni, grazie anche all'aiuto dei migliori giocatori.

Un problema era costituito dalla sicurezza delle applet Java. In Crobots il codice di un robot era visibile a tutti e non avrebbe potuto in ogni caso contenere codice "malizioso". In Jrobots il codice era compilato nel bytecode Java e quindi illeggibile, perciò l'autore avrebbe potuto aprire dei popup nel browser del giocatore o reindirizzarlo verso pagine esterne. L'unica soluzione praticabile è consistita nel proibire l'uso di determinate classi Java, che potevano essere facilmente individuate in modo automatico dallo script sul server che accettava i robots in upload.

Un altro problema consisteva nel determinare quale fosse il miglior algoritmo di selezione dei robot per i vari incontri. A causa del "time-sharing", che poteva essere interrotto dal visitatore senza portare a termine determinate partite, si è dovuta ideare una tecnica di selezione pesata.

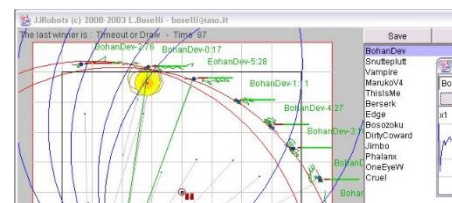
Un po' come avviene nei tornei di scacchi, dopo una serie di incontri casuali, si procedeva a far scontrare robot di punteggio simile per spargiarli e ottenere una classifica di merito il più possibile corretta.

Ma il problema principale, che poteva minare alle fondamenta il gioco, era costituito dall'ampio spettro di situazioni in cui il codice dei robot girava. Le classi Java erano eseguite in diversi browser, su differenti implementazioni della virtual machine, su diversi sistemi operativi e su macchine con processori più o meno potenti. Un robot forte in un contesto avrebbe potuto rivelarsi inefficace in un altro (un problema che in Crobots non si poteva verificare per la virtual machine indipendente dal sistema che la eseguiva).

La soluzione venne individuata realizzando un clock virtuale. In pratica, un ulteriore thread, oltre a quelli dei robot, eseguiva del codice di riferimento che scandiva il tempo delle simulazioni. Ciò ha reso più omogeneo il comportamento dei robot su macchine differenti.

Restava solo un ultimo possibile problema: codice "malizioso" avrebbe potuto eseguire cicli che non richiama alcuna funzione del gioco o allocare array su array accaparrandosi tutta la memoria (per entrambe le tecniche, l'effetto era quello di "congelare" il browser e l'esecuzione dell'incontro).

Alla fine conclusi che un tale comportamento antisportivo veniva evidenziato dalle classifiche (i robot "dannosi" mostravano un basso numero di incontri disputati) e da un semplice controllo nell'arena offline. Perciò, individuato il colpevole, lo si poteva tranquillamente bannare. Comportamenti di questo tipo si sono verificati, ma molto meno di quanto temessi.



### Il gioco nel dettaglio

Dopo questa lunga parentesi estremamente tecnica e per molti di certo noiosa, torniamo al gioco vero e proprio. Su YouTube è disponibile un video introduttivo sull'argomento:

<http://youtu.be/lyqGTuWovrY>.

Qui di seguito, invece, entro nei dettagli tecnici del gioco.

Il campo di battaglia di Jrobots è costituito da un quadrato di 1 Km di lato. Quando un robot urta le pareti, subisce 2 punti di danno su un totale di 100 e il motore si arresta. Non è la cosa peggiore che possa capitare, ma non è neppure la migliore, perché quando un robot raggiunge i 100 punti di danno, viene disabilitato e perciò è meglio evitare gli urti con le pareti (è possibile farlo mediante una specie di semplice GPS).

I robot hanno un motore e possono muoversi in ogni direzione. La velocità massima dei robot è 30 m/s, cioè 100 Km/h, e l'accelerazione è di 5 m/s<sup>2</sup>. Questo significa che un robot necessita di sei secondi per raggiungere la velocità massima.

Quando il motore ha lo 0% di potenza, il robot si ferma con una decelerazione di 5 m/s<sup>2</sup>, mentre il 100% di potenza fornisce gradualmente la velocità massima. Quando un robot urta le pareti, il motore scende allo 0% di potenza e la velocità si riduce a zero immediatamente.

I robot hanno un cannone che spara missili. Il robot può puntare il cannone in qualsiasi direzione e può sparare tutti i missili che vuole. C'è solo il limite del tempo di ricarica di un secondo.

I missili possono raggiungere una distanza massima di 700 metri e hanno una velocità di 300 m/s, così ci vogliono 2,33 secondi per raggiungere la distanza limite. La velocità del missile non dipende da quella del robot. Quando un missile esplode, danneggia tutti i robot nelle vicinanze. I punti di danno dipendono dalla distanza dell'esplosione: 5 metri per 10 punti, 20 metri per 5 punti e 40 metri per 3 punti. Perciò, se un robot spara un missile ad una distanza inferiore ai 5 metri, si procura da solo 10 punti di danno e quindi è meglio evitarlo.

Il robot ha uno scanner che gli permette di trovare gli altri robot. Perlustra il campo di battaglia con un'apertura da 1 a 20 gradi. Più piccolo è l'angolo, migliore è la risoluzione. Con la scansione dell'arena, lo scanner determina la distanza del robot più vicino (che può essere amico o nemico) o un valore nullo

se nessun robot viene individuato nel raggio d'azione.

### L'evoluzione delle tattiche

Con una struttura così semplice non sembra che sia possibile un'ampia varietà di tattiche. Invece, nel corso degli anni, si è potuto assistere a una notevole evoluzione. Da KillerBees di Walter Nisticò (il primo robot a vincere tornei in tutte e tre le categorie) a IonStorm di Alan Lund (che ha primeggiato per anni), le tecniche sono state le più disparate.

Nelle tre modalità di gioco (singolo, doppio e a squadre), la più difficile è quella a squadre: bisogna colpire i nemici evitando il fuoco amico. Inutile dire che nei primi tornei, dove i giocatori si concentravano sui più semplici singoli, si assisteva spesso a una sparatoria indiscriminata, dove le squadre più accorte prevalevano facilmente.

All'inizio i robot sparavano la loro cannonata dove rilevavano il nemico (o, nel peggiore dei casi, un amico che non riconoscevano come tale), ma il missile impiegava un certo tempo per giungere a destinazione e, nel frattempo, il nemico si era spostato. Una prima innovazione è stato l'algoritmo di calcolo che prevedeva la posizione in cui si sarebbe trovato il nemico all'arrivo del missile, per correggere preventivamente il tiro. Occorrevano due letture consecutive della posizione per determinare la velocità del nemico e prevedere le posizioni successive.

Una volta ideato l'algoritmo, la contromisura consistette nel cambiare spesso direzione di marcia e comparvero robot che, attraverso tecniche di analisi dei segnali, cercavano di prevedere i pattern irregolari. Insomma, un'escalation della competizione che non avrei creduto possibile in un gioco con così poche variabili.

Lo stesso si è osservato nelle tattiche di squadra. Come ho già detto, all'inizio tutti sparavano a tutti, amici e nemici. Poi si è capito come comunicare la posizione ai compagni di squadra.

A quel punto, si è visto un po' di tutto: c'erano squadre che si radunavano negli angoli più vicini per bersagliare da varie direzioni i malcapitati rimasti al centro; c'erano robot

che, individuato un nemico, ne comunicavano la posizione ai compagni e, quindi, lo sfortunato di turno finiva per essere bersagliato a ripetizione e terminato rapidamente; c'erano squadre che percorrevano i lati dell'arena avanzando in fila indiana come una falange ben organizzata e così via (per una sintesi delle tattiche, è disponibile [un gameplay su YouTube relativo ai robots più avanzati](#)).

Insomma, ho visto cose che voi umani non potete nemmeno immaginare: per un decennio, centinaia di robots si sono affrontati nello spazio di un chilometro quadrato con un impegno e un accanimento degno di migliori cause.

Ne potete leggere la storia nella "Hall of Fame" di Jrobots sul sito e da lì potete scaricare l'arena offline per provare voi stessi l'ebbrezza della sfida.

Ma la storia dei miei "programming games" non termina con Jrobots. Intorno al 2005, vent'anni dopo il Crobots di Tom Poindexter, è comparso **Gun-Tactyx**, "a Crobots-like game, with Quake3-style graphics"... ma questa è un'altra storia.

### Link utili

Su YouTube:

[Jrobots 01 – Gameplay \(robot di base\)](#)

[Jrobots 02 – Gameplay \(robot avanzati\)](#)

Il sito ufficiale di Jrobots è <http://jrobots.sf.net> (gli scontri online non funzionano per mancanza di manutenzione, però è ancora possibile scaricare l'arena offline).

Il sito italiano (ma oserei dire mondiale) di riferimento per Crobots gestito da Maurizio Camangi è: <http://crobots.deepthought.it/>

L'attuale sito ufficiale del Crobots di Tom Poindexter è: <http://github.com/tpoindex/crobots/>.

Sito originale dell'autore di Crobots: <https://tpoindex.github.io/crobots/>

# B.B.S - Ritorno al passato

di Marco Pistorio e Dante Profeta

Oggi riprendiamo una discussione iniziata nello scorso numero, dove spiegavo come potersi collegare ad un BBS sfruttando VICE.

Affrontiamo però la cosa da un punto di vista diverso.

Intanto riprendiamo alcuni concetti dall'articolo sul numero scorso, concetti che ci serviranno per mettere meglio a fuoco la tematica.

*(...) Un BBS (acronimo di Bulletin Board System) è un computer che utilizza un particolare software per permettere ad utenti esterni di connettersi ad esso attraverso la linea telefonica, dando la possibilità di utilizzare funzioni di messaggistica, file sharing centralizzato, ma anche semplici giochi di ruolo testuali, i M.U.D. (Multi Users Dungeons). Questo sistema nasce negli anni '70 ed ha costituito il fulcro delle prime comunicazioni telematiche amatoriali. I movimenti che diedero vita ai concetti di shareware e software libero sono nati nei BBS, attraverso lo scambio di software non protetto da limitazioni di copia, come per esempio PKZIP e le demo di Wolfenstein 3D e Doom (giochi divenuti celebri proprio grazie alla loro diffusione libera). Per gestire o utilizzare un BBS erano necessarie conoscenze tecniche più elevate della media e attrezzature particolari, il che rendeva solitamente i BBS regno di appassionati ed esperti.*

*Per un breve periodo di tempo, intorno agli anni '90, insieme ad un amico ne gestii uno personalmente, JONICA BBS. (...)*

Partendo da questa mia passata esperienza maturata "sul campo" ho deciso di provare a mettere su un BBS tradizionale. Aggiungo per dovere di cronaca che dopo quasi trent'anni ho rimosso parecchie cose.

Tuttavia, almeno in linea di massima, sono riuscito nel mio intento.

Non è stato semplice, questo ve lo anticipo subito!

In questo articolo cercherò di raccontarvi

come è andato questo progetto, il materiale necessario, Vi suggerirò l'approccio per configurare il tutto, passo dopo passo, e spero quindi che questo articolo potrà essere utile a qualcuno per ripercorrere le mie stesse "orme" ☺.

Ho cercato di mettermi nelle stesse condizioni di quando operavo agli inizi degli anni '90, per fruire di una esperienza quanto più "autentica" possibile.

Pertanto sono partito lavorando su un 486 con 16 Mbytes di RAM, disco fisso da 500 Mbytes, unità Floppy da 3 pollici e mezzo e modem analogico interno ISA a 2400 baud, con a bordo MS-DOS versione 6.22

## TELIX

Ho effettuato una serie di verifiche preliminari circa la funzionalità del modem. Per fare ciò ho installato il programma Telix versione 3.51, programma che funge da emulatore di terminale che permette di gestire sessioni ANSI, trasferimenti di files mediante protocollo ZMODEM, ha un suo sistema di scripting interno ed offre altro ancora. Tuttavia mi sono limitato a sfruttarlo per determinare con certezza la porta COM associata al modem interno, l'interrupt associato, la velocità di connessione. Telix ha una utility, MODEMCFG.exe, che serve per impostare i parametri necessari a Telix per colloquiare con il modem in maniera corretta. Tramite gli interruttori DIP presenti sulla scheda modem ho ottenuto l'informazione relativa alla porta COM adoperata dal MODEM, che si è rivelata essere la COM3.

Facendo poi una serie di tentativi via MODEMCFG ho ottenuto l'IRQ (ovvero 4) tra quelli possibili ed infine, appena Telix è partito, ho verificato tramite appositi comandi AT la funzionalità del MODEM, osservando il contenuto dei vari registri.

Tramite il comando AT ho letto il corrispondente messaggio di risposta proveniente dal MODEM ("OK") e tramite un ulteriore comando comando ATA ho verificato che il modem si mettesse in

modalità di risposta, generando gli opportuni toni udibili grazie al piccolo altoparlante integrato all'interno del MODEM.

Telix parte indipendentemente dal fatto che i settaggi effettuati dall'eseguibile MODEMCFG siano corretti o meno. E' facile però comprendere se i settaggi siano corretti oppure no perché Telix, nel secondo (malaugurato) caso, non riesce ad instaurare alcuna comunicazione con il modem.

Per chi fosse interessato a conoscere più in dettaglio i comandi AT, o quantomeno quelli più standard, nonché il significato del contenuto dei registri normalmente adoperati nei modem, fornisco questo utile link:

[http://www.febat.com/Elettronica/Elettronica\\_comandi\\_AT.html](http://www.febat.com/Elettronica/Elettronica_comandi_AT.html)

Circa il download di Telix non dovrete aver problemi nel trovarlo in rete, considerando tra l'altro la sua grande popolarità negli anni '90. Io ho scaricato da questo link ma ce ne saranno sicuramente molti altri altrettanto validi: <http://linuxmafia.com/pub/ms-dos/>

Una nota ulteriore relativa a Telix. Questa versione, la 3.51, è shareware e funziona per un periodo di valutazione di 45 giorni trascorsi i quali deve essere registrato.

Allo scopo di effettuare qualche test (e quindi per un utilizzo temporalmente limitato), la registrazione non sarà strettamente necessaria.

## MAXIMUS - download ed installazione pacchetto software

Veniamo ora al "pezzo forte" della nostra chiacchierata ovvero l'applicazione che ci permette di gestire un BBS. Perché ho scelto MAXIMUS? I motivi sono molteplici.

E' personalizzabile nei minimi dettagli, offre notevoli potenzialità (alcune delle quali le scopriremo insieme leggendo questo articolo), è esattamente il programma che adoperai negli anni '90 ed era considerato già allora un programma estremamente

affidabile, robusto ed avanzato per gestire efficacemente un BBS.

Ulteriore punto a suo favore, ultimo ma non ultimo in termini di importanza, ne è stato rilasciato anche il codice sorgente, anche se non viene mantenuto da parecchio.

Il software l'ho scaricato da questi due link:

[http://software.bbsdocumentary.com/IBM/DO  
S/MAXIMUS/max300c.zip](http://software.bbsdocumentary.com/IBM/DOS/MAXIMUS/max300c.zip)

[http://software.bbsdocumentary.com/IBM/DO  
S/MAXIMUS/max300r.zip](http://software.bbsdocumentary.com/IBM/DO<br/>S/MAXIMUS/max300r.zip)

e servono entrambi i files contenuti nei due archivi zippati.

Il primo, max300c.zip, contiene l'installer ed i cosiddetti eseguibili comuni, mentre il secondo, max300r.zip contiene eseguibili MSDOS.

Esiste un file ulteriore, max300p.zip che contiene gli eseguibili per OS/2, sistema che però non tratteremo e che comunque non serve per procedere con l'installazione di MAXIMUS sotto MSDOS. L'installazione è abbastanza semplice, parte lanciando l'eseguibile INSTALL.EXE contenuto in max300c.zip.

Tenete presente però che verrà richiesta la porta COM collegata al MODEM nonché la sua velocità,

Consiglio caldamente di saltare a piè pari la configurazione del BBS relativamente alla rete Fidonet, scegliendo NO alla relativa domanda in merito alla appartenenza del BBS a tale rete.

I motivi sono essenzialmente due: il primo è che la rete Fidonet è inattiva ormai da tempo quindi non avrebbe senso effettuare tale configurazione. Il secondo è che la configurazione del BBS come nodo della rete Fidonet è una operazione laboriosa ed, alla luce di quanto detto nel precedente punto, è del tutto inutile al giorno d'oggi.

Faccio una breve parentesi in merito a Fidonet.

Tali informazioni non sono indispensabili per l'installazione e/o alla configurazione del nostro BBS.



*Finestra "Monitor di stato Maximus". Il BBS è in attesa di chiamate*

Fidonet era una rete molto diffusa a cavallo degli anni '80-'90, anche da noi in Italia.

Ogni BBS appartenente alla rete Fidonet è inquadrata in una struttura gerarchica ad albero e possiede un indirizzo numerico su base geografica: i BBS sono dei nodi della rete, mentre gli utenti che accedono direttamente utilizzando gli standard Fidonet sono point, cioè foglie dell'albero.

Ogni BBS Fidonet riceve e spedisce messaggi solo per via gerarchica, cioè solo a BBS (nodi) di livello superiore (uplink) o inferiore (downlink) al suo, e mai direttamente fra pari livello tranne che per i BBS di livello più alto.

Ogni utente ed ogni BBS del sistema Fidonet ha un indirizzo interno che permette di inviare matrix (o netmail), la versione Fidonet della posta elettronica.

I messaggi, sia pubblici che privati, potevano essere locali, ovvero confinati nell'ambito del nodo di appartenenza dell'utente oppure globali. In quest'ultimo caso i messaggi venivano replicati tra i vari nodi in base alla struttura gerarchica della rete Fidonet facendo sì che questi messaggi fossero leggibili in tutti i nodi Fidonet, nel caso di messaggi pubblici.

Nel caso di messaggi privati veniva raggiunto il nodo al quale era collegato l'utente destinatario del messaggio.

Se per qualsiasi motivo un nodo veniva meno,

anche gli altri nodi a livello inferiore nonché tutti i relativi points diventavano isolati perché non più raggiungibili.

E' l'esatto contrario di quanto avviene oggi in INTERNET. Nella rete INTERNET infatti le comunicazioni continuano a viaggiare normalmente anche nel caso in cui vengano meno alcuni nodi in quanto il percorso dei pacchetti di dati, dalla partenza alla destinazione, è determinato in maniera dinamica.

Per ulteriori approfondimenti: <https://it.wikipedia.org/wiki/FidoNet>

#### **FOSSIL (prerequisito per il funzionamento di Maximus)**

Affinchè il sistema funzioni è necessario soddisfare un prerequisito importante, un driver FOSSIL.

FOSSIL è un protocollo standard per consentire la comunicazione seriale per i programmi di telecomunicazione con il sistema operativo MS-DOS. FOSSIL è l'acronimo di Fido Opus SEAdog Standard Interface Layer.

Tramite FOSSIL si ottiene una separazione logica tra l'applicazione BBS ed il sottosistema

delle telecomunicazioni. In questo modo il codice della applicazione è indipendente dal fatto che ci sia un solo modem collegato ad



*Layout grafico della videata di benvenuto dell'utente, che accoglie l'utente all'inizio della sua sessione di collegamento al BBS. Questo layout è stato realizzato tramite ACIDDRAW.*

una sola linea analogica, oppure ce ne siano due, tre etc ad esempio.

I più usati sono fondamentalmente BNU, Xoo ed OPUSCOMM.

Ho impiegato il primo perché è già incluso nel pacchetto di MAXIMUS, all'interno del file bnu170.zip

Basta scompattare il file in apposita directory (ad esempio c:\BNU) e poi richiamarlo.

Il richiamo è già previsto nel file batch RUNBBS.BAT di cui Vi parlerò successivamente.

### MAXIMUS (configurazioni iniziali)

L'installatore, lanciato con INSTALLER.EXE effettuerà tutte le operazioni necessarie per poter utilizzare MAXIMUS.

A parte le informazioni di cui abbiamo parlato (ovvero porta COM associata al modem, la sua velocità e l'appartenenza o meno alla rete FIDONET per procedere alla eventuale configurazione necessaria) non è necessario cambiare nessuno dei valori suggeriti di default, oltre al nome di base del BBS ed al nome del suo SysOp, ovvero della persona identificata come il "gestore del BBS".

Al termine della installazione è necessario riavviare il pc in quanto saranno stati

impostati dei settaggi sul file di sistema c:\AUTOEXEC.BAT

E' necessario poi entrare nella cartella di lavoro di MAXIMUS (c:\max se non è stato impostato diversamente) e lanciare MAXIMUS così: max -c in modo da poter registrare il primo utente ovvero il SysOp (termine che risulta dalla abbreviazione di "System Operator", ovvero Operatore del Sistema).

Ho effettuato piccoli cambiamenti nel file batch che fa partire il BBS, ovvero il file

RUNBBS.BAT presente nella cartella di lavoro di MAXIMUS.

Ho aggiunto il flag /f (fast ANSI) alla riga dove viene richiamato il driver FOSSIL (bnu /f).

Ho commentato la riga max -w -p%port% -b38400 -s38400 ivi presente, antepoendo un comando REM a sinistra della stessa istruzione ed ho aggiunto subito dopo un semplice comando: max -w

Ciò perché preferisco non rendere fisso il baud rate per il collegamento e delego al modem il compito di negoziare la velocità di connessione.

L'opzione -w è indispensabile per far mettere e Maximus in attesa di connessioni.

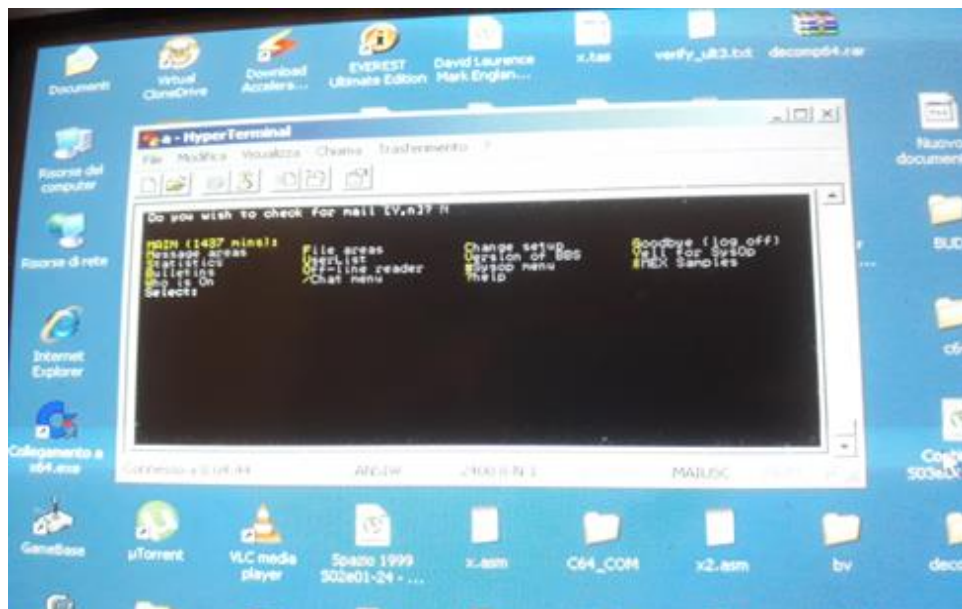
Suggerisco di adoperare sempre tale file batch per far partire il BBS.

Grazie alla esecuzione di questo file batch (RUNBBS.BAT) il BBS ripartirà in automatico dopo ogni disconnessione di un utente.

Osservate bene la finestra riprodotta.

In particolare, le funzioni Alt+X che chiudono il BBS, oppure l'altrettanto utile funzione Alt+K che permette di loggarsi al BBS sfruttando la tastiera locale oppure ancora la funzione Alt+I per reiniziare il modem.

**Nota bene:** in caso di irregolarità con il driver FOSSIL e/o con il colloquio con il modem non vedrete nulla nel riquadro "Modem



*Menù principale di MAXIMUS*



Logon di Dante Profeta

Responses" quindi se dovesse presentarsi tale circostanza sapete già cosa controllare.

### MAXIMUS (personalizzazione BBS - parte I)

Veniamo ora alla personalizzazione di Maximus, aspetto certamente interessante.

In particolare, osserviamo uno dei meccanismi di base di MAXIMUS che ci fanno capire la complessità e nel contempo la sua flessibilità nonché la sua architettura avanzata, almeno relativamente al periodo temporale al quale si riferiscono i BBS.

Iniziamo con il dire che le pagine proposte da Maximus sono ottenute mediante compilazione, attraverso un apposito compilatore (M.E.C.C.A. ovvero Maximus Embedded Command Compiler) di files con estensione .mec

Tale compilazione produce un file .bbs per ogni file .mec compilato.

Osserviamo uno stralcio del contenuto ad esempio del file WELCOME.MEC che si trova all'interno della subdirectory \MISC di c:\MAX

```
[cls onexit]Misc\Bulletin

[lightgreen on
blue]+-----+ [white]

[lightgreen on
blue]| [FG yellow>Welcome! [FG
lightgreen] | [white]
```

```
[lightgreen on
blue]+-----+ [white]

[gray] [black]

[yellow]Thank you for calling
[sys_name], [first].
```

Come potete notare vi sono una serie di informazioni che riguardano azioni da compiere ([cls on exit] ovvero cancella schermo all'uscita dalla pagina), file .mec da aprire subito dopo (\MISC\Bulletin), attributi relativi al colore del testo o del fondo dello schermo ([white] piuttosto che [FG yellow]) ed ancora campi riempiti con informazioni prelevate dal sistema stesso ([sys\_name], il nome del BBS oppure [first], il nome dell'utente collegato) ed infine delle semplici scritte costanti (Welcome! oppure le varie barre orizzontali e verticali)

Tutte queste informazioni vengono messe insieme e preparate dal compilatore MECCA ottenendo un file binario incomprensibile ma eseguito direttamente da MAXIMUS.

Questa scelta dal punto di vista della sua architettura ci fa comprendere una cosa essenziale: quando si modificano i files .mec, bisogna sempre ricompilarli.

Come si compilano? Nel caso di welcome.mec basta digitare: mecca welcome.mec

E se volessi creare una bella pagina di benvenuto con un logo accattivante

graficamente, come quelli che siamo abituati ad incontrare nei BBS?

E' chiaro che ottenere tali loghi editando direttamente il relativo file .mec è una cosa estremamente difficile. Richiederebbe la perfetta conoscenza dei vari tags, richiederebbe certamente molto tempo, molta pazienza e molta abilità. Si può fare qualche modifica "al volo" della parte testuale magari, ma non più di tanto.

C'è una strada decisamente più agevole!

Un editor di files ANSI (American National Standards Institute) ci permetterà di creare tutta la parte grafica della nostra pagina.

Successivamente, mediante apposita utility fornita con Maximus, (ans2mec.exe) convertiremo il file così ottenuto nel formato .mec

A questo punto potremo poi incorporarlo con un eventuale copia ed incolla nella pagina dove vogliamo introdurre la grafica da noi creata.

Come si adopera l'utility ans2mec.exe?

Basta digitare dalla directory di lavoro di Maximus:

```
ans2mec
c:\editor\file_da_convertire.ans
c:\editor_file_convertito.mec
```

Ovviamente occorre digitare il corretto percorso dell'editor da noi utilizzato nonché il nome del file da convertire e quello che vogliamo ottenere.

Ho ottenuto velocemente discreti risultati con l'editor ACIDDRAW che potrete scaricare a questo indirizzo:

[ftp://archives.thebbs.org/ansi\\_utilities/ad%2dosa.zip](ftp://archives.thebbs.org/ansi_utilities/ad%2dosa.zip)

### MAXIMUS (personalizzazione BBS - parte II)

Analogamente al file welcome.mec, vi sono altri files presenti nella sottodirectory /MISC della cartella di lavoro di Maximus (c:\max) che è possibile personalizzare.

Uno di questi files è ad esempio logo.mec che contiene di base solo delle informazioni molto scarse relativamente al BBS, che si trova anch'esso dentro /MISC. Per questo file però,



*La postazione AMIGA per i test di collegamento allestita da Dante Profeta*

cioè logo.mec, consiglio di non aggiungere elementi grafici e di modificarne il contenuto testuale solo quanto basta per identificare il Vostro BBS.

Suggerisco di modificare anche il file bulletin.mec che contiene i bollettini, comunicazioni che vengono presentate automaticamente all'inizio di ogni sessione di collegamento di ciascun utente.

Se non volete implementare una veste grafica particolare potrete agevolmente editare il contenuto di quest'ultimo file .mec tramite l'editor di base di MS-DOS, ovvero EDIT.

Sarà necessario sicuramente personalizzare anche le aree messaggi e files.

La configurazione avviene editando opportunamente i due files filearea.ctl e msgarea.ctl che sono presenti nella directory di lavoro di Maximus ovvero c:\max

Vi mostro ora come gestire le aree files.

I files del BBS sono ospitati nella subdirectory /FILE e sono suddivisi secondo una nomenclatura basata sul contenuto di filearea.ctl del BBS.

Se non vengono effettuate modifiche, tali subdirectory saranno /UNCHECK per tutti i nuovi files uploadati in BBS e non ancora controllati dal responsabile del BBS, /NEWUP per i files nuovi, /UTIL per i programmi di utilità generale ed infine /MAX per i

programmi collegati a Maximus in linea generale.

A titolo di esempio, riporto la definizione della nuova area files che ho configurato:

```
FileArea 4
        ACS          Demoted
        Desc         Uscite di
Retromagazine
        Download
        File\Retromag\
        Upload
        File\Uncheck\
End FileArea
```

Tali modifiche sono state inserite editando il file relativo alle aree files presenti in BBS, cioè il file filearea.ctl

Osservare la sintassi adoperata.

E' molto simile, per certi versi, alla configurazione dei server FTP sotto Linux (diritti di accesso alla directory, nome mappato e percorso reale della directory etc). Chi ha avuto esperienze in merito ne converrà con me.

Descrivo brevemente i vari parametri:

**ACS** indica il controllo di accesso. Demoted è un livello basso, non privilegiato come ad esempio SysOp. Ciò farà sì che l'area sarà accessibile a tutti gli utenti.

**Desc** contrassegna il nome dell'area che sarà "Uscite di RetroMagazine".

**Download** invece serve per individuare la sottodirectory dove si troveranno i files da poter scaricare. In questo caso sarà \max\File\Retromag

**Upload** invece contrassegna la sottodirectory dove si troveranno i files che verranno caricati dagli utenti nel momento in cui si troveranno all'interno di questa area file 4.

L'impostazione è in questo caso \max\File\Uncheck ed è identica per TUTTE le aree files.

Tutti questi parametri di configurazione si trovano racchiusi tra i tag FileArea 4 ed End FileArea.

Il numero dell'area file sarà quindi "4" mentre il suo nome, come già specificato, sarà "Uscite di RetroMagazine"

Per quanto concerne le aree messaggi, le operazioni sono similari.

Le aree sono definite all'interno del file msgarea.ctl

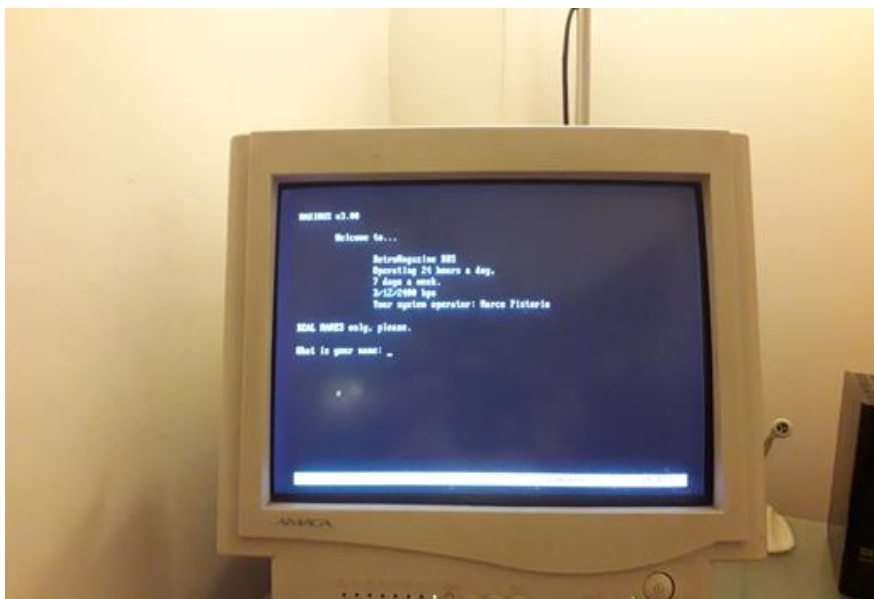
Per ogni area definita esisteranno tre files, creati da Maximus, all'interno della subdirectory /MSG aventi lo stesso nome dell'area ed estensione rispettivamente .sql, .sqi ed infine sqd.

Riporto la definizione della nuova area messaggi che ho configurato:

```
MsgArea 3
        ACS          Demoted
        Desc         Public      Messages
RetroMagazine-related
        Style        Squish Local Pub Pvt
        Path         Msg/PubRM
        Renum Max    300
End MsgArea
```

Faccio dei brevi cenni relativi ai parametri anche in questo caso.

**ACS** indica il controllo di accesso. Demoted è un livello basso, non privilegiato come ad



**Testo di presentazione del BBS (file logo.mec)**

esempio SysOp. Ciò farà sì che l'area sarà accessibile a tutti gli utenti.

Style contiene alcuni parametri: Squish (che è il formato di memorizzazione dei messaggi nonché il programma di posta che si adoperava su FidoNet), Local indica che si tratta di un repository locale, Pub che è possibile creare in tale repository messaggi Pubblici ed infine Pvt indica che è possibile creare anche messaggi privati nello stesso repository. Renum Max 300 indica il numero massimo di messaggi che saranno contenuti nell'area, superato il quale i messaggi inizieranno ad essere ricoperti.

Tutti questi parametri di configurazione si trovano racchiusi tra i tag MsgArea 3 ed End MsgArea.

Il numero dell'area messaggi che stiamo definendo sarà quindi "3" e la sua descrizione sarà "Public Messages RetroMagazine-related", come già specificato.

Tali operazioni richiederanno un minimo di attenzione e con l'esperienza diventerà intuitivo e facile interagire con questi files di configurazione di Maximus. Considerate che, sebbene operazioni fondamentali, una volta creata la struttura di base delle aree messaggi e files le modifiche dovrebbero diventare via via minime.

Dopo aver effettuato modifiche ai files .ctl è necessario ricompilare i files di controllo del sistema Maximus.

Solo così renderete attive tali modifiche. Si adopera allo scopo un compilatore apposito, fornito con il sistema, in maniera abbastanza analoga a quanto si fa con i files .mec incontrati in precedenza.

Cambia però il compilatore. Per i files .ctl il compilatore è SILT (Maximum Control File Compiler).

Per effettuare una ricompilazione di tutti i files di controllo basta digitare: `SILT Max`

Rinominare e conservare i files nella loro forma originale PRIMA di modificarli fa vivere più sereni :)

Ad esempio, per conservare il file msgarea.ctl prima di modificarlo, potreste adoperare il comando MSDOS: `copy msgarea.ctl msgarea.old`

Ricordate: Sia i files di configurazione (.ctl) sia quelli relativi alle pagine del BBS (.mec) devono essere ricompilati ad ogni loro modifica altrimenti tali modifiche non saranno applicate.

Durante i test di collegamento, effettuati grazie alla infinita pazienza dell'amico e collega redattore Dante Profeta (che ringrazio per essersi "immolato" per questa causa) è stato rilevato un problema nella stringa di inizializzazione del modem, che non permetteva il rilascio della linea da parte del modem al termine del collegamento dell'utente da remoto.

Di conseguenza, armandomi anch'io di molta pazienza ho rivisto i settaggi necessari per far funzionare il tutto correttamente, aiutandomi un po' anche con Internet.

Al posto del comando: `at&b1hoso=0` che era impostato originariamente, ho adoperato un comando un po' più elaborato che è il seguente:

```
ath0&c1&d2v1X4S0=0S7=55S11=60.
```

Per i dettagli circa i parametri del comando AT vi rimando al link già citato in precedenza ([http://www.febat.com/Elettronica/Elettronica\\_comandi\\_AT.html](http://www.febat.com/Elettronica/Elettronica_comandi_AT.html))

E' stato necessario editare il file max.ctl nella "Equipment Section", e più precisamente la voce Init inserendo la nuova stringa di inizializzazione del modem. Poi ho ricompilato tutto tramite apposito comando citato in precedenza silt max

Tale settaggio è valido per il mio modem. Vi invito ad effettuare con attenzione i test necessari al fine di trovare il corretto settaggio anche per il Vostro modem, nel caso che la mia stringa di connessione non dovesse funzionare per Voi.

Ho anche effettuato dei test con un secondo PC WinXP ed HyperTerminal provvisto di modem. E' inusuale ma è comunque possibile far colloquiare due modem analogici collegati a 2 PC diversi sfruttando la medesima linea (anche ADSL se correttamente filtrata). Chi volesse approfondimenti al riguardo può scriverci,

come di consueto, alla mail della redazione: [retromagazine.redazione@gmail.com](mailto:retromagazine.redazione@gmail.com)

Tenete presente che sarà possibile creare sia messaggi pubblici che privati nelle varie aree messaggi. Tuttavia non sarà possibile sfruttare le funzionalità che prevedono il collegamento ad altri nodi dal momento che la rete Fidonet è ormai in disuso e non è stata quindi configurata.

Non ho trattato l'argomento packers (programmi per compressione dei files) né l'argomento mailers (applicazioni che processano le aree messaggi e la posta elettronica tipo SQUISH), perché ho supposto che il nodo creato non si colleghi a nessun altro nodo e sia quindi isolato.



Ci sarebbero molti altri argomenti da trattare ad esempio MEX e MECCA, i due linguaggi di programmazione mirati alla personalizzazione del BBS, oppure una trattazione dettagliata del contenuto del file max.ctl, oppure ancora i meccanismi per interfacciare applicazioni con le quali gli utenti del BBS possono interagire ma evito di dilungarmi ulteriormente ed invito tutti coloro che intendono approfondire ad iniziare a studiare il "Manuale delle Operazioni di Sistema di Maximus", purtroppo disponibile solo in lingua inglese, che potrete scaricare qui:

[http://software.bsdocumentary.com/IBM/DO  
S/MAXIMUS/max300.txt](http://software.bsdocumentary.com/IBM/DOS/MAXIMUS/max300.txt)

### Uno sguardo al presente

Oggi ha senso utilizzare i BBS? Internet oggi offre molto di più di quanto offrivano i BBS. La risposta quindi potrebbe essere no. Tuttavia ha senso conoscere i BBS e riconoscere loro l'importanza che meritano per essere stati i pionieri in un'era in cui le telecomunicazioni facevano i loro primi passi e ci si domandava quali vantaggi potessero provenire dalle telecomunicazioni e come poter sfruttare queste nuove tecnologie, prima che Internet entrasse nelle case di tutti noi.

E' mia convinzione che conoscere il passato può essere di aiuto sia nel comprendere il presente che nell'immaginare il futuro.

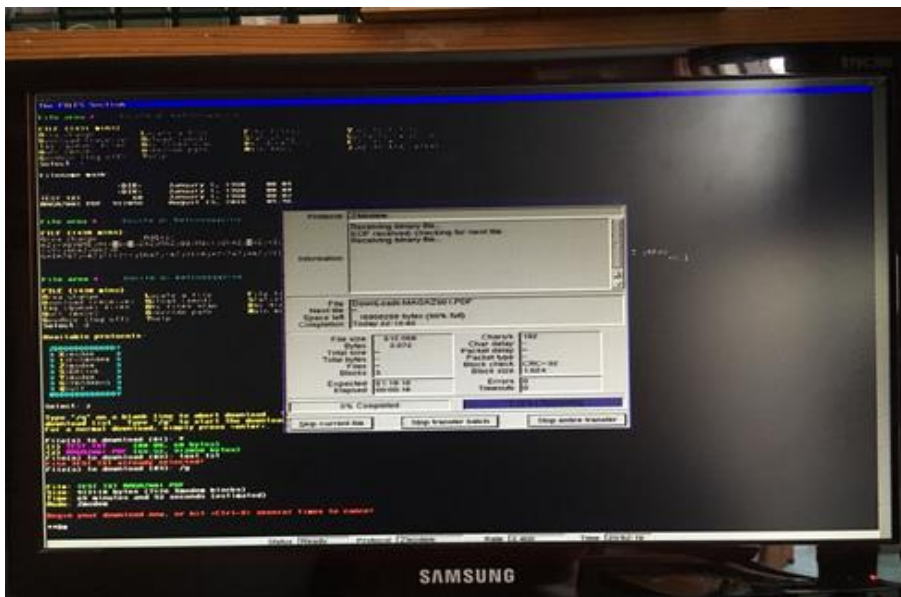
Oggi comunque, a differenza che nel passato, esistono dei software di BBS che ci permettono di evitare l'uso di una linea analogica dedicata. E' possibile ad esempio gestire connessioni multiple via Telnet, "emulando" il driver FOSSIL di cui Vi ho accennato all'inizio.

Tratteremo forse in un prossimo articolo di questi software di nuova concezione per creare BBS più "moderni".

### Adesso tocca a Voi (forse):-D

Credo che la chiacchierata sia stata sufficientemente ampia pertanto Vi ringrazio per avermi seguito fin qui.

Spero che questo articolo Vi sia stato utile quantomeno per comprendere la filosofia alla base dei BBS. Sono riuscito a risvegliare il "SysOp" che si celava dentro di Voi?



*Finestra di Download durante lo scaricamento di un file dal BBS*

Io ho un piacevole ricordo dei BBS, forse anche perché si tratta di ricordi legati ad un periodo in cui ero molto giovane, ed è stato piacevole guardare le stesse videate, effettuare le stesse operazioni e sentire i "fischi" del modem che mi erano tanto familiari una volta.

Ci fu un periodo in cui il collegamento ai BBS era il modo più semplice ed immediato per trovare il software o il gioco che ci interessava ottenere, ed era anche un potente strumento per ottenere informazioni.

Ricordo ad esempio che ebbi un problema con un Commodore 16. Per trovare il corretto valore capacitivo di un condensatore di cui conoscevo solo il codice impresso sulla mainboard mi bastò formulare una domanda in un'area messaggi pubblica. La risposta la lessi un paio di giorni dopo.

Oggi con Internet il problema si risolverebbe molto prima e molto facilmente ma questo mio aneddoto risale ad un momento in cui il protocollo http non era ancora nato e precedente quindi alla sua diffusione a livello planetario. Ma, prima che vi lanciate a capofitto in una avventura come questa, è bene che teniate presente alcuni "inconvenienti" che potrebbero nascere dalla gestione di un BBS.

Ad esempio, se ci fossero violazioni dei diritti di copyright relativamente ai files pubblicati, la prima persona ad essere raggiunta è il SysOp che può poi rivalersi sull'utente che ha

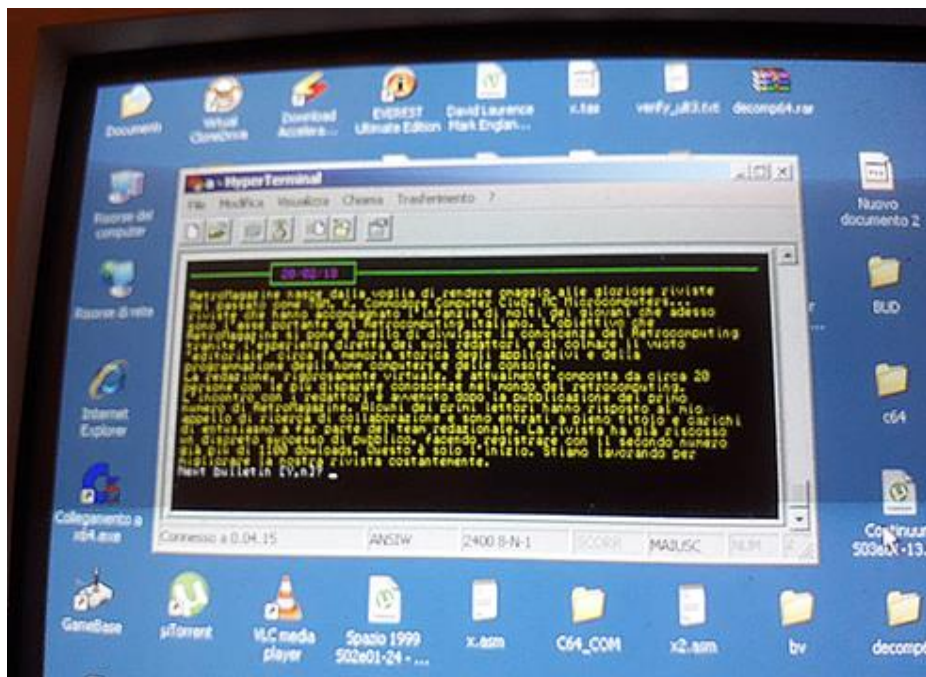
caricato i files ma, in primis, resta lui il responsabile, se non per diffusione di materiale coperto da diritti d'autore quantomeno perché non è stato diligente nel verificare i contenuti presenti nel BBS che gestisce. Stesso ovviamente dicasi per eventuali commenti spiacevoli e/o calunniosi nelle aree messaggi. Facciamo ora alcune considerazioni ulteriori per quanto concerne la posta elettronica veicolata tramite BBS.

Tutte le mail venivano memorizzate sul BBS di appartenenza dell'utente. Era possibile poi scaricarle sul proprio PC con appositi programmi, tuttavia lo spazio impegnato primariamente era sul disco fisso del BBS.

La consegna della posta tra utenti appartenenti allo stesso nodo era istantanea ma le operazioni non erano altrettanto veloci quando la mail doveva arrivare ad un utente iscritto ad un altro nodo.

Ciò perché la posta "esterna" veniva replicata di nodo in nodo, secondo la struttura gerarchica Fidonet (vedi paragrafo relativo alla rete Fidonet), fino all'arrivo alla sua destinazione. Questo comportava un certo numero di "scambi" tra i nodi coinvolti, ciascuno dei quali richiedeva un certo tempo. Il sistema funzionava, almeno in linea generale, ma con lentezza.

I collegamenti tra nodi erano pianificati per avvenire in giorni e ad orari ben precisi e spesso interessavano nodi che venivano raggiunti tramite chiamate extraurbane. I



Uno dei "Bollettini" che viene presentato subito dopo il login

giorni scelti ed i relativi orari per tali collegamenti venivano impostati con attenzione per cercare di minimizzarne i relativi costi.

I gestori dei BBS dovevano essere molto coscienti (forse troppo) e dovevano anche essere disposti a sostenere alcune spese. Considerando che i BBS erano gestiti per la maggior parte da appassionati che molto spesso non ricavano alcun compenso da tale attività, è chiaro che l'entusiasmo via via sia venuto meno un po' in tutti loro, o quasi.

In definitiva possiamo concludere che i BBS non siano tramontati soltanto per motivi squisitamente tecnologici. Ed il "ritorno al passato" oggettivamente non è fattibile né auspicabile, soprattutto se l'obiettivo che si intendesse raggiungere fosse quello di ottenere qualcosa di realmente comparabile a ciò che abbiamo oggi comodamente a disposizione grazie alla diffusione capillare di Internet nelle nostre case e nei nostri devices ovvero telefonini, tablets etc.

Tuttavia, per scambi di volumi ridotti di files, di informazioni tecniche, oppure ancora per fare quattro chiacchiere in chat con un vecchio amico in stile "retro", magari collegandosi a BBS raggiungibili via TELNET che permettano connessioni multiple senza penare troppo e senza costi di linee

analogiche dedicate o altro, può ancora avere un senso, almeno secondo me.

La risposta alla precedente domanda "Oggi ha senso utilizzare i BBS?" quindi non è poi così scontata...

Ciao a tutti ed alla prossima! *Marco Pistorio*

### Tempi moderni: La telematica, il modem, le BBS (contributo di Dante Profeta)

"La telematica (termine raramente sostituito da teleinformatica) è una disciplina scientifica e tecnologica che nasce dalla necessità di unificare tra loro le metodologie e le tecniche delle telecomunicazioni e dell'informatica per integrare i sistemi di elaborazione e di trasmissione a distanza delle informazioni." – fonte: Wikipedia.

Un modo nuovo di comunicazione è apparso. Quasi improvvisamente il telefono ha acquistato un nuovo utilizzo balzando tutti in una nuova dimensione. Non solo più la voce poteva essere trasmessa a grandi distanze... ricordo ancora da bambino quando mio papà parlava al telefono con i suoi parenti d'oltreoceano, componendo il numero di telefono sul disco, facendolo roteare con precisione fino al raggiungimento del fine corsa, con un occhio sempre attento all'orologio: telefonare era costoso; si pagava a scatti telefonici, e la temporizzazione degli scatti era tabellata in base alla distanza dell'apparecchio ricevente. In città esisteva una tariffa denominata TUT, Tariffa Urbana a Tempo. Per chiamare il resto del mondo si

entrava nelle tabelle della Teleselezione, ovvero fuori dalla TUT, e qui gli scatti si facevano sempre più rapidi, e chi possedeva una piccola borchia collegata all'enorme telefono in bachelite, chiamata *contascatti*, sentiva un click periodico che avvertiva del trascorrere del tempo e del progressivo aumentare del costo della chiamata telefonica. La teleselezione intercontinentale ravvicinava gli scatti paurosamente, mettendo ansia sempre maggiore a chi aveva composto il numero.

Avevo dieci anni quando uscì il film *WarGames*, pietra miliare nella storia della cinematografia e dell'informatica di massa, e sbalordito sognavo già quali universi mi avrebbe spalancato quella meraviglia fantascientifica che era *l'accoppiatore acustico*: un fantastico apparecchio su cui si poggiava la robusta e standardizzata cornetta telefonica, affinché fischio dopo fischio catene di bit modulati entravano e uscivano dal porto<sup>1</sup> seriale dell'Home Computer.

Passò ancora qualche annetto per me: Percorrendo tutta la strada che dall'Atari 2600 mi avrebbe portato inesorabilmente a oggi, fu tappa decisiva l'acquisto del fantastico Commodore 64. Di lì a poco, a casa di un amico feci la conoscenza di una strana scatola, che presto avrebbe cambiato il corso della mia vita: il modem. Un modem a 300 Baud<sup>2</sup>.

Gli scherzi telefonici. Selezionando un numero a caso, incuranti della TUT, per fischiare al telefono di uno stupito malcapitato che urlava invano "pronto?" mentre gli venivano vomitate sequenze ripetute di handshake modulate.

Le connessioni fugaci e rade all'unica BBS Fidonet dell'area urbana, per non incappare nelle ire funeste del genitore, con la bolletta telefonica in una mano e l'altra orba, roteante come una lama di Goldrake.

Poi fu il tempo di Amiga 1000, del modem Spider 1200 e di MC-Link. Un mondo immenso, brulicante di vita virtuale. Flussi di dati e di caratteri semanticamente corretti viaggiavano avanti e indietro da casa mia a Roma, ma solo dopo le 18:00, quando i costi telefonici in teleselezione erano più bassi. Le aree di discussione tematica erano affascinanti... di più, erano un salotto dell'élite telematica del periodo, fonte di apprendimento e di scambio culturale ad altissimo livello. Persino i troll utilizzavano un linguaggio dalla punteggiatura corretta.

Moderavo l'area Assembler 68000 e le mie aree di riferimento, casa mia, erano quelle di Amiga. Prima una sola, moderata dal caro

amico Andrea Suatoni, poi anche su Amiga Programmazione, generatasi per cariocinesi e moderata dall'amico Nick!, che non immaginava certo che alcuni anni dopo avrebbe acquisito notorietà mondiale col suo vero nome, Nicola Salmoria.

Di lì a poco Itapac Easy-way mi avrebbe consentito di trascorrere intere serate in chat col NightGaunt, la Ghira Impavida, ... e altri loschi individui con cui ancora ci frequentiamo.

Adesso il modem era uno ZyXEL U-90E, che quando tutti gli altri andavano al massimo a 14400 bps, raggiungeva la stratosferica velocità di 16800 bps con protocollo proprietario, ovvero se non ti collegavi a un altro XyZEL di almeno pari caratteristiche viaggiavi a 14400 come tutti gli altri. Acquisto tuttavia oculato poiché in Commodore avevano modem ZyXEL e io, in qualità di Sviluppatore Commodore iscritto al CATS (Commodore Amiga Techincal Support) mi collegavo tramite uucp, trasformando il mio Amiga 3000 in un nodo unix, così da avere contatto diretto con i mitici Ingegneri Commodore Amiga e ricevere supporto sulle problematiche di programmazione a cui si andava incontro con i Kickstart ancora in beta e non perfettamente documentati.

\*\*\*

Nel 2018 mai avrei immaginato di rivivere il sogno della telematica degli anni '80 e '90.

Tra il lusco e il brusco ricevo un messaggio dal bravissimo amico Marco Pistorio. Non potevo credere a quello che stavo leggendo. Mi chiedeva supporto per fare dei test per una BBS che stava mettendo su un vecchio PC. Eccitato all'idea mi sono subito offerto, così in un caldo pomeriggio di agosto, in preda all'emozione ho montato sul banco il mio Amiga 3000, fedelissimo compagno d'avventura, sul cui hard disk risiede ancora il perfetto Term, scritto da Olaf 'Olsen' Barthel, un altro personaggio famoso del tempo che fu.

Il modem, adesso un U.S. Robotics Sportster da 56 Kbps. Dall'altro capo, a rispondere, una gloriosa scheda modem su slot con velocità massima di 4800 bps.

Il mio 1950 è là sul pavimento che mi fissa. Lo fisso di rimando. Saranno 25 anni che non lo accendo. Meglio non correre rischi. Vado di là e tiro fuori un lcd 16:9 Samsung dalla risoluzione strana di 1300 e qualcosa... Nulla da fare. Il software Cybervision che gestisce la Picasso II non ha questa risoluzione predefinita. Posso crearla... la creo... no!

desisto. Troppo curioso di connettermi alla BBS di Marco.

Ok, scelgo una risoluzione che renda i caratteri del Workbench abbastanza decenti da essere leggibili. Avvio Term, lo saluto. Mi scuso per tutto questo tempo. I log mi riportano indietro a metà anni '90. Ci sono tutti. Tutte le mie chat su MC-Link trascritte. Mano tremante. Non ho tempo. Ci torno dopo.

La finestra è vuota e a tutto schermo. I menù sono tutti lì sopra in attesa di essere cliccati. Che faccio? Mi muovo impacciato... folgorazione! Scrivo: "ATZ" e premo invio. Leggo: "Ok" ... WOW! Un banco di neuroni torna on-line nel mio cervello. Continuo: "ATDT0952..." e premo di nuovo invio. Un relè scatta dentro il modem e sobbalzo: "tu-tut". Sento la sequenza dei toni di chiamata. Una risposta dall'altro capo. Comincia la sequenza di fischi dell'handshak. Un brivido mi percorre. Mi batte forte forte il cuore. Ho la voce spezzata. Mi ricompongo ancora tachicardico. Marco al cellulare esulta. Esultiamo. Vedo comparire le scritte con una velocità dimenticata. Non riesco a leggere. Sono ancora emozionato. Compare il logo RetroComputer BBS, composto con i caratteri ASCII a riempire l'intera pagina. Nella mia mente recupero il logo ASCII di MC-Link e lo fondo con quello di Marco. Emozioni recenti e antiche ormai sopite riaffiorano e si fondono.

Marco mi guida tra i menù. Proviamo a trasferire un file. Spunta un menù ASCII con tutti i protocolli supportati, in primis X-Modem. Scorro con gli occhi lucidi velocemente alla ricerca di Z-Modem. Non lo trovo. Vedo Kermit e il mio cervello mi dice: affidabilità e lentezza. Marco mi parla al cellulare e mi rassicura che c'è. Mi calmo e rileggo. Ecco! Scelgo Z-Modem. Term apre una finestra piena zeppa di informazioni e statistiche sul file che sto per scaricare. Il download comincia. Conto i bytes. Dimentico la mia connessione in fibra ottica al Gigabit con OpenFiber. In quel momento il mondo creato passa solo attraverso un doppino telefonico in rame... e l'ombra di un pensiero torbido mi assale: chi mai utilizzerebbe un modem su doppino in rame oggi, in una connessione punto-punto? Penso a gente losca dai fini illeciti. La mia distorsione professionale da responsabile dei sistemi di sicurezza mi fa dire: se fossi nei panni della Polizia Postale le connessioni via modem sarebbero i primi campanelli d'allarme che farei scattare. Marco mi parla chiedendomi

altri test e mi riporta alla realtà. Dimentico tutto e proviamo ad avviare una chat... Mi manca MC-Link! Cribbio quanto mi manca...

1. *La traduzione corretta di port e' porto e non porta; così serial port andrebbe tradotto come porto seriale, parallel port come porto parallelo, e così via... molto probabilmente, in tempi ormai andati, qualche divulgatore scientifico nostrano, poco avvezzo alla lingua anglosassone, deve aver tradotto port in porta, e da allora si è continuato ad utilizzare la traduzione erronea. Qui si è voluto porre l'accento sulla questione, con la scusa di riportare l'orologio indietro al tempo del crudele misfatto.*
2. *La velocità di connessione si misurava in baud rate, che rappresenta il numero di caratteri o simboli trasmessi in un'unità di tempo, tipicamente il secondo, BPS, Baud Per Second. Da non confondere con l'unità di misura corrente che è il bps, ovvero bits per second, oppure Bps, ovvero Bytes per second.*

#### Link utili

Elenco dei comandi AT:

[http://www.febat.com/Elettronica/Elettronica\\_a\\_comandi\\_AT.html](http://www.febat.com/Elettronica/Elettronica_comandi_AT.html)

Informazioni su FidoNet:

<https://it.wikipedia.org/wiki/FidoNet>

Telnet download:

<http://linuxmafia.com/pub/ms-dos/>

Maximus download:

<http://software.bsdocumentary.com/IBM/DOS/MAXIMUS/max300c.zip>

<http://software.bsdocumentary.com/IBM/DOS/MAXIMUS/max300r.zip>

Manuale di Maximus:

<http://software.bsdocumentary.com/IBM/DOS/MAXIMUS/max300.txt>

AcidRaw download:

[ftp://archives.thebbs.org/ansi\\_utilities/ad%20do5a.zip](ftp://archives.thebbs.org/ansi_utilities/ad%20do5a.zip)

# Fantasy Console: TIC 80

di Marco Pistorio

"TIC-80 è un computer fantasy per creare, giocare e condividere piccoli giochi.

Esistono strumenti integrati per lo sviluppo: codice, sprite, mappe, editor audio e riga di comando, che permettono agevolmente di creare un mini gioco retrò. Il risultato finale sarà un file di cartuccia, che potrà essere caricato localmente ed eseguito ma anche condiviso sul web.

Inoltre, il gioco può essere inserito in un player che funziona su tutte le piattaforme più diffuse e che permette di distribuire il gioco sulla piattaforma desiderata.

Per realizzare un gioco in stile retrò, l'intero processo di creazione avviene sotto alcune limitazioni tecniche: display 240x136 pixel, 16 palette colori, 256 sprite 8x8 colori, suono a 4 canali e così via."

Questo è quanto viene riportato nel sito di TIC 80, precisamente all'indirizzo:

<https://tic.computer/>

Perché realizzare qualcosa del genere? Le ragioni che si possono individuare sono molteplici.

Innanzitutto, realizzando qualcosa di diverso dai sistemi già esistenti si salta a piè pari il problema di ottenere le licenze per utilizzare roms coperte normalmente da copyright. Poi è giusto considerare che le emulazioni, sia software che hardware (note come "simulazioni") non garantiscono una retrocompatibilità assoluta, sebbene ci si arrivi a volte molto vicini. La gestione degli "illegal opcodes" e/o di tutte le features conosciute e soprattutto quelle meno conosciute e documentate rende molto complesso, se non impossibile, il compito di scrivere una emulazione software perfetta al 100%, e ciò vale anche per le simulazioni hardware basate ad esempio su tecnologie FPGA. Realizzare qualcosa di completamente slegato dai vecchi sistemi, che giri nativamente su PC, permette di ottenere in generale migliori prestazioni e, dal momento che si tratta in effetti di un sistema nuovo, non ci si pone il problema della retrocompatibilità, a parte eventualmente una retrocompatibilità limitata alle precedenti versioni dello stesso progetto. Le limitazioni del sistema, ovvero il display 240x136 pixel, palette 16 colori, 256

sprites 8x8 colori, suono a 4 canali etc nascono ovviamente dall'obiettivo di creare una esperienza che ricordi da vicino le "vecchie glorie" che conosciamo un po' tutti noi. Il sistema è open source.

Il codice di "tic-80 tiny computer" è disponibile per i sistemi operativi più diffusi (Windows, Linux 32 e 64 bit, Mac OS X, Android) ed è possibile creare l'eseguibile dai sorgenti per altre piattaforme, ad esempio per Raspberry Pi. Le cartucce gioco vengono realizzate sfruttando gli strumenti di sviluppo integrati, e sono scritti in LUA oppure MoonScript oppure ancora JavaScript. Lo slot di memoria a disposizione è di 64Kb, interamente disponibili.

```

DIR LS      show list of files
CD          change directory
MKDIR      make directory
FOLDER     open working folder in OS
ADD        add file
GET        download file
EXPORT     export html or native game
IMPORT     import sprites from .gif
DEL        delete file or dir
CLS        clear screen
DEMO       install demo carts
CONFIG     edit TIC config
VERSION    show the current version
EDIT       open cart editor
SURF       open carts browser

Press ESC to enter UI mode
  
```

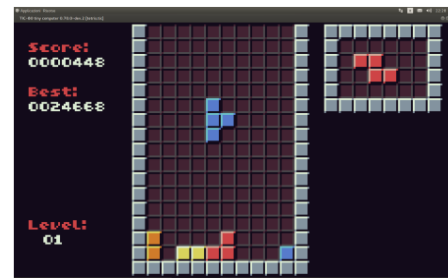
Possibilità di fare "business"? Il business, oltre che nel far pagare per ottenere il software da installare sul device desiderato, nasce anche dalla possibilità di offrire le cartucce dei giochi dietro compenso.

Come probabilmente avrete già intuito a questo punto, esistono progetti analoghi a questo anche non open-source, PICO-8 ad esempio:

<https://www.lexaloffle.com/pico-8.php>.

Personalmente ho testato solo TIC-80, che ho compilato per il mio Linux Ubuntu, seguendo le istruzioni presenti sul repository github di TIC-80 <https://github.com/nesbox/TIC-80>.

Passiamo ora alla parte "ludica", lanciando l'eseguibile di TIC-80. Come fare per caricare una cartuccia? Da riga comandi, basta scrivere: **load nomecartuccia.tic** e quindi **run**. Vedere le cartucce disponibili? Basta digitare **dir** oppure **ls**. Come vedete, è fin troppo semplice ed intuitivo, sia per chi ha già un minimo di dimestichezza con una shell sia per chi è completamente nuovo a questo approccio.



Vi risulteranno sicuramente familiari. Il primo è un clone del famoso Tetris, mentre il secondo ricorda da vicino l'altrettanto famoso Wolfenstein 3D. Comunque, il catalogo dei giochi disponibili è abbastanza vasto ed in continua espansione. Potrete visionarlo a questo indirizzo: <https://tic.computer/play>

Per maggiori informazioni tecniche invece, il riferimento è: <https://tic.computer/learn>

Per scaricare "TIC-80" potete seguire invece i link presenti in questa pagina: <https://tic.computer/create>

Solo per dovere di cronaca, Vi informo che esiste una versione pro a fronte di un modico pagamento (5 Dollari) relativamente a "TIC-80", che fornisce ovviamente qualcosa in più rispetto alla versione base.

Tuttavia, leggiamo dalla pagina di github del progetto che "(...) per gli utenti che non vogliono spendere soldi, abbiamo reso semplice costruire la versione pro dal codice sorgente" quindi, pur non godendo ad esempio di un supporto utenti privilegiato, potreste comunque provare anche la versione "pro" di TIC-80.

Concludo questo articolo con una mia considerazione, che probabilmente apprezzeranno tanti amici retroappassionati ma non quelli più "integralisti": "TIC-80, perché no?" ☺

# RetroMath:

## Pianificare traiettorie ed evitare ostacoli

di Giuseppe Fedele

In questo articolo viene presentata una tecnica ereditata dalla robotica mobile per la pianificazione di traiettorie. Con il termine "robot mobile" si intende una struttura meccanica che sia capace di muoversi autonomamente in ambienti terrestri (indoor, outdoor) marini o aerei. Per muoversi il robot dovrà essere dotato di un sistema di locomozione, di un sistema sensoriale e di un'unità elaborativa. Pianificare una traiettoria significa determinare un percorso da un punto iniziale ad un punto finale (goal) che il robot mobile deve seguire. Questo percorso dovrà essere pianificato in modo tale da far fronte alle incertezze presenti nell'ambiente come ad esempio la presenza di ostacoli fissi o mobili rilevati durante il movimento.

### Bibliografia

[1] <http://www.robocup.com>

[2] M. Buckland, Programming Game AI by Example, Wordware Publishing, Inc.

[3] M.H. Overmars, Path Planning For Games, <http://www.cs.uu.nl/groups/AA/movies/publications/PDF/Oo5.pdf>

[4] O. Khatib, Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, Int. J. of Robotic Research, 5, 1, pag.90-98, 1986.

[5] G. Fedele. L. D'Alfonso, Obstacles Avoidance Based on Switching Potential Functions, J. of Intelligent & Robotic Systems, 90, 3-4, pag.387-405, 2018.

In calce all'articolo e' presente un listato in **Simon's Basic** per il **Commodore 64** che mostra una semplice applicazione dei concetti sviluppati nello stesso. Per semplicita' abbiamo creato anche un file D64 scaricabile dal nostro sito web all'indirizzo: <http://www.retromagazine.net/download/potrob.d64>

Per questo motivo si ha a che fare con due sotto-problemi principali che vanno sotto il nome di "path planning" e di "obstacle avoidance" affrontati seguendo tre fasi principali:

- **Mapping:** fase preliminare in cui il robot utilizza i sensori per esplorare l'ambiente in cui si trova al fine di ricavare tutte le informazioni necessarie per poter interagire con esso.
- **Planning:** è la fase in cui vengono calcolate le traiettorie che il robot deve seguire con tutte le operazioni necessarie per raggiungere il goal.
- **Driving:** consiste nel far eseguire il cammino individuato nella fase precedente imponendo una serie di vincoli sulla dinamica del robot.

Le fasi descritte non sono necessariamente sequenziali ma possono essere ripetute più volte (ad esempio quando si incontra un ostacolo potrebbe essere necessario rifare il mapping, pianificare una nuova traiettoria, ecc.).

Nei casi più complicati è possibile avere diversi robot mobili, ognuno dei quali deve raggiungere un goal evitando di collidere con gli altri robot (che quindi rappresentano degli ostacoli in movimento). Una tipica applicazione di queste problematiche è la RoboCup [1], una competizione calcistica tra squadre di robot (Figura 1). E' un'iniziativa ideata nel 1993 e avviata a partire dal 1997 con l'obiettivo di realizzare, entro il 2050, una squadra di robot umanoidi autonomi in grado di sfidare e, possibilmente, battere la squadra di calcio campione del mondo.



Figura 1. RoboCup.

Questi problemi sono presenti anche nella progettazione dei videogiochi [2]. In [3] sono riportati alcuni esempi di giochi di strategia in cui i veicoli non obbediscono ai vincoli strutturali reali e ruotano sul posto. In altri giochi i personaggi camminano attraverso gli ostacoli o rimangono bloccati. Spesso i personaggi seguono percorsi che li avvicinano ad unità nemiche. Nel caso di movimento in gruppo le cose peggiorano. I personaggi camminano l'uno attraverso l'altro, si comportano in modo innaturale e scelgono percorsi individuali sbagliati, non rispettando gli obiettivi di gruppo. In Figura 2 è riportata una scena dal gioco Command and Conquer: Generals della EA Games in cui è evidente uno dei problemi del path planning per unità multiple. Il gruppo che deve raggiungere l'obiettivo mostrato dalla freccia si divide e un'unità sceglie un percorso diverso da quello scelto dal gruppo.



Figura 2. Command and Conquer: Generals, EA Games.

Una trattazione di tutti gli algoritmi per il path planning e l'obstacle avoidance esula dallo scopo di questo articolo introduttivo. Qui si vuole presentare una delle tecniche base sulla generazione dei percorsi che va sotto il nome di **Tecnica dei Potenziali Artificiali** [4].

Una funzione potenziale non è altro che una funzione reale che ammette derivata in ogni punto dello spazio  $\mathbb{R}^m$  ( $m$  è la dimensione

dello spazio in cui il nostro robot mobile si muove: per un drone, ad esempio,  $m = 3$ ).

Il valore di una funzione potenziale può essere vista come una specie di energia. Questo significa che il robot dovrà muoversi cercando di raggiungere le zone in cui l'energia è minima. Ma procediamo con ordine!

Le più semplici funzioni potenziali sono quelle attrattive/repulsive. L'intuizione sotto questa teoria è abbastanza semplice: il goal attrae il robot mentre gli ostacoli lo respingono. La somma di questi due effetti attirerà il robot verso il goal facendolo deviare dagli ostacoli. Per semplificare la trattazione supponiamo che robot e ostacoli siano modellati come particelle puntiformi. Lo spazio in cui il robot si muove (nel nostro caso bidimensionale  $m = 2$ ) è sotto l'influenza di una funzione potenziale costruita come la somma di potenziali attrattivi e repulsivi:

$$U(q) = U_{att}(q) + U_{rep}(q)$$

dove  $q$  è la posizione corrente del robot

$$q = \begin{bmatrix} x \\ y \end{bmatrix}$$

**Potenziale attrattivo**

Ci sono diversi criteri che la funzione  $U_{att}$  deve soddisfare. Per prima cosa essa deve crescere monotonicamente all'aumentare della distanza dal goal.

La scelta più semplice è utilizzare una funzione del tipo

$$U_{att}(q) = \frac{1}{2} \mu_a [(x - x_G)^2 + (y - y_G)^2]$$

se il goal è in posizione  $q_G = \begin{bmatrix} x_G \\ y_G \end{bmatrix}$ , dove  $\mu_a$  è un parametro usato per scalare l'effetto del potenziale attrattivo. In questo caso,  $U_{att}(q)$  rappresenta una parabola centrata sul goal.

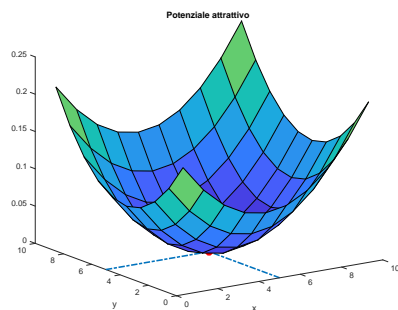


Figura 3. Potenziale attrattivo.

In Figura 3 è mostrato un potenziale attrattivo verso il goal (pallino rosso). L'idea è quella di considerare la funzione potenziale come un

particolare scivolo: se posizioniamo una biglia in un qualunque punto di questo scivolo, la biglia tenderà a raggiungere il punto di minimo della parabola che coincide appunto con il goal. Il gradiente (ovvero il vettore contenente le derivate della funzione rispetto alle due variabili  $x$  e  $y$ ) di  $U_{att}$  è:

$$\nabla U_{att}(q) = \mu_a \begin{bmatrix} x - x_G \\ y - y_G \end{bmatrix} = \mu_a (q - q_G)$$

che è un vettore dipendente dalla posizione corrente del robot che punta dal goal verso il robot ed ha modulo proporzionale alla distanza del robot dal goal. Più lontano  $q$  è da  $q_G$ , più grande è il modulo del vettore. In altre parole, quando il robot è lontano dall'obiettivo, si avvicina rapidamente; quando il robot è vicino all'obiettivo, si avvicina lentamente. La Figura 4 mostra il gradiente di  $U_{att}$  cambiato di segno (antigradiente). Le frecce puntano tutte verso il goal. Seguendo le frecce (ovvero la direzione dell'antigradiente di  $U_{att}$ ) il robot si avvicina al goal.

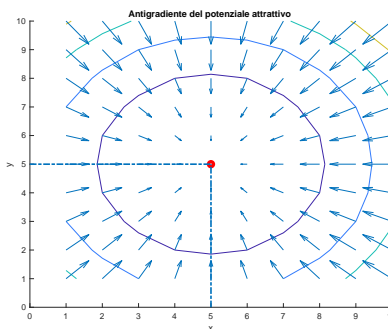


Figura 4. Antigradiente del potenziale attrattivo.

**Potenziale repulsivo**

Per ogni ostacolo è possibile definire un potenziale repulsivo che allontani il robot dall'ostacolo. Il potenziale repulsivo totale sarà dunque la somma dei vari potenziali repulsivi degli ostacoli.

L'entità della forza repulsiva dipende dalla vicinanza del robot all'ostacolo. Più vicino è il robot all'ostacolo, più forte dovrà essere la repulsione da esso. Considerando l'ostacolo in posizione

$$q_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

il potenziale repulsivo può essere definito in termini della distanza dall'ostacolo  $D(q) = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ , cioè:

$$U_{rep}(q) = \begin{cases} \frac{1}{2} \mu_r \left( \frac{1}{D(q)} - \frac{1}{Q^*} \right)^2, & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases}$$

dove  $Q^* \in \mathbb{R}$  è un fattore che permette al robot di ignorare ostacoli sufficientemente lontani e  $\mu_r$  può essere visto come un guadagno sul gradiente repulsivo.

Il gradiente del potenziale repulsivo è

$$\nabla U_{rep}(q) = \begin{cases} \mu_r \left( \frac{1}{Q^*} - \frac{1}{D(q)} \right) \frac{1}{D(q)^2} \nabla D(q), & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases}$$

dove

$$\nabla D(q) = \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} = q - q_0$$

Si noti che  $U_{rep}$  assume valore infinito in corrispondenza dell'ostacolo.

Numericamente conviene quindi inserire un limite sul valore massimo del fattore repulsivo. Le Figure 5 e 6 mostrano, analogamente al caso attrattivo, le caratteristiche del potenziale repulsivo.

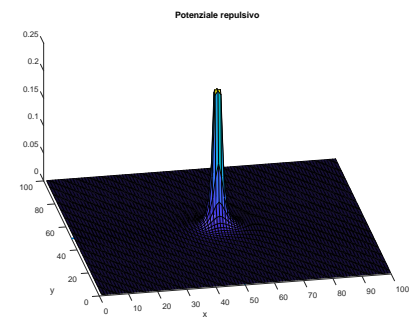


Figura 5. Potenziale repulsivo.

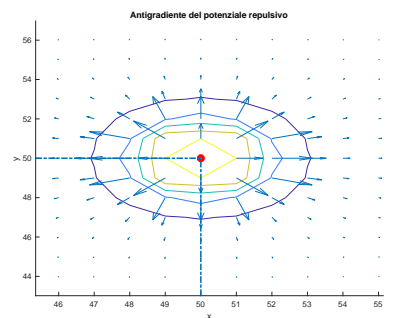


Figura 6. Antigradiente del potenziale repulsivo.

E adesso muoviamoci!!!

L'idea è semplice: partendo da una configurazione iniziale, facciamo un piccolo passo nella direzione opposta del gradiente. Questo fornisce una nuova posizione al robot e il processo si ripete fino a quando il gradiente è zero o quando il robot è vicino al goal.

Nello specifico l'algoritmo è il seguente:

1.  $q(0) = q_{start}$
2.  $i = 0$
3. while  $\nabla U(q(i)) \neq 0$  do
4.      $q(i+1) = q(i) + \alpha(i)\nabla U(q(i))$
5.      $i = i + 1$
6. end while

Il valore dello scalare  $\alpha(i)$  determina l'ampiezza del passo all'iterazione  $i$ -ma. E' importante che  $\alpha(i)$  sia scelto piccolo in modo che il robot non salti negli ostacoli. Infine è improbabile che la condizione  $\nabla U(q(i)) = 0$  sia soddisfatta esattamente. Per questo motivo è preferibile considerare una condizione del tipo

$$\|\nabla U(q(i))\| < \epsilon$$

ovvero il modulo del gradiente sia sufficientemente piccolo.

Ma attenzione!!!

L'approccio al path planning basato sulla definizione di un campo potenziale artificiale rappresenta dunque una tecnica semplice ed intuitiva che sfrutta un principio fisico di tipo energetico. Se ci si trova in condizioni di ambiente statico e completamente noto a priori, il potenziale può essere valutato offline producendo direttamente il profilo di velocità da applicare al robot. Il profilo è costruito per fare in modo che esso si muova lungo il campo energetico dal punto di partenza fino all'obiettivo. La tecnica può essere però facilmente estesa ed applicata in versione online, così da poter tenere conto, oltre che delle informazioni sull'ambiente note a priori, anche della presenza di eventuali ostacoli incontrati lungo il percorso e rilevati dai sensori a bordo del robot mobile.

Questa caratteristica particolarmente vantaggiosa consente di ottenere in sostanza quello che abbiamo chiamato obstacle avoidance, ovvero, è sufficiente determinare una volta per tutte il potenziale attrattivo dovuto all'obiettivo e quello repulsivo dovuto agli ostacoli noti, mentre si aggiorna ad ogni passo il potenziale repulsivo generato dagli ostacoli imprevisti e rilevati dai sensori di bordo durante il movimento. Anche se particolarmente semplici da implementare ed efficaci in buona parte delle situazioni di interesse pratico, i metodi basati sul potenziale artificiale, così come descritto nella sua forma generale, presentano diversi svantaggi che ne compromettono il corretto funzionamento.

Il limite di questa tecnica è rappresentato dalla eventuale presenza di minimi locali nella funzione potenziale (Figura 7) [5]. Questi punti critici non voluti costituiscono un

notevole problema qualora il robot vi si trovi intrappolato.

Un minimo locale si presenta, ad esempio, in corrispondenza di un perfetto bilanciamento tra potenziale attrattivo e repulsivo generando così un annullamento del gradiente che potrebbe indurre il robot a classificarlo come un falso goal. In particolari condizioni quindi la presenza di minimi locali diversi dall'obiettivo, ma comunque caratterizzati da un gradiente nullo, possono far sì che il robot resti intrappolato senza poter completare il compito che gli è stato assegnato.

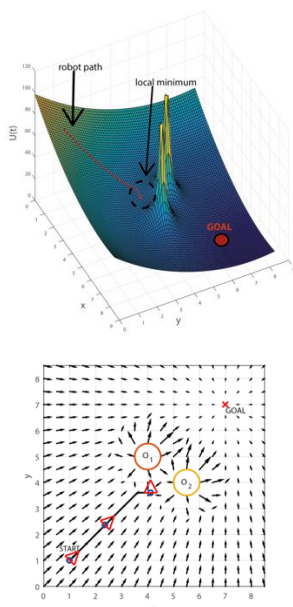


Figura 7. Esempio di minimo locale.

#### Un esempio su C64+Simons' Basic

Il codice riportato di seguito mostra una semplice implementazione delle cose viste prima. Il robot parte dalla posizione indicata con il cerchio in alto a sinistra e ha come obiettivo il goal indicato con il doppio cerchio in basso a destra. Sul percorso si trovano i due ostacoli mostrati al centro della Figura 8.

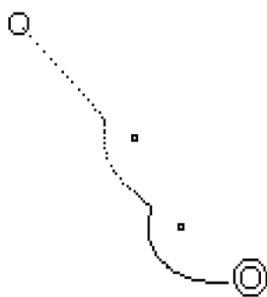


Figura 8. Esempio su C64

```

10 REM POS INIZIALE
20 DIM S(2) : S(1)=50 : S(2)=20
30 REM GOAL
40 DIM G(2) : G(1)=150 :
G(2)=120
50 REM OSTACOLI
60 NO=2 : DIM O(2,NO)
70 O(1,1)=100 : O(2,1)=65
80 O(1,2)=120 : O(2,2)=100
90 :
100 REM INIT HIRES
110 GOSUB 5000
1000 DIM P(2) : P(1)=S(1) :
P(2)=S(2)
1010 REM POTENZIALE ATTRATTIVO
1020 DIM GA(2)
1025 MA=0.1
1030 REM POTENZIALE REPULSIVO
1040 DIM GR(2,NO)
1045 MR=10000 : QS=100
1050 A=0.3
2000 GA(1)=MA*(P(1)-G(1))
2010 GA(2)=MA*(P(2)-G(2))
2015 P(1)=P(1)-A*GA(1)
2016 P(2)=P(2)-A*GA(2)
2020 FOR I=1 TO NO
2030 : D=SQR((P(1)-
O(1,I))^2+(P(2)-O(2,I))^2)
2040 : GR(1,I)=MR*(1/QS-
1/D)*(P(1)-O(1,I))/(D^3)
2050 : GR(2,I)=MR*(1/QS-
1/D)*(P(2)-O(2,I))/(D^3)
2060 : IF GR(1,I)>20 THEN
GR(1,I)=20
2070 : IF GR(2,I)>20 THEN
GR(2,I)=20
2080 : IF D<=QS THEN P(1)=P(1)-
A*GR(1,I) : P(2)=P(2)-A*GR(2,I)
2090 NEXT I
2100 PLOT P(1),P(2),1
2110 DG=SQR((P(1)-G(1))^2+(P(2)-
G(2))^2)
2120 IF DG>10 THEN GOTO 2000
4000 A$=""
4010 GET A$ : IF A$="" THEN GOTO
4010
4020 NRM
4030 END
5000 HIRES 0,1
5010 CIRCLE S(1),S(2),5,5,1
5020 CIRCLE G(1),G(2),5,5,1
5030 CIRCLE G(1),G(2),8,8,1
5040 FOR I=1 TO NO
5050 : CIRCLE
O(1,I),O(2,I),2,2,1
5060 NEXT I
5070 RETURN

```

# Il mistero del primo compilatore nella storia dell'informatica

di Alberto Apostolo

Dopo la pubblicazione dell'articolo "**La trinità femminile dell'informatica**" (*apparso sul numero 8 di RM, pag.2*), Matt Davies (un lettore britannico di RM) ci segnala che il primo compilatore non fu l'A0 progettato da Grace Murray Hopper, ma Autocode di Alick E. Glennie presso l'Università di Manchester (fonte: Wikipedia).

Confesso pubblicamente la mia ignoranza riguardo ad Alick Edwards Glennie. Ho cercato di documentarmi perchè sembrava non esserci unanimità sulla questione, in quanto la stessa Wikipedia su Autocode afferma: "...is considered by some to be the first compiled programming language".

Alla fine degli anni '40 del XX secolo, stava nascendo l'esigenza di rendere più maneggevole l'uso del computer da parte degli esseri umani. Per esempio, nel 1949, sul computer EDSAC esisteva un programma (assemblatore) per tradurre in linguaggio macchina una serie di simboli e per riutilizzare certe sequenze di istruzioni senza doverle riscrivere ogni volta che servivano (macro-istruzioni).

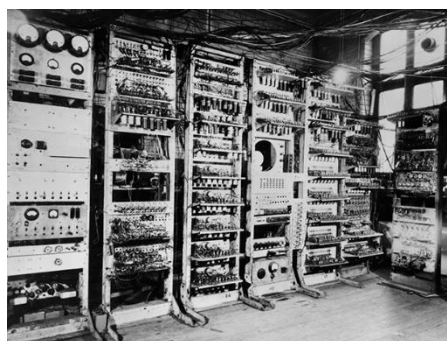
Ovviamente ciò non era sufficiente e gli sforzi degli scienziati dell'informazione erano volti a creare qualcosa di livello più alto che potesse tradurre formule matematiche in linguaggio macchina.

I primi lavori teorici sui compilatori sono attribuiti a K. Zuse e H. Rutishauser ("Automatische Rechenplanfertigung") con una presentazione ufficiale di Rutishauser al meeting G.A.M.M. del 1951 [Nih14], tenuto a Freiburg im Breisgau ([www.gamm-ev.de](http://www.gamm-ev.de)). Altri lavori teorici furono compiuti in quello stesso periodo anche dall'italiano Corrado Böhm [Met14].

Alick Edwards Glennie (1925-2003) fu uno scienziato britannico, laureato presso l'Università di Edimburgo, che ebbe l'occasione di lavorare con Alan Turing su algoritmi inerenti al gioco degli scacchi [Atk98].

In [Reio3] si riporta che nel 1952 Edwards Glennie progettò, per il computer

Ferranti/Manchester Mark I, un linguaggio ad alto livello chiamato Autocode con il suo compilatore chiamato Autocoder e che nel 1954 Tony Brooker implementò il "vero" Autocode.



*Il Manchester Mark I nel Manchester University's Computer Machine Laboratory.  
Fonte: SSPL/Getty Images*

Secondo la testimonianza dello scienziato britannico Christopher Strachey (1916-1975), il software di Glennie fu operativo a partire dal Settembre 1952 [Met14].

In seguito Glennie presentò una lezione su Autocode nel Febbraio 1953 alla Cambridge University basandosi su appunti da egli stesso scritti nel Dicembre 1952 [Das14].

Contemporaneamente negli Stati Uniti era stato portato avanti il concetto di "interpretare" una sorta di pseudo-codice ma, alla fine del 1951, Grace Hopper sviluppò l'idea che ciò non era necessario e che si poteva tradurre una volta per tutte il codice sorgente di un programma in linguaggio macchina.

Grace Hopper e i suoi collaboratori presso UNIVAC misero a punto un programma sperimentale che avrebbe dovuto fare quella traduzione e lo chiamarono "compiling routine".

Il loro primo compilatore (in senso propriamente detto) fu l'A0, reso operativo nella primavera del 1952, quando Grace Hopper tenne un discorso sull'argomento alla prima conferenza della Association for Computing Machinery [Met14]. Poco tempo dopo apparve la versione A1 (Gennaio 1953) seguita dalla A2 (Agosto 1953).

Fu così che, dal biennio 1953-1954, la parola "compilatore" cominciò a divenire di dominio pubblico per definire un tipo di processo richiesto per tradurre in linguaggio macchina programmi scritti in forma algebrica [Das14].

Successivamente, nel 1954, John Backus e il suo team iniziarono a sviluppare (presso IBM) quello che poi divenne il linguaggio FORTRAN (FORmula TRANslator).

Pertanto in base alle fonti in Bibliografia, compreso il testo di N. Metropolis (con la soluzione del "mistero"), si deduce che il primo compilatore operativo fu quello di Grace Hopper (per pochi mesi di differenza).

Spero di essere stato chiaro e ringrazio comunque Matt Davies per la segnalazione che mi ha permesso di imparare qualcosa di nuovo. E spero anche che Matt continui a leggere con interesse RetroMagazine.

## Bibliografia

[Atk98] G.W. Atkinson, "**Chess and Machine Intuition**", Intellect Books, 1998.

[Das14] S. Dasgupta, "**It Began with Babbage: The Genesis of Computer Science**", Oxford University Press, 2014.

[Met14] N. Metropolis, "**History of Computing in the Twentieth Century**", Elsevier, 2014.

[Nij14] A. Nijholt, "**Computers and Languages: Theory and Practice**", Elsevier, 2014.

[Reio3] E.D. Reilly, "**Milestones in Computer Science and Information Technology**", Greenwood Publishing Group, 2003.



# Esplorando l'Amiga - parte 2

di Leonardo Giordani

## La jump table delle librerie

Come già visto precedentemente, quando una libreria viene caricata in memoria il sistema crea una jump table subito prima dell'indirizzo base della libreria. Questa tabella contiene gli indirizzi delle funzioni pubbliche della libreria, e anche Exec stessa ne ha una.

L'ordine delle funzioni nella jump table della libreria Exec è specificato in uno dei file include forniti dall'NDK:

```
include_i/exec/exec_lib.i.
```

```
FUNCDEF Supervisor
FUNCDEF execPrivate1
FUNCDEF execPrivate2
FUNCDEF execPrivate3
...
FUNCDEF OpenLibrary
...
```

Come si può vedere il file fa uso della macro `FUNCDEF`, che non è fornita e deve essere implementata dal programmatore. L'idea alla base della macro è molto semplice: siccome l'ordine della jump table non cambia possiamo sostituire il primo `FUNCDEF` con l'offset della prima funzione della libreria, per poi incrementarlo con la dimensione di default di un indirizzo. L'output atteso della macro è il seguente:

```
_LVOSupervisor      EQU    -30
_LVOexecPrivate1    EQU    -36
_LVOexecPrivate2    EQU    -42
_LVOexecPrivate3    EQU    -48
...
_LVOOpenLibrary     EQU    -552
...
```

Si faccia attenzione al fatto che il nome delle funzioni è stato sostituito aggiungendo il prefisso `_LVO` per evitare di utilizzare simboli già definiti (`LVO` sta per `Library Vector Offset`).

I numeri utilizzati nel codice precedente vengono dalle costanti fornite in `include_i/exec/libraries.i`

```
*----- Special Constants -----
LIB_VECTSIZE      EQU 6      ;Each library entry takes 6 bytes
LIB_RESERVED      EQU 4      ;Exec reserves the first 4 vectors
LIB_BASE          EQU -LIB_VECTSIZE
LIB_USERDEF       EQU LIB_BASE-(LIB_RESERVED*LIB_VECTSIZE) ;First user func
LIB_NONSTD        EQU LIB_USERDEF
```

Come si vede dai commenti, Exec considera riservati i primi 4 vettori, pertanto l'indirizzo della prima funzione sarà `LIB_USERDEF`. Per comprendere il motivo per cui gli indirizzi sono negativi e come l'offset è calcolato consideriamo uno schema della memoria una volta che la libreria è stata caricata:

Potete trovare una versione ufficiale di questo schema nella documentazione [http://amigadev.elowar.com/read/ADCD\\_2.1/AmigaMail\\_Vol2\\_guide/node018g.html](http://amigadev.elowar.com/read/ADCD_2.1/AmigaMail_Vol2_guide/node018g.html).

Attenzione al fatto che l'immagine nella documentazione rappresenta la memoria all'inverso, con gli indirizzi che crescono verso il basso.

```
INDIRIZZI ALTI IN MEMORIA
+-----+
Ultimo byte della | Fine della libreria |
libreria caricato ----->+-----+
in memoria       | [...] |
+-----+
| Contenuto della libreria|
+-----+
| Struttura della libreria|
Indirizzo base ----->+-----+
| 1o vettore riservato |
+-----+<--- LIB_BASE
| 2o vettore riservato |
+-----+<---+
| 3o vettore riservato | | LIB_VECTSIZE
+-----+<---+
| 4o vettore riservato |
+-----+
| 1a funzione definita |
+-----+<--- LIB_USERDEF
| 2a funzione definita |
+-----+
| [...] |
+-----+
Primo byte della | Fine della jump table |
libreria caricato ----->+-----+
in memoria       | INDIRIZZI BASSI IN MEMORIA
```

Come si può notare la libreria viene caricata come ci si aspetta dall'indirizzo base verso gli indirizzi più alti, ma allo stesso tempo la jump table viene aggiunta *in ordine inverso*. Questo permette di trovare l'indirizzo di una funzione con un indice semplice invece di dover utilizzare un algoritmo più complesso. La funzione numero 1 è all'indirizzo `-1 * address_size`, la funzione numero 2 all'indirizzo `-2 * address_size`, eccetera.

Questo è il motivo per cui si utilizzano offset negativi per chiamare le funzioni di libreria, ma offset positivi per accedere a dati e strutture interne a quest'ultima.

Dalla figura si vede anche chiaramente dove le costanti `LIB_BASE` e `LIB_USERDEF` sono utilizzate.

I valori effettivi delle costanti sono:

```
LIB_BASE EQU -6
LIB_USERDEF EQU -30
```

Una buona definizione della macro `FUNCDEF`, pertanto, è la seguente:

```
INCLUDE "exec/libraries.i"
MACRO FUNCDEF
_LVO\1 EQU FUNC_CNT
FUNC_CNT SET FUNC_CNT-
LIB_VECTSIZE
ENDM
FUNC_CNT SET LIB_USERDEF
```

L'ultima riga crea il simbolo `FUNC_CNT` con il valore `LIB_USERDEF`. Ogni chiamata successiva della macro `FUNCDEF <arg>` fa le seguenti cose:

1. Crea il simbolo `_LVO<arg>` con il valore `FUNC_CNT` (ad esempio `_LVOSupervisor EQU -30`)
2. Decrementa il simbolo `FUNC_CNT` del valore `LIB_VECTSIZE`

Si faccia attenzione al fatto che la funzione `FUNCDEF` di esempio che si trova (commentata) in `libraries.i` non funziona senza modifiche in quanto `FUNC_CNT` è definita all'interno della macro stessa, mentre dovrebbe essere definita prima del primo utilizzo della macro.

Vedi riquadro:

```
*----- FUNCDEF is used to parse library offset tables. Many applications
*----- need a special version of FUNCDEF - you provide your own macro
*----- to match your needs. Here is an example:
*
* FUNCDEF MACRO
* _LVO\1 EQU FUNC_CNT
* FUNC_CNT SET FUNC_CNT-6 * Standard offset-6 bytes each
* FUNC_CNT EQU LIB_USERDEF * Skip 4 standard vectors
* ENDM
```

Potete mettere il codice della macro `FUNCDEF` in un include fail locale come `funcdef.i`. Includere questo file nel proprio codice permette di utilizzare i prefissi che iniziano con `_LVO` per chiamare la funzioni che vogliamo.

```
INCLUDE "funcdef.i"
INCLUDE "exec/exec_lib.i"
move.l 4.w,a6
clr.l d0
move.l #libname,a1
jsr _LVOOpenLibrary(a6)
libname:
dc.b "somename.library",0
```

Infine, se si vuole essere ancora più espliciti, si può utilizzare la macro `CALLLIB` definita in `libraries.i` e scrivere qualcosa come

```
INCLUDE "funcdef.i"
INCLUDE "exec/exec_lib.i"
INCLUDE "exec/libraries.i"
move.l 4.w,a6
clr.l d0
move.l #libname,a1
CALLLIB _LVOOpenLibrary
libname:
dc.b "somename.library",0
```

### I quattro vettori riservati

Come si è visto l'Amiga riserva 4 vettori all'inizio della jump table di una libreria. Questi 4 spazi ospitano 3 funzioni standard che devono essere fornite da ogni libreria: `Open()`, `Close()`, ed `Expunge()`. Il quarto slot è riservato per possibili future espansioni e deve contenere una funzione che restituisce 0.

Gli offset di queste funzioni sono contenuti nella libreria:

```
include_i/exec/libraries.i
*-----
*
* Standard Library Functions
*
*-----
LIBINIT LIB_BASE
LIBDEF LIB_OPEN
LIBDEF LIB_CLOSE
LIBDEF LIB_EXPUNGE ; must exist
in all libraries
LIBDEF LIB_EXTFUNC ; for future
expansion - must return zero.
```

e il prodotto finale di queste macro, assieme alle costanti viste prima, è

```
LIB_OPEN EQU -6
LIB_CLOSE EQU -12
LIB_EXPUNGE EQU -18
LIB_EXTFUNC EQU -24
```

Potete provare a seguire la definizione delle macro `LIBINIT` e `LIBDEF`, confrontando il vostro risultato con quello fornito.

### Tipi e strutture

Vediamo come la libreria `Exec` definisce i suoi tipi, che sono i componenti base del sistema operativo dell'Amiga. Il principale punto di partenza per questa ricerca è `include_i/exec/types.i`.

Quando si definiscono delle strutture in Assembly tutto viene espresso in termini di offset. L'idea principale alla base di una struttura è di creare qualcosa come:

```
STRUCT1 EQU 0
OFFSET SET 0
FIELD1 EQU OFFSET
```

```

OFFSET      EQU
OFFSET+SIZE_OF_FIELD1

FIELD2      EQU      OFFSET

OFFSET      EQU
OFFSET+SIZE_OF_FIELD2

[...]

STRUCT1_SIZE EQU      OFFSET

```

che, una volta eseguite le macro, si trasforma nel seguente codice:

```

STRUCT1      EQU      0
FIELD1      EQU      0

FIELD2      EQU
SIZE_OF_FIELD1

FIELD3      EQU
SIZE_OF_FIELD1+SIZE_OF_FIELD2

[...]

STRUCT1_SIZE EQU
SIZE_OF_FIELD1+...+SIZE_OF_FIELDn

```

Le macro che definiscono i tipi sono quindi composte da codice come questo

```

TYPENAME    MACRO
\1          EQU      SOFFSET

SOFFSET     SET
SOFFSET+SIZE_OF_TYPE

ENDM

```

Per esempio la macro BYTE è

```

BYTE        MACRO      ; byte (8
bits)
\1          EQU      SOFFSET

SOFFSET     SET      SOFFSET+1

ENDM

```

Si noti che il campo della struttura è definito con EQU per evitare sovrascritture accidentali, mentre SOFFSET usa SET che permette la ridefinizione dei simboli.

Vediamo ora una struttura reale. Un buon esempio è fornito da LN, definito in include\_i/exec/nodes.i, che rappresenta un nodo di una linked list.

```

STRUCTURE   LN,0      ; List
Node

  APTR      LN_SUCC ; Pointer to
next (successor)

  APTR      LN_PRED ; Pointer to
previous (predecessor)

  UBYTE     LN_TYPE

  BYTE      LN_PRI  ; Priority,
for sorting

  APTR      LN_NAME ; ID string,
null terminated

  LABEL     LN_SIZE ; Note: word
aligned

```



La macro STRUCTURE è definita in types.i come

```

STRUCTURE   MACRO      ;
structure name, initial offset

\1          EQU      0

SOFFSET     SET      \2

ENDM

```

E il codice risultante, una volta che le macro sono state eseguite, è il seguente

```

LN          EQU      0
LN_SUCC     EQU      0
LN_PRED     EQU      4
LN_TYPE     EQU      8
LN_PRI      EQU      9
LN_NAME     EQU      10
LN_SIZE     EQU      14

```

Come si può vedere i nomi dei campi non sono altro che offset all'interno della struttura e non c'è una spaziatura specifica (padding) alla fine della struttura. In questo caso non ce n'è bisogno, in quanto la dimensione della struttura è già un multiplo di una word (14 byte).

### Come allineare le strutture

Se necessitassimo di allineare i byte della struttura, ad ogni modo, possiamo utilizzare un piccolo trucco dell'aritmetica binaria. Se si ignora il bit meno significativo di un numero binario lo si converte nel numero pari più vicino (verso il basso). Un esempio in Python è il seguente:

```

>>> bin(13)
>>> '0b1101'
>>> bin(12)
>>> '0b1100'

```

Siccome i numeri pari sono separati esattamente da due posizioni possiamo, dato un qualsiasi offset, incrementarlo di uno e poi arrotondarlo verso il basso al numero pari più

vicino. In questo modo abbiamo ottenuto un numero pari maggiore o uguale a quello di partenza. Se il numero di partenza è la dimensione di una struttura in byte abbiamo allineato la struttura a multipli di 2 byte, cioè una word. La macro ALIGNWORD definita in include\_i/exec/types.i implementa esattamente questo algoritmo:

```

ALIGNWORD   MACRO      ; Align
structure offset to nearest word

SOFFSET     SET
(SOFFSET+1) &$fffffffe

ENDM

```

Questa macro può essere vista in azione nella struttura CardHandle definita in include\_i/resources/card.i.

Per questo mese è tutto, appuntamento al prossimo numero.

### Prossimi articoli...

Nel prossimo articolo vedremo come viene definita la jump table della libreria Exec e inizieremo a lavorare sul codice di Kickstart 1.3 per vedere come questi concetti sono stati applicati in un caso reale.

### Risorse

Amiga System Programmers Guide, Abacus: [https://archive.org/details/Amiga\\_System\\_Programmers\\_Guide\\_1988\\_Abacus](https://archive.org/details/Amiga_System_Programmers_Guide_1988_Abacus)

AmigaOS Developer Docs: <http://amigadev.elowar.com>

### \*\*\* Note for the English readers \*\*\*

If you are interested in the English version of this article, it can be found on Leonardo's blog at the url:

<http://blog.thedigitalcatonline.com/blog/2018/05/28/exploring-the-amiga-2/>

# Le stravaganti calcolatrici di Clive Sinclair

di Alberto Apostolo

Chi ha posseduto un computer Sinclair, o comunque ha vissuto i mitici anni Ottanta del XX secolo e l'epopea degli home-computer a 8 bit, conosce Clive Sinclair come geniale e visionario costruttore di computer (basti pensare alla fortunata serie ZX e al meno fortunato QL). Inoltre si ricorda che cerco' di immettere sul mercato il Sinclair C5, un lento e basso triciclo elettrico che non ebbe successo a causa di svariati problemi tecnici.



Figura 1: Il veicolo Sinclair C5.

Ma forse non tutti sanno che Clive Sinclair (nato a Richmond nel Surrey il 30 luglio 1940) comincio' giovanissimo fondando nel 1961 la Sinclair Radionics Ltd che produceva radioline, TV, impianti HI-FI miniaturizzati come anche un orripilante orologio a LED (il Black Watch del 1975, la cui scarsa qualita' era colpa di un subcontraente).

Nel corso degli anni '70 del XX secolo inizio' la produzione e la vendita (anche in kit di montaggio!) di calcolatrici a basso costo con display a LED, che tentavano di imitare quelle prodotte dalla Texas Instruments e dalla Hewlett-Packard nello stesso periodo.

Tra i modelli piu' curiosi vi era senza dubbio la calcolatrice da polso (Figura 4), il cui insuccesso commerciale (unito all'avvento delle calcolatrici a cristalli liquidi) mise in crisi la Sinclair Radionics e costrinse Clive Sinclair ad avvalersi di aiuti statali.

In questa sede non si descriveranno tutti i modelli di calcolatrice prodotti da Sinclair (a tale scopo si rimanda a [VC18] con il link in Bibliografia) ma solo alcuni modelli per evidenziarne le originali idee costruttive. Infine (anche se non e' una calcolatrice) si dara' un brevissimo accenno al poco noto multimetro digitale realizzato dalla Sinclair.

In Appendice sono indicati alcuni link a siti interessanti per altri approfondimenti.



Figura 2: Clive Sinclair da giovane.



Figura 3: Clive Sinclair negli anni '80.

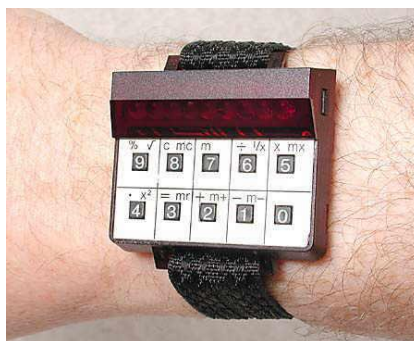


Figura 4: Sinclair Wrist Calculator del 1977.

## La prima calcolatrice Sinclair

La Sinclair Executive (Figura 5) fu una delle prime calcolatrici al mondo veramente tascabili, prodotta e messa in commercio a partire dall'Agosto 1972 al prezzo di 79 Sterline piu' VAT (IVA britannica, n.d.A.). Poteva eseguire le quattro operazioni e poco piu' (la radice quadrata, il calcolo del reciproco e la scelta tra calcolare in virgola fissa o in virgola mobile).

L'hardware era caratterizzato da 22 transistor, 50 resistenze, 17 condensatori e il chip GLS 1802 prodotto dalla Texas Instruments (allora considerato una meraviglia della elettronica).

Per limitare il consumo delle batterie a causa del chip, era stato adottato un trucco scoperto da Jim Westwood (il capo progettista della Sinclair Radionics): se si interrompeva l'alimentazione per un brevissimo tempo, il display e il contenuto della memoria non si cancellavano. Con i transistor saldati nel circuito che potevano immagazzinare fino a 5 secondi di carica, era piuttosto semplice accendere e spegnere l'alimentazione con un impulso temporizzato. In tal modo la potenza assorbita scendeva da 350 a 30 mW.

Un trafiletto riportato sul numero 805 della rivista americana New Scientist [NS72] descriveva il processo: "...La Sinclair Executive dispone di un singolo chip MOS contenente 7000 transistor. Questo tipo di circuito integrato normalmente consuma 350 mW di potenza, fornita da una batteria ricaricabile piuttosto voluminosa. Il trucco della miniaturizzazione di Sinclair dipende da uno speciale circuito che riduce l'assorbimento a circa 20mW. Cio' e' realizzato con l'eliminazione di tutti i resistori dalla circuiteria e con un clock che interrompe l'alimentazione per la maggior parte del tempo. Tre piccole batterie al mercurio forniscono 20 ore di energia (circa quattro mesi di normale uso).

L'alimentazione e' in funzione solo per impulsi di 1.7 microsecondi, una quantita' determinata dal tempo di immagazzinamento di un transistor di controllo. L'unita' ha un temporizzatore che opera a 200 kHz durante i calcoli e poi scende a 15 kHz tra una operazione e l'altra. Cosi' il tempo di spegnimento dura da 3.3

microsecondi durante i calcoli a piu' di 65 nella fase intermedia. Il dispositivo si basa sulla capacita' dei chip di memorizzare informazioni durante i momenti di spegnimento, e 1.7 microsecondi e' abbastanza affinche' il chip effettui un singolo cambio di stato dell'elettronica. Ogni calcolo puo' essere fatto in 1000 tali cambiamenti.

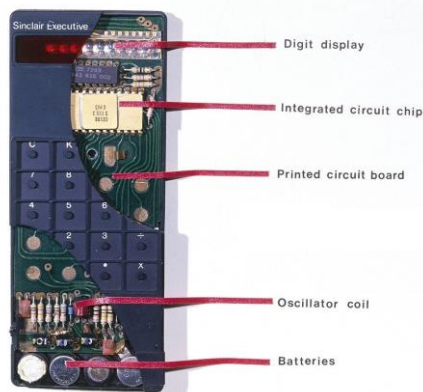


Figura 5: Calcolatrice Sinclair Executive.

Le critiche non mancarono. Pochi mesi piu' tardi, nell'autunno del 1972, Kenneth Corfield (per conto di New Scientist) chiese di effettuare dei test di qualita' sulla Sinclair Executive ottenendo risultati sconcertanti (non tanto sull'elettronica ma sulla parte meccanica del progetto [Cor85]).

Inoltre se lasciata troppo tempo accesa, le sue batterie si surriscaldavano fino ad esplodere.

### Il modello Sinclair Scientific

Nel 1974 Clive Sinclair intendeva mettere in vendita a 100\$ una versione economica della calcolatrice Hewlett-Packard HP35 che ne costava 395 (e la cui progettazione aveva richiesto lo sforzo di un gruppo di 20 ingegneri e l'investimento di un milione di dollari dell'epoca).

Dopo un infruttuoso accordo commerciale con il costruttore Bowmar, Sinclair guardo' alla Texas Instruments che gli propose il chip TMS0805 con tecnologia PMOS. Era un chip a buon mercato che poteva effettuare solo le quattro operazioni, aveva 3 registri di memoria senza la gestione delle subroutine e incorporava una R.O.M. con una capacita' massima di 320 istruzioni.

In pochi giorni, Nigel Searle (genio della programmazione con un PhD in matematica) riusci' a sbalordire gli esperti della Texas Instruments scrivendo il codice per la prima calcolatrice scientifica del mondo a chip singolo (con seno, coseno, tangente, arco-

seno, arco-coseno, arco-tangente, logaritmo e antilogaritmo in base 10).

Il modello Sinclair Scientific (Figura 6) faceva uso della Notazione Polacca Inversa (Reverse Polish Notation o R.P.N.) e rappresentava i numeri sul display solo in notazione scientifica (mantissa ed esponente).

Data la logica circuitale in base 10, le quattro operazioni erano eseguite con gli stessi metodi che si imparano a scuola.

Per calcolare le funzioni trigonometriche seno, coseno e tangente di un angolo  $\theta$  si ricorreva a un ingegnoso algoritmo consistente nell'effettuare ripetute rotazioni di 0.001 radianti del vettore di coordinate (1; 0.0005) fino a raggiungere l'angolo  $\theta$  (usare 0.0005, invece di 0, forniva l'arrotondamento).

Essendo 0.001 radianti un angolo molto piccolo, le formule per effettuare una rotazione si potevano approssimare come segue:

$$\begin{cases} X_{succ} = X_{prec} \cos 0.001 - Y_{prec} \sin 0.001 \approx X_{prec} - Y_{prec}/1000 \\ Y_{succ} = X_{prec} \sin 0.001 + Y_{prec} \cos 0.001 \approx X_{prec}/1000 + Y_{prec} \end{cases}$$

La divisione per 1000 di un registro si eseguiva ripetendo tre volte uno shift a destra del registro stesso. Al raggiungimento dell'angolo  $\theta$  le coordinate ( $X_{succ}$ ,  $Y_{succ}$ ) avevano valore ( $\cos \theta$ ,  $\sin \theta$ ). Infine la tangente si otteneva dividendo seno per coseno.

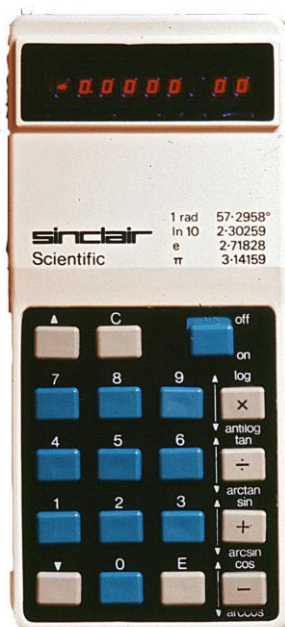


Figura 6: Calcolatrice Sinclair Scientific.

Per le funzioni trigonometriche inverse del seno e del coseno, si calcolava la rotazione del vettore (1, 0.0005) finche' seno o coseno non raggiungevano il valore di input. Poi si considerava il numero di rotazioni che erano state necessarie e si divideva per 1000 (il risultato di tale divisione dava il valore desiderato dell'arco-seno o dell'arco-coseno).

Invece il procedimento per calcolare arctg z era leggermente diverso: la rotazione partiva dalle coordinate (z, 1) fino a raggiungere (0, 1).

I metodi per il logaritmo e l'antilogaritmo erano molto complicati, in quanto effettuati in base 0.99. Il cuore dei procedimenti era moltiplicare ripetutamente il valore di input x per 0.99, ossia calcolare  $x - x/100$  (sottrarre a x lo stesso valore shiftato due volte a destra in un altro registro).

Lo sforzo per rendere scientifica una calcolatrice con un hardware non predisposto in tal senso, fu pagato in termini di precisione e di velocita' di esecuzione nei calcoli (a causa del ridotto numero di registri, sul display

apparivano i risultati intermedi durante un calcolo).

A volte poteva succedere che il codice di Nigel Searle non fosse in grado di gestire un valore pertinente in input a una certa funzione. In quei casi, la calcolatrice snocciolava numeri come fosse "impazzita" bloccandosi dopo diversi secondi con un risultato completamente sbagliato sul display (sic!).

### Il modello Sinclair Cambridge Programmable

La Sinclair Cambridge Programmable (Figura 7) fu messa in commercio agli inizi del 1977 ed era basata sui circuiti integrati National Semiconductor MM5799EHY/N e DS7784N. Aveva un display a 8 cifre a LED rossi (sempre fabbricato dalla National Semiconductor).

L'impiego di una batteria da 9V in formato PP3 conferiva all'involucro un caratteristico rigonfiamento nella parte sottostante (le sue dimensioni erano 50 mm x 111 mm x 28 mm).

Come altre calcolatrici Sinclair, era proposta come alternativa dai costi ragionevoli rispetto ai modelli Hewlett-Packard e Texas Instruments.

Assieme alla calcolatrice si fornivano 4 manuali di programmazione sui seguenti argomenti:

- Uso generale
- Finanza e statistica
- Matematica, fisica e ingegneria
- Elettronica

I manuali (scritti molto bene) permettevano di sfruttare al massimo le funzionalita' della macchina.



Figura 7: Sinclair Cambridge Programmable.

Per la programmazione erano riservati 36 passi (numerati da 00 a 35 in una memoria volatile) ed erano stati approntati degli accorgimenti per risparmiare spazio. Dal momento che la pressione di un tasto costava un passo, le funzioni erano accessibili in modalita' programma tramite il tasto shift in alto a sinistra. Per inserire un numero bisognava precedere l'input con il carattere # ottenuto tramite il tasto [3], il quale costava un passo. Era previsto un solo tipo di salto condizionale, **goto if neg**, dal quale ogni altro test poteva derivare stando attenti alla stesura del programma. Infine dopo il passo 35 era previsto il ritorno al passo 00, che consentiva spesso di risparmiare un salto incondizionato **goto**.

*Domanda semi-seria riservata agli utenti Sinclair ZX: vi ricorda qualcosa la modalita' di selezione e la disposizione delle funzioni sui tasti della calcolatrice?*

A forza di usarla, con il passare del tempo si scoprivano altri trucchi come ad esempio il

seguito ([Voi18]): premendo il tasto corrispondente alla divisione, si interrompeva l'esecuzione del programma e il display mostrava il valore corrente. Premendo il tasto **RUN** l'esecuzione continuava (un aspetto non ben documentato sui manuali). Questo permetteva di scrivere qualche piccolo "gioco" (lancio dei dadi, numeri casuali, un cronometro, ecc.).

Nonostante l'economia di realizzazione, i tasti non soffrivano il problema del "rimbalzo" che affliggeva le Texas Instruments (semmai si fosse verificato sulla Sinclair, spesso era risolvibile premendo con forza sul tasto desiderato).

### Il multimetro digitale

Anche se non e' una calcolatrice, merita una menzione il multimetro digitale Sinclair PDM35 Multimeter (risalente al Dicembre 1977) con involucro preso da quello delle calcolatrici Sinclair Oxford [VC18] (Figura 8).

Il display LED consentiva la visualizzazione di tre cifre e mezza (ossia di valori tra -1999 e 1999) con l'inclusione del punto decimale (Figura 9). Il multimetro era alimentato da una batteria PP3 da 9V e le sue dimensioni erano 73 mm x 155 mm x 33 mm.



Figura 8: Sinclair PDM35 Multimeter.

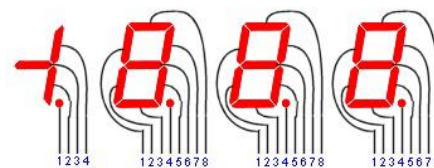


Figura 9: Esempio di display a 3 cifre e mezza.

### Appendice

Per scaricare i manuali delle calcolatrici Sinclair (scannerizzati e salvati in formato PDF): <http://www.wass.net/manuals/>

Il Museo Web delle calcolatrici vintage (non solo Sinclair): <http://www.vintagecalculators.com/index.html>

Il blog di Ken Shirri nel quale si possono trovare notizie interessanti ed emulatori online delle calcolatrici vintage: <http://www.righto.com/>

### Bibliografia

[Boz96] M. Bozzo, La grande storia del computer: dall'abaco all'intelligenza artificiale, Edizioni Dedalo, 1996.

[Cor85] K. Corfield, A triumph of persistence, New Scientist, n.1486, 12 dicembre 1985.

[Gru12] A. Grussu, 1982-2012: 30 anni di Spectrum, [http://www.gamesark.it/mostra\\_speciale.asp?c=154201220002396278&a=0&f=1](http://www.gamesark.it/mostra_speciale.asp?c=154201220002396278&a=0&f=1)

[LE18] sito aziendale della London Electronics Limited, [https://www.london-electronics.com/LEL\\_New/panel-meter-information.php#resolution](https://www.london-electronics.com/LEL_New/panel-meter-information.php#resolution)

[NS72] AA.VV., Pocket calculators add up to a big market, New Scientist, n.805, 20 luglio 1972.

[PS18] sito Planet Sinclair, Sinclair Executive 1972-1973, <https://planetsinclair.meulie.net/calculators/executive.htm>

[Shi18] K. Shirri, Reversing Sinclair's amazing 1974 calculator hack - half the ROM of the HP-35, [http://files.righto.com/calculator/sinclair\\_scientific\\_simulator.html](http://files.righto.com/calculator/sinclair_scientific_simulator.html)

[VC18] Vintage Calculators Web Museum, <http://www.vintagecalculators.com/html/sinclair1.html>

[Voi18] Pagina web: <http://www.voidware.com/calcs/sinclaircambridge.htm>

# Console 8bit: Atari 7800

di Starfox Mulder

La console di cui sto per parlarvi ha per me un doppio valore affettivo. Da un lato è la mia console Atari preferita di sempre, dall'altro sono stato partecipe della sua riscoperta attraverso un piccolo saggio (presto edito da RetroEdicola Videoludica) in cui ho recensito tutti i suoi titoli: "Discovery Atari 7800".

Il Pro System (altro nome dell'Atari 7800) è il primo sistema marchiato Atari progettato da un'azienda terza, nello specifico la General Computer Corporation. Consci degli errori fatti con l'Atari 5200, dotato di controller scomodi e di difficile digeribilità per l'utente medio, ma soprattutto con la mancata retrocompatibilità nei confronti dei titoli del VCS, stavolta Atari volle che il tiro fosse corretto al meglio. Compatibilità totale quindi coi giochi dell'ammiraglia ed un nuovo controller che riprendesse l'originale joystick aggiungendo un secondo tasto azione per aumentare le possibilità ludiche. Il presidente Atari dell'epoca, quel Jack Tramiel proveniente da Commodore, voleva massimizzare i guadagni e sapeva che Atari era diventata celebre per le sue conversioni dei titoli Arcade, quindi l'imperativo era spingere in quella direzione, per quanto poi la storia ci avrebbe insegnato che il pubblico dei videogiocatori era cambiato. Il vero autogol di Tramiel fu però la sfiducia nei confronti del mondo console. Il 7800 era già pronto per essere venduto nel Maggio del '84, ma solo due anni dopo venne messa in commercio, dopo cioè una lunga gestazione fatta di conti non pagati (con la GCC) e investimenti su tutt'altro settore, ovvero quello degli home pc. Quando infine la console uscì, nel giugno del '86, gli avversari erano già sul mercato da abbastanza tempo per farla sembrare anacronistica e vetusta.

La console non fu neppure un fuoco di paglia, semplicemente non riuscì a bruciare nella maniera giusta proprio per il modo in cui i gusti generali erano cambiati e per un chip sonoro vetusto, trattandosi dello stesso utilizzato per l'Atari 2600. Atari aveva titoli storici in esclusiva, capolavori come Joust o Robotron 2084 che però all'epoca dell'uscita nei negozi dell'Atari 7800 erano già preistoria. Le sale giochi erano popolate da Gradius e Double Dragon, titolo quest'ultimo di cui si azzardò pure la conversione per la console Atari ma con risultati decisamente inferiori alle console della concorrenza (Nes e Master System in testa). Per lo più si insisteva su

versioni migliorate dei classici, giochi bellissimi che purtroppo già da tempo avevano smesso di interessare. Inutile pure la decisione di puntare sul costo di lancio concorrenziale di 140\$ per salvarla dal fallimento nei mercati di tutto il mondo.

Va specificato però che *non è tutto bruciato ciò che fa fumo* e da quel semi-insuccesso commerciale abbiamo oggi una delle console col più grande rapporto qualitativo che si possa sperare a livello di software. Dei 58 titoli rilasciati durante il ciclo di vita della console almeno 20/25 sono ottimi titoli e di vere brutture possiamo parlare giusto per una decina di giochi. Aggiungendo che la produzione dei titoli per 7800 ha ripreso da anni, grazie al sito atariage.com, sempre con risultati che oscillano dal buono all'eccellente (Frenzy e Pacman Collection sono tra i migliori giochi mai fatti per 7800) capirete che ci sono ottimi motivi per recuperare una di queste console. Dotandovi come me di un controller europeo ed una prolunga per lo stesso, vi fornirete di una delle migliori macchine mai ideata per l'esperienza arcade in ambiente domestico. Un vero gioiellino che se fosse stato rilasciato nei tempi previsti forse avrebbe avuto un destino migliore.

Alla prossima console!



## CARATTERISTICHE TECNICHE

<b>Produttore</b>	Atari
<b>Tipo</b>	Home Video Game Console
<b>Generazione</b>	Terza
<b>In vendita</b>	Maggio 1984
<b>Dismissione</b>	Gennaio 1992
<b>Supporto</b>	Cartuccia
<b>Unità vendute</b>	3,7 milioni

FONTE:

[HTTPS://EN.WIKIPEDIA.ORG/WIKI/ATARI\\_7800](https://en.wikipedia.org/wiki/Atari_7800)



XEVIIOUS



### Sfondi competitivi

Agli occhi di un giocatore moderno la grafica potrebbe apparire invecchiata particolarmente male ma per l'epoca la conversione da battere era quella fatta sul Nes di Nintendo. Chi vinse il confronto? Difficile a dirsi ma di certo gli sfondi erano ben più curati su Atari 7800.

#### GIUDIZIO SUL GIOCO

GIOCABILITA'

75%

Con il controller standard del 7800 rischiate di evocare tutti i santi immaginabili ma con il controller europeo (curiosamente emulo del pad Nes) le cose migliorano enormemente e la navetta risponde molto bene ai comandi, proiettandovi verso un entusiasmante competizione contro voi stessi o un eventuale giocatore 2 che si alternerà ad ogni vostra morte.

LONGEVITA'

70%

Quanto ci impiegherete ad annoiarvi nella ricerca del miglior record possibile?

# Xevious

Atari - Anno 1987 - Piattaforma Atari 7800

Dovete sapere che trovare un gioco da recensire per Atari 7800 rappresenta una vera sfida per me. Non solo amo la console ma sono fermamente convinto che almeno la metà della sua soft-teca meriti di essere giocata ed apprezzata. L'imbarazzo della scelta insomma, quel tipico impasse dettato da troppo coinvolgimento. Mi sono smarrito con la classica mossa del "chiedi al capo" ed ecco che il buon Francesco si è riservato di pensarci su un paio di giorni per poi venirsene fuori con un secco: parlatemi di Xevious! (*Così mi fai apparire un padre/padrone... quando in realta' sono molto peggio...* © N.d.F.F.)

Agli ordini signore, sono prontissimo e le faccio pure un bel paragone con l'arcade.

Classe 1983, figlia di quella Namco che non smise più di sfornare capolavori, la creatura nata dal genio di Masanobu Endo si distinse nelle sale giochi per aver accorpato tante idee in un mix davvero efficace. Il boss gigante da affrontare, i segreti (o easter eggs) da scovare all'interno dello schermo di gioco, il doppio sistema di sparo che ci permetteva di colpire sia i nemici in volo che quelli a terra. Ecco, almeno su quest'ultimo punto ci sono ottime probabilità che Xevious fece addirittura d'avanguardia al genere, ma in caso mi smentiate non c'è problema: sono ben conscio che la forza del titolo stava nel mettere tutte assieme queste caratteristiche.

A bordo della fidata navetta spaziale il nostro compito sarà quello di procedere all'interno dei 16 livelli di giochi, di uno dei primi shoot'em ups a scorrimento verticale della storia, con il fine di eliminare le orde aliene comandate dal biocomputer Gamp. Grafica colorata, nemici variegati e, come sempre accadeva nei titoli dell'epoca, una ricerca spasmodica del record. Già, perché quando dico che i livelli erano sedici non dovete prenderla come una speranza di poter completare il gioco dal momento che superato il sedicesimo stage il gioco ricomincia dal settimo e così in un loop infinito. Nell'arcade, oltre al joystick centrale ad 8 direzioni, i tasti a nostra disposizione erano due (ripetuti anche a sinistra per i giocatori mancini), uno per tipologia di sparo. I nemici a terra si vedranno colpiti da delle bombe che noi sganceremo dall'alto e per

simulare la cosa ci verrà dato un mirino che indicherà la parabola di caduta del colpo, di poco distante dal nostro mezzo.

Com'è andata quindi sul 7800? Il feeling è rimasto identico ma ovviamente si fecero moltissimi cambiamenti, necessari per convertire un'esperienza arcade su di una macchina ben meno potente. La grafica costrinse gli sviluppatori ad una compressione dello schermo ed un numero di pixels assai minore, ma già sul fronte sonoro fecero un mezzo miracolo (specie considerati i limiti del pro system). La licenza era già in mano ad Atari visto che furono loro a distribuirlo fuori dal Giappone, e per questa conversione l'azienda, allora in mano a Jack Tramiel, volle apportare un paio di modifiche al gameplay: un unico tasto del pad per sparare con entrambe le armi a nostra disposizione e la possibilità di scegliere tra quattro livelli di difficoltà. Il risultato è più che dignitoso considerato che la fluidità del gameplay resta sempre costante e riprendeva tutte le dinamiche che resero famoso il cabinato, dal già citato big boss ai segreti da scovare. Per chi possiede un 7800 questo è sicuramente un gioco da avere!

di Starfox Mulder





## THE DAWN OF KERNEL



## Comincia tutto da qui...

All'inizio del gioco il livello di difficoltà è veramente basso...



## La difficoltà cresce

Ma man mano che andiamo avanti il numero dei nemici incrementa e di conseguenza la difficoltà.



## Secondo livello

Siamo ormai infiltrati nella base K3R-NL e le cose cominciano a farsi interessanti.

## GIUDIZIO SUL GIOCO

## GIOCABILITA'

90%

Un gioco veramente carino, mai frustrante, dalla giocabilità innegabile e dal gameplay accattivante. Provatelo!

## LONGEVITA'

75%

La difficoltà è ben calibrata e non si rischia mai di rimanere bloccati. Anche una volta terminato avrete la voglia di rigiocarlo.

## The Dawn of Kernel

Juan J. Martínez - Anno 2018 - Piattaforma Amstrad CPC 464

Proprio di questi giorni è la notizia che **Juan J. Martínez**, prolifico programmatore spagnolo già apprezzato per alcuni giochi moderni per Commodore 64, ZX Spectrum ed Amstrad CPC, ha rilasciato un nuovo gioco per l'Amstrad CPC dal titolo di **The Dawn of Kernel**. Quale miglior occasione quindi di parlare nello stesso numero di RetroMagazine della serie Amstrad CPC e di recensire un gioco, per di più inedito!



Dopo una schermata statica in perfetto stile 8 bit il gioco ci accoglie con un messaggio piuttosto inquietante ricevuto dalla base K3R-NL nel sistema Leonis: un virus ha infettato di tutti i supporti vitali, forzando tutto il personale ad evacuare la base.

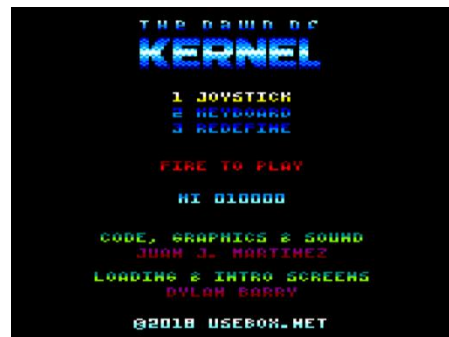
Alla fine del messaggio un sibillino KERNEL LIVES ci fa drizzare i peli dietro la schiena, svelando forse il reale problema che affligge la base K3R-NL.



Passata la doverosa introduzione accediamo al menu vero e proprio del gioco dove possiamo scegliere se controllare la nostra astronave con il joystick oppure con la tastiera. Nel caso scegliessimo la tastiera è possibile ridefinire i tasti oppure utilizzare la configurazione proposta (vedi tabella più avanti).

Ma parliamo del gioco! TdK ci mette alla guida di un'astronave che per proseguire nel gioco deve disattivare i campi di forza che la bloccano di volta in volta in ogni schermata. L'astronave si muove fluida tra i livelli ed il

fatto che si trovi soggetta alla forza di gravità della base, aggiunge il giusto livello di difficoltà ad un gameplay che risulterebbe altrimenti troppo semplice.



Il gioco è in pratica un flip-screen shoot'em up con elementi alla Cybernoid, forse un po' meno frenetico nell'azione ma decisamente più colorato. Non c'è bisogno di memorizzare i quadri per proseguire nel gioco, ma la conoscenza di come approcciare ogni schermata è decisamente d'aiuto per proseguire con il minor danno possibile. Ogni volta che veniamo colpiti da un nemico o da un missile, perdiamo infatti energia e, con una sola vita a disposizione, una volta terminata la barra energetica, dovremo inesorabilmente ricominciare da capo. Ma niente paura, alcune delle power station, una volta distrutte, rilasceranno ricariche energetiche o armi secondarie che ci torneranno molto comode per completare il gioco.

Non voglio svelarvi altro, il gioco è fin troppo carino per non essere provato almeno una volta e vi assicuro che vi *acchiapperà* sicuramente. L'unico appunto che mi sento di muovere è l'assenza di un sistema di salvataggio (...almeno una volta completato ogni livello...) che ci obbliga a finire il gioco tutto d'un fiato.

Azione	Tasto
Fuoco arma primaria	Barra Spaziatrice
Fuoco arma secondaria	↓ + Fuoco
Selezione arma secondaria	↓↓
Attivare Beam Pad	↓

The Dawn of Kernel può essere scaricato direttamente dal sito dell'autore:

<https://www.usebox.net/jjm/dawn-of-kernel/>

di Francesco Fiorentini

## THE TOWER OF EVIL



## Stanza del livello

Notate gli accessi alle stanze collegate, che in questo caso sono 4 ma non sarà così per tutte le stanze del piano.



## Un colore per ogni livello

Ogni livello avrà un suo colore di base diverso.

## GIUDIZIO SUL GIOCO

## GIOCABILITA'

90%

Gioco veloce. Personaggio semplice da comandare ed antagonisti decisamente "tosti per una esperienza di gioco immediata, semplice ma anche divertente, come nella migliore tradizione dei giochi anni '80.

## LONGEVITA'

65%

Una volta completato sarà difficile che troviate stimoli per giocarci nuovamente, a meno forse che vogliate raggiungere un punteggio più alto.

## The Tower of Evil

Creative Sparks - Anno 1985 - Piattaforma Commodore 16

Correva l'anno 1985 quando trovai questo gioco all'interno della cassetta acclusa alla rivista "Jackson Soft Compilation" numero 1. Il gioco girava su Commodore 16 e fu uno dei primissimi giochi che conobbi. Il gameplay è veloce e semplice da comprendere, (anche per un ragazzino di 12 anni o poco più, all'epoca dei fatti) e riuscii a completarlo senza "trucchetti" (anche perché allora non era facile ottenere tali informazioni e le mie competenze non erano ancora mature per fare da me) semplicemente giocando con la dovuta perizia ed attenzione!

Il gioco si sviluppa all'interno della "Torre del Male", una torre di 8 piani dove ciascun piano è composto da 9 stanze. In ogni piano vi sono dei tesori da raccogliere ed in più un elemento fondamentale, la "chiave" per passare al livello successivo. E' necessario quindi andare in giro per le varie stanze del piano, raccogliere i tesori ed in particolar modo la chiave, cercando di evitare il contatto con i mostriciattoli presenti in ciascun livello con il nostro personaggio. Abbiamo la possibilità di lanciare delle scariche per difenderci e, soprattutto, dalla nostra parte abbiamo la velocità ed il "colpo d'occhio", che ci permetterà di esaminare velocemente il contenuto di una stanza ed andare subito via evitando le attenzioni degli "inquinili" che si faranno sempre più tosti via via che si andrà avanti di livello.

Scopo ultimo del gioco? Restar vivi e conquistare la principessa che, alla fine dell'ultimo livello, se tutto andrà per il verso giusto, ci verrà incontro per dimostrarci tutto il suo affetto! Non Vi dico altro altrimenti rovino la sorpresa a chi di Voi si cimenterà nell'impresa di giocare e cercare di terminare il gioco... ☺

Per superare ciascun livello, bisogna trovare la "porta", dalla forma caratteristica, con due braccia che si aprono verso il basso descrivendo insieme una forma poliedrica, chiusa dall'alto, e bisogna entrare al suo interno con il nostro personaggio che dovrà essere in possesso della chiave che avrà raccolto durante la perlustrazione del livello di gioco. Se dovesse entrare senza tale chiave, il

nostro personaggio tornerà sullo stesso livello e non potrà lasciarlo finché non sarà entrato nella porta con in possesso anche la chiave.

Nel livello finale i "tesori" dovranno essere depositati all'interno di una cassaforte ed infine la principessa Vi verrà incontro.

Altro oggetto speciale degno di nota è il "calice della vita" che dona al nostro personaggio, il cui nome è Adros, una temporanea invulnerabilità. Vedrete il personaggio infatti "pulsare" finché l'effetto del calice non sarà cessato.

Per rendere il gioco un po' più difficile, alla perdita di ciascuna delle 5 vite a disposizione "Adros" dovrà cercare nuovamente la chiave, in quanto viene persa automaticamente insieme alla vita anche se fosse stata raccolta immediatamente prima. Altro elemento che ci renderà il gioco duro è il fatto che la posizione della "porta" e di tutti i tesori nel livello cambia ogni qualvolta perdiamo una delle vite a nostra disposizione.

I tesori in nostro possesso saranno visualizzati via via nella parte alta dello schermo. E' possibile giocare da soli oppure in due giocatori. Il personaggio può essere controllato via joystick oppure con gli appositi tasti sulla tastiera (";" per andare verso l'alto, "/" per procedere verso il basso, "z" verso sinistra ed "x" per andare verso destra. Spazio per lo sparo). Lo sparo è sempre in direzione del moto di "Adros". E' possibile anche sparare tenendo fermo il personaggio.

Che dirVi di più? Il gioco conquista per la sua semplicità e, nel contempo, riesce a dimostrarsi difficile ed avvincente via via che si sale di livello, tanto da farVi preferire spesso di evitare lo scontro con i temibili antagonisti quando non è assolutamente indispensabile. Uno dei pochi titoli che, secondo me, sfrutta adeguatamente le risorse del Commodore 16, notoriamente non eccelse.

Provare per credere.

di Marco Pistorio

DOCTOR HAUZER



## Angolo Oscurita': Doctor Hauzer

River Hill Software - Anno 1994 - Piattaforma Panasonic 3DO

Quando mia moglie mi ha visto giocare a Doctor Hauzer si è lasciata andare in una considerazione sulla grafica invecchiata malissimo e subito dopo ha domandato "Ma ha a che fare con il Dottor House?" In effetti no, i ragazzi della River Hill Software sono arrivati con 10 anni di anticipo sulle avventure del dottore più scorbuto della TV e l'ambientazione è di tutt'altro genere.

Siamo ormai abituati all'idea che una formula vincente venga ripetuta all'infinito, basti ricordare che dopo Resident Evil uscirono rapidamente titoli come Deep Fear, Dino Crisis, Silent Hill e tutta una serie di survival che imitavano le dinamiche del clamoroso gioco Capcom. Quasi tutti sappiamo anche che RE non fu il primo survival vero e proprio ma tra i tanti giochi a cui l'opera di Shinji Mikami doveva render merito, uno dei più celebri fu quell'Alone in the Dark di Infogrames uscito nel 1992. Ma in quei 4 anni che trascorsero tra l'avventura di Edward Carnby/Emily Hartwood e quella di Chris Redfield/Jill Valentine, non accadde nulla?

Accadde Doctor Hauzer.

Il gioco uscì come esclusiva 3DO nel solo Giappone e non vide mai nessuna conversione su altri sistemi, motivo per cui fino a poco tempo fa giocarci significava conoscere almeno un poco di ideogrammi, oppure avere tanta pazienza con google translate alla mano. Da due anni a questa parte invece, grazie al lavoro di alcuni appassionati, è possibile giocare l'intera avventura tradotta in inglese, con risultati magari amatoriali ma ben comprensibili. Le dinamiche sono effettivamente le medesime di Alone in the Dark, portandoci a controllare un personaggio in 3D che si muoverà all'interno di una magione "infestata", ma le differenze ci sono e garantiscono un certo entusiasmo per il (brevissimo) tempo che vi richiederà concludere il tutto.

Siamo nel 1952, il giocatore interpreterà un investigatore chiamato ad indagare sulla scomparsa del Doctor Hauzer, celebre archeologo di cui recentemente non si ha più notizia alcuna. Il nostro decide di iniziare le indagini direttamente dalla casa del dottore in

questione è la scelta si rivelerà particolarmente saggia (e pericolosa) per due principali motivi: 1- perché in effetti la risposta all'enigma si trova proprio in quella magione; 2- perché la casa stessa è stata costruita dal dottore intorno al sito archeologico su cui stava lavorando, cosa che ha portato la dimora stessa a diventare....strana.

Qui però mi fermo, non amo anticipare troppo della trama dei giochi.

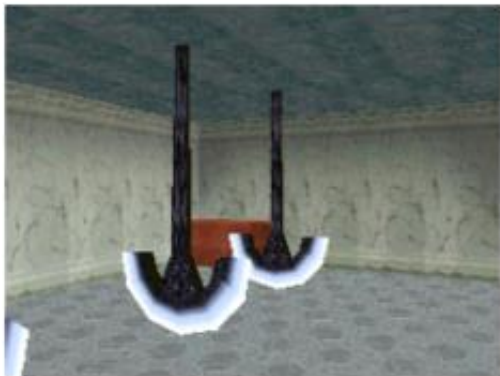
Il controller del 3DO, come sempre ottimo, si rivela ben sfruttato per la gestione del gameplay. La croce direzionale ci permette di muovere il nostro personaggio, che esattamente come in tutti i titoli dell'epoca potrà andare solo avanti ed indietro, utilizzando le direzioni destra e sinistra per far ruotare sul posto il nostro alter-ego prima di avviarlo.

I tre tasti principali sono dedicati rispettivamente ad "Azione" (A), Tira/Spingi/Corri (B) e Salta (C). Quest'ultima funzione è abbastanza rara nei survival ma si rivelerà assai utile invece per evitare di morire nei vari trabocchetti disseminati per la casa. Il tasto P permette al contempo di mettere in pausa ed accedere all'inventario, dove oltre a selezionare gli oggetti che vorremo utilizzare tra quelli raccolti lungo l'avventura ci sarà sempre possibile salvare, cosa che consiglio di fare spesso. Il gioco non presenta in sostanza alcun tipo di scontro.

Niente zombie o mostri alati come se ne vedevano a Derceto (magione di Alone in the Dark), sebbene in compenso la casa stessa farà di tutto per ucciderci. Stanze che si allagano dopo averci chiuso dentro, pavimenti che crollano, lampadari che ci finiscono in testa, massi pronti a schiacciarci e chi più ne ha più ne metta.

Una quantità così enorme di modi per morire dovuti all'ambiente di gioco rischiava di creare una grandissima frustrazione negli utenti dell'epoca data la ben nota difficoltà nella gestione delle telecamere dei primi titoli in 3D, motivo per cui alla River Hill decisero di fornire il gioco di ben 3 telecamere tra cui "switchare" con la semplice pressione del

## DOCTOR HAUZER (PAGE 2)



tasto L del pad. La prima visuale è quella classica da survival mentre la seconda ci porterà a vedere il tutto dall'alto, come si trattasse di un adventure top-down alla Legend of Zelda. La terza visuale infine, la più utile per guardarsi attorno ma la peggiore per muoversi, è quella in prima persona. In quest'ultimo tipo di inquadratura sarà utile prendere anche dimestichezza col tasto R che ci permetterà di ruotare la testa senza muoverci, opzione completamente inutile nelle altre visuali.

Come avrete intuito, morire è facilissimo e sommato alla mancanza di colonna sonora (cosa che ci porterà ad ascoltare solo i nostri passi o gli effetti sonori che ci circondano) avremo gli elementi perfetti per creare un climax di paura old-style gustosissimo. Nulla di seriamente spaventoso, sia chiaro, ma nei primi anni 90 penso proprio sarebbero bastati per farmi addormentare con difficoltà.

Seguendo la logica del "prova a parlare bene anche dei difetti" vi potrei dire che è un titolo perfetto da godere in un pomeriggio di pioggia invernale, ma la verità è che non ci riempirete neppure il tempo tra la merenda e l'aperitivo per concluderlo: il gioco è corto e per giunta troppo lineare. Io che sono avvezzo al genere l'ho terminato in un'ora e me la sono presa comoda perché spesso morivo apposta per vedere le animazioni (sempre salvando preventivamente).

Se fosse in vendita nei negozi vi direi di starne alla larga poiché qualsiasi fosse il prezzo non

meriterebbe l'acquisto per il tempo che vi concederà, ma per fortuna è scaricabile gratuitamente (ormai è a tutti gli effetti un abandonware quindi nulla di illegale), tradotto ed immediatamente giocabile su un qualsiasi 3DO che, come forse saprete, non necessita di alcuna modifica per leggere le copie di backup.

Emularlo? Fattibile, solitamente i risultati per la macchina di Trip Hawkins sono abbastanza decenti, ma non metto le mani sul fuoco perché l'ho affrontato direttamente da console.

Sarà pure stato un fuoco di paglia ma per quel che ha bruciato mi ci sono divertito un sacco!

*di Starfox Mulder*

## GIUDIZIO SUL GIOCO

## GIOCABILITA'

70%

Per i ritmi moderni è oltre il legnoso, eppure l'utilizzo delle tre diverse telecamere fu una grandissima trovata, così come la scelta di incentrare tutto (o quasi) sull'esplorazione.

## LONGEVITA'

30%

Si finisce in un'oretta circa. Se siete poco avvezzi al genere magari ci impiegherete di più ma immaginandomi lo sventurato giapponese che lo pagò a prezzo pieno, diciamo che deve averla presa parecchio male all'epoca.

RIMRUNNER



*Ah, ancora i cari, vecchi, amati insettacci che combatto già da mezz'ora, sembrano proprio nati per essere eliminati!*



# Rimrunner

Palace Software - Anno 1988 - Piattaforma Commodore 64

Quante volte nella giornata ci siamo trovati, anzi, oserei dire ritrovati, il cellulare in mano, come in preda ad un automatismo, con l'intento di completare un livello di Candy Crush Saga (e suoi simili)? Questa dinamica nasce sovente in quei momenti che proprio non vogliamo pensare a nulla, o siamo in attesa di un appuntamento o che stiamo aspettando un mezzo di trasporto, eccetera.

Proviamo allora a catapultarci da questa pacifica situazione colorata e caramellosa ad un mondo alieno, contemplato e sviluppato nei sani e mansueti anni ottanta, precisamente nell'anno 1988. Accogliamo Rimrunner, un breve videogioco che a prima vista sembrerebbe di dubbia importanza. In realtà non deve essere assolutamente cestinato o dimenticato perché in quegli anni svolgeva un compito pressoché simile all'odierno ed acclamato Candy Crush.

In Rimrunner il protagonista è un essere appartenente alla razza aliena degli Insettoidi, a cavallo di un dinosauro simile ad un Velociraptor. Essendo un guerriero d'élite, in questa colonia, dovrà proteggere i propri simili dall'invasione, correndo lungo i perimetri chiusi della colonia stessa, in senso orario ed antiorario, in modalità loop-line.

L'ambiente di gioco si sviluppa attraverso la classica modalità a scorrimento orizzontale, con libertà di cavalcare in entrambe le direzioni. In totale le colonie sono tre, con differenti e meravigliose ambientazioni, scenari dotati di gamme cromatiche magiche, includendo pietre e vegetazione di egual accuratezza.

L'alieno, sebbene dotato di arti, a prima vista esili, gode di numerosi pregi. Primo tra tutti, un ottimo rapporto di fedeltà ed amicizia col suo mezzo di locomozione: l'insetto fischia, il velociraptor sorridente (vedi la locandina) corre da lui. Per caso un meteorite uccide il fido velociraptor? Nessun problema! Un fischio e ne arriva un altro ed un altro ancora, ben quattro dinosauri per ogni colonia da difendere. Questo legame pressoché indissolubile è sottolineato anche dal medesimo colore di carnagione: un bel verde lucertola.

Scherzi a parte, il fischio di chiamata generato dall'insetto, è un capolavoro di precisione del nostro amato chip sonoro del Commodore. Quando il protagonista, a causa di ripetuti colpi inferti dai nemici, cade disarcionato, immediatamente si rialza e si gratta il capo con una gestualità perfettamente riuscita. Quando il dinosauro muore, sia la scena della caduta, che il lampeggiare della sua spoglia (ed ossa) in dissolvenza, sono incredibilmente accurati. Gli sfondi e la musica sono un tripudio di precisione ed evocano con estrema

efficacia il mondo alieno, rendendolo forse meno inospitale di quanto sembri. Sebbene cadano senza sosta meteoriti, orribili insetti, brutti ragni, missili...

L'audio dei vari generatori di ricarica, il calpestio del terreno ad opera del dinosauro e la carica del fucile, sono piccole opere di grande perfezione. E' Impossibile ignorare tutti questi preziosi elementi. In effetti, personalmente parlando, il ricordo mi è rimasto nitido a distanza di quasi un terzo di secolo. Ammettiamolo, Steve Brown, presso la Palace Software, aveva creato un altro piccolo capolavoro.

Questo videogioco, per vari motivi, probabilmente non ha meritato l'ingresso nel quartier generale delle titaniche ditte come la Thalamus, però bisogna ammettere che negli anni del suo sviluppo, la guerra e la guerriglia degli otto bit, era sostanzialmente carica di esplosivo. Il porting era selvaggio. Copyright e licenze scalzavano il posto delle vendite al puro godimento del "non indispensabile". Questa pentola a pressione era veramente carica, pronta a cedere nella struttura e portare l'hardware molto lontano da essa, come il mondo alieno di questo videogioco, trascinando con timori e speranze, tutti quanti gli argomenti dell'informatica. Per questo motivo Rimrunner è il Candy Crush di quel tempo: vestiva le più pure proprietà del termine "passatempo", annunciando altresì la fine di quell'epoca magica.

Il grande Steve Brown aveva precedentemente creato il suo più famoso capolavoro, intitolato Barbarian. Quindi mi chiedo, Rimrunner potrebbe rappresentare le idee più intime dell'autore, che non potevano venire alla luce attraverso una seconda opera monumentale? Oppure nel suo progetto vi erano idee così differenti tra loro, che non avrebbero mai portato alla nascita di un lavoro completo a meno di essere membro di un agguerrito staff? Oppure, forse i tempi non erano maturi, per affrontare il tema, probabilmente cestinabile, del videogame "da passeggio" come oggi lo è Candy Crush?

Ricordiamoci bene che Rimrunner è stato sviluppato presso gli anni 80 e la condivisione di conoscenza dei pochi pionieri eletti dell'informatica, giaceva in situazioni poco rassicuranti e rapidamente copiabili. Oggi tra motori grafici, copyright, staff di indiscutibile potere come Sony, quartieri di sviluppo software realmente militarizzati, siamo lontani parsec su parsec da quei romantici anni. Questi elementi di disagio social-multimediale degli anni ottanta, però, non hanno minato la creazione di un videogame ambientato in un pianeta alieno, misterioso, sconosciuto, un poco angosciante.

## RIMRUNNER (PAGE 2)



## GIUDIZIO SUL GIOCO

## GIOCABILITA'

70%

Spara in alto, spara in orizzontale, spara in diagonale, tutto ciò è fantastico, un gioco rilassante con nemici infiniti, a cavallo di un sorridente dinosauro. La giornata è andata storta? No problem il fido fucile la migliorerà notevolmente, distruggendo asteroidi, ragni e alieni vari! Il videogioco è stato sviluppato per un solo giocatore. Purtroppo la funzione "due giocatori" non è mai stata implementata. Le piattaforme utilizzate sono state il Commodore64 e 128. Era stato programmato anche per Amstrad CPC, ZX Spectrum e Atari ST ma ad oggi purtroppo non si hanno notizie ufficiali del porting. Sicuramente questi aspetti hanno indebolito l'ascesa al trono dei grandi videogiochi.

## LONGEVITA'

30%

La cassetta veniva venduta a 19.000 Lire, il floppy invece 25.000 Lire. Queste cifre, di tutto rispetto, rispecchiavano un gioco difficilmente comparabile all'odierna evoluzione dei passatempo sparattutto, specialmente quelli dotati di poche pretese. Associarlo a Candy Crush, sebbene condivida alcuni aspetti, rimane come un puro ossimoro. Infatti possiamo fare una veloce partita con le caramelle mentre aspettiamo l'autobus ma sicuramente non potevamo, al tempo, fare altrettanto con questa formica Insettoide: serviva un tavolo, un televisore, una costosa cassetta, joystick ed ovviamente tastiera e del tempo libero. Questi elementi ne hanno decretato una breve ed alquanto perturbata esistenza!

Questi aggettivi sono giustificati da uno scorrimento grafico dello sfondo in modalità parallasse. Inoltre, presso le colonie, non vi è alcuna traccia antropomorfa degna di nota. Le uniche razze presenti nel gioco sono il protagonista, il fido dinosauro intercambiabile ed i nemici Aracnoidi, che mirano a spodestare la razza Insettoide ed appropriarsi della colonia.

Il compito tattico del protagonista non è particolarmente complesso, la colonia di Insettoidi è circondata da generatori di campo che vanno caricati e stabilizzati per tutto il tempo della missione. I nemici sono pressochè infiniti e il tempo scorre rapidamente mentre la barra digitale dell'energia prosegue nel proprio inesorabile decadimento che conduce al fallimento della missione, oppure fino alla morte del protagonista o del suo mezzo di locomozione.

Nella parte alta dello schermo c'è la mappa del perimetro che ci mostra la nostra posizione. Quando un elemento lampeggia nella mappa, significa che quel generatore di campo necessita di essere ricaricato: dovremo raggiungerlo quanto prima. Questi generatori servono a mantenere i nemici a debita distanza. Durante la cavalcata, dobbiamo mantenere tutto ben carico ed eliminare Aracnoidi e meteoriti infiltrati lungo il perimetro esterno dei generatori. Solo così potremo avanzare di livello. Se i generatori si indeboliscono eccessivamente, i nemici entreranno copiosamente attraverso le barriere.

E' un vero peccato cestinare questo videogioco nel dimenticatoio. Lo sviluppo è stato curato dalla Binary Vision e il design pressochè interamente da Steve Brown. Dovremmo ricordarlo come un ottimo lavoro anche per le numerose perle di accuratezza e bellezza. Difatti, la musica all'inizio della partita, è molto orecchiabile (di Richard Joseph). Poi, oltre che essere un egregio passatempo, si può ricordare per le particolarità della tabella di "high score". All'inizio del livello infatti, il protagonista è in piedi a fianco della tabella e carica il fucile a pompa con estrema nitidezza. E' curiosa anche la modalità di inserimento del proprio nome nella tabella dei punteggi: l'insettoide viene direzionato attraverso il joystick per sparare sulle lettere selezionate dall'utente. Scelta molto simpatica!

Le recensioni del tempo sono state magnanime con questo videogioco? Soltanto CVG Reviews n°79 elargì un gratificante 9 su 10; più precisamente, sia la grafica che la giocabilità, sono state valutate con un bel 9, mentre audio e valore del gioco con un 8. Le altre riviste sono state molto più severe. Ricordiamo comunque che questi anni infuocati di porting, copyright e veleni, non davano spazio al candido e puro entertainment né al passatempo: sugli scaffali contavano solo le vendite dei grandi titoli e la guerra tra i colossi.

Riassumendo i punti deboli del gioco. Possiamo anche ragionare per ore ed ore sul delicato tema

che gli Insettoidi alieni colonizzano pianeti sperduti negli spazi interplanetari disseminati nella galassia e vengono attaccati continuamente dai nemici Aracnoidi, eccetera, eccetera! Niente da fare. Purtroppo questo permane come un grave punto debole, poiché vige poca fantasia. Inoltre il protagonista ci obbliga a tenere il dito sul pulsante di fuoco in maniera continuativa. Non è una buona cosa, prima o poi arriverà il "crampo da frullatore" che avviene quando pigiamo il famoso tasto degli elettrodomestici che funzionano unicamente con un pulsante tenuto premuto. Il problema è che l'ingegnere dell'elettrodomestico, sa bene che, con la stanchezza del dito, coinciderà il riposo del motore elettrico progettato per un uso moderato. Invece in Rimrunner? Nemici infiniti. Nessuna pausa. Grave discrepanza.

Altro punto debole, grande staticità di gameplay, oltre che essere scaraventati a terra da nemici ed asteroidi, che si fa? Poco e nulla! A volte si scende dal dinosauro per le ricariche dei generatori, altre volte i nemici ci lanciano contro dei missili traccianti. Insomma manca una buona dose di fantasia per poterlo annoverare tra i grandi videogames.

Punti di forza? La tabella high score è veramente simpatica, sparare sulle lettere è intrigante come spostare le caramelle di Candy Crush! La grafica è sovrana, sprite e gamma di colore eleganti ed affascinanti. L'audio e gli effetti sonori sono curati in maniera maniacale, personalmente parlando, il voto più alto lo darei a questi ultimi elementi.

Concludo con una domanda provocatoria. Perché milioni di utenti utilizzano un "passatempo" simpatico, colorato, grazioso, oserei dire "petaloso" come Candy Crush e noi dovremmo dare un punteggio basso a Rimrunner? Io propongo invece di cavalcare il nostro fedele Velociraptor e sparare qua e là, in alto e in basso, a casaccio, in un mondo alieno. Non trovate che possa trasformarsi in un ottimo passatempo dopo una stressante giornata? Allora vi invito a detenere, legalmente, una copia di Rimrunner ed utilizzare la ROM su Android, magari giocandoci in emulazione mentre abbiamo cinque minuti di tempo libero!

Vi anticipo che nei prossimi numeri inizierò a parlare dei grandi giochi del passato, sviluppati in ambiente vettoriale. In questo numero l'intervista con Fabrizio Radica, ha riaccessi in me i magici ricordi del fascino vettoriale, puro, preciso, sensuale, dove poche linee sfociavano in grande immaginazione. Vi parlerò di Stellar7, un big del passato che dovrebbe entrare nell'ipotetica rubrica di RM "ricordarsi di non dimenticare", quindi, a presto cari lettori!

di Michele Ugolini

PSYCHO NICS OSCAR



## Giocatore uno pronto!

La schermata iniziale che precede l'inizio del gioco



## Un pò Turrigan un pò Super Mario

Oltre ad eliminare robot ed astronavi dobbiamo saltare da una piattaforma all'altra come Mario e Sonic



## Ci siamo quasi

Ecco il terribile boss finale, dopo il quale termina la nostra missione e possiamo far ritorno sulla terra

GIUDIZIO SUL GIOCO

GIOCABILITA'

90%

LONGEVITA'

80%

# Psycho Nics Oscar

Data East - Anno 1987 - Piattaforma Arcade

Nel panorama dei videogiochi arcade e delle console degli anni 80 e 90 ci sono titoli che sono vere e proprie pietre miliari che, anche se non li abbiamo giocati, basta solo nominarli per richiamare alla nostra mente immagini ed emozioni. Così se cito il nome Turrigan sono sicuro che in tutti noi emerge il ricordo di quel bellissimo e frenetico platform futuristico che è stato un successo un pò su tutte le piattaforme, dal c64 passando per lo Snes e il Megadrive.

Ma sono altrettanto sicuro che pochi sanno che dietro il successo di questo gioco si nasconde un titolo che ha avuto minor fortuna ma che è stato importantissimo per averlo ispirato e che meriterebbe oggi di essere riscoperto.

Infatti nel 1987, cioè due anni prima del lancio del primo Turrigan, la famosa Data East fece uscire nelle sale giochi un titolo metà platform e metà sparattutto dal titolo Psycho Nics Oscar che con il suo gameplay ha sicuramente influenzato gli autori del suo fratello maggiore. Lo so che il titolo sembra più il nome di una squadra di basket, ma vi assicuro che Oscar, chiamato semplicemente così dai suoi fan, è un gioco che merita davvero di essere rivalutato e per questo ve ne voglio parlare.



In questo titolo vestiamo i panni, forse è meglio dire l'armatura, di un robot che è stato mandato su un lontano pianeta che l'uomo vuole colonizzare ma per fare ciò dobbiamo eliminare i mostri e gli alieni che lo popolano. In questa terra ostile possiamo muoverci e saltare da una piattaforma all'altra e come arma abbiamo in dotazione un fucile che inizialmente sparerà un semplice laser di breve gittata.

Ma alcuni nemici che uccidiamo lasciano delle capsule con incisa una lettera N che, una volta raccolte, attivano una barra di potenza in perfetto stile Gradius che mostra cinque diversi power up con i quali possiamo migliorare abilità e armi del nostro robot.

Se attiviamo infatti il primo slot miglioriamo la capacità di salto, con il secondo e terzo ci

equipaggiamo rispettivamente di granate e missili, con il quarto attiviamo oggetti rotanti che aumentano la nostra potenza di fuoco e con l'ultimo, sicuramente il più importante per avanzare nel gioco, miglioriamo la qualità e la gittata del nostro laser.

Ogni power up, oltre ad essere combinabile con gli altri, può essere attivato più volte per un livello sempre maggiore. Inoltre è possibile abilitarne ulteriori inizialmente nascosti come quello che migliora l'armatura e la resistenza ai colpi del nostro eroe.



Prima di poter liberare il pianeta dobbiamo attraversare sei diverse aree ricche di foreste, grotte e fabbriche metalliche piene di nemici con diverse caratteristiche e con il classico gigantesco boss che ci aspetta alla fine di ogni livello.

Oltre alle capsule per potenziare il robot, alcuni nemici possono anche lasciarne delle altre con incisa la lettera K. Se ne raccogliamo venti avremo una vita extra.

La grafica di Oscar è molto colorata e dettagliata anche nei fondali e il sonoro, pur non avendo una soundtrack di sottofondo da oscar (permettetemi il gioco di parole), è adatto ad ogni momento del gioco. Il gameplay poi è davvero ben fatto e, dopo aver preso un po' di dimestichezza con le meccaniche di gioco, è davvero un piacere muovere il nostro robot in questa avventura. Come detto prima Oscar presenta degli elementi che poi sono stati ripresi da Turrigan e a mio avviso anche da un altro grande successo come Metroid. A differenza però di questi due titoli, Oscar è meno frenetico e segue lo schema classico dei platform del suo periodo. Questo lo rende molto piacevole da giocare anche perché non si tratta di un gioco impossibile da terminare.

Il mancato successo di Oscar sta solo nel fatto che non ha avuto una conversione casalinga per le varie console dell'epoca e questo lo ha reso un gioco di nicchia del mondo delle sale giochi.

Querino Ialongo

# RetroGiochiAmo: Impossible Mission

di Daniele Brahimi

Nel numero 8 di RetroMagazine abbiamo lanciato la rubrica **RetroGiochiAmo**, dedicata alla riscoperta dei giochi che non abbiamo avuto la possibilità di giocare o di portare a termine quando eravamo ragazzi. Contemporaneamente, nell'articolo di chiusura, abbiamo invitato i nostri lettori a collaborare con noi scrivendo di un gioco che hanno particolarmente amato o che hanno portato a termine solo dopo tanto tempo. Ebbene l'invito non è caduto nel vuoto, **Daniele Brahimi** ci ha scritto offrendosi di creare una guida per un gioco che definire epico e' riduttivo.

Beh, basta chiacchiere, la parola a **Daniele**.



Nulla è impossibile!

Frase che ho ripetuto qualche sera fa dopo aver completato questo meraviglioso titolo targato Commodore 64 che uscì nel lontano 1984. All'epoca avevo solo undici anni quando lo ricevetti e una volta partito il gioco non seppi minimamente cosa bisognava fare a parte gironzolare nelle stanze evitando gli odiosi robot e fare su e giù con l'ascensore, ma come dicevo qualche sera fa, decisi di rigiocarci grazie all'emulatore e decisi di portarlo a termine una buona volta e così è stato. Dopo tre lunghissime ore finalmente riuscii a comporre il puzzle finale fermando così quel simpaticone del Dr. Atombender.

Visto che non pochi di voi hanno avuto difficoltà con questo gioco ed in particolar

modo con la parte finale, ho deciso di darvi qualche aiuto e consiglio per far sì che anche voi lo possiate riprendere da dove lo avevate lasciato e finalmente finirlo!



Cominciamo: sappiate che ad ogni partita la collocazione delle stanze cambia ed anche quella dei robot, di conseguenza non so quanto convenga seguire passo a passo le videosoluzioni, a meno che la vostra partita corrisponda nei minimi dettagli al video pubblicato.

Il vero nemico di questo gioco non sono né il robot né il pazzoide del Dr Atombender, bensì la fretta! Se in questo gioco avrete fretta sarete sconfitti in partenza.

Appena entrate in una stanza, esaminatela, studiatela tutta (arredamento, posizione e movimento dei robot) prima di agire e soprattutto non dimenticate di esaminare tutti gli oggetti. Perustrate le stanze secondo l'ordine in cui si presentano dal basso verso l'alto o viceversa come preferite, in modo da non dimenticarvene e non tornare in un secondo momento sempre se le condizioni non lo richiedano espressamente. Quando trovate i codici per fermare i robot e riportare gli ascensori nella posizione originale non sprecateli perché nel gioco sono davvero pochi, usateli con saggezza.

Avete sei ore di tempo reale per completare il gioco che potrebbero sembrare molte ma in realtà non lo saranno visto che ad ogni vita

persa l'orologio vi toglierà dieci minuti. Una volta recuperati tutti i pezzi del puzzle finale, mettetevi nel corridoio appena fuori l'ascensore che porta alla stanza del Dr Atombender (quella con la porta) e cominciate così a comporre il mosaico.

Ciascun mosaico comprende quattro pezzi dai quali gli potete cambiare colore/invertirli sotto sopra e da destra a sinistra. Cominciate con un colore, estraete un pezzo ed esaminatelo attentamente per trovare quello dal quale combacia. Una volta trovati i quattro pezzi ed aver composto il primo mosaico dovrebbe scomparire, se ciò non dovesse accadere provate a cambiare colore, inversione ecc. Più andrete avanti con la composizione più i pezzi diminuiranno e di conseguenza sarà più facile trovare quelli mancanti.

Ah dimenticavo, nemmeno lì i video tutorial vi saranno d'aiuto perché quelli che avevo visto io erano molto diversi, probabilmente di un'altra versione idem sui siti. Poi scusate, non c'è più gusto a finirlo senza aiuti esterni? Questo che vi sto dando io non basta forse? Vi posso assicurare che una volta terminato vi sentirete come veterani del Vietnam e vi verrà voglia di urlare in piena notte questa frase che sto per dirvi prima di darvi i più cari saluti a voi tutti retrogamer di qualsiasi piattaforma: Nothing is impossible!!!

Ringraziamo Daniele per la sua guida e per i preziosi consigli che ci permetteranno di sconfiggere il malvagio Dr. Atombender.

Parlando però di Impossible Mission, probabilmente uno dei giochi più famosi del Commodore 64, se non il più iconico in assoluto, non potevamo pubblicarne la guida senza un'adeguata recensione.

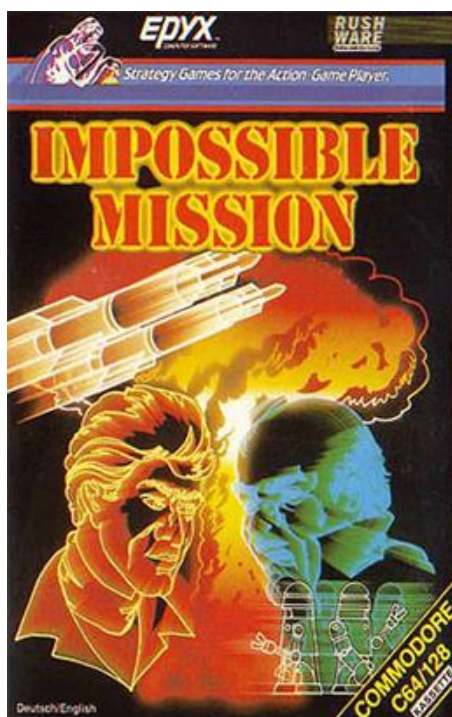
Che cosa state aspettando quindi?

Se non lo avete ancora fatto, andate alla pagina successiva per leggere la recensione di Impossible Mission che **Dante Profeta** ha scritto per noi!



# Impossible Dennis Caswell Mission

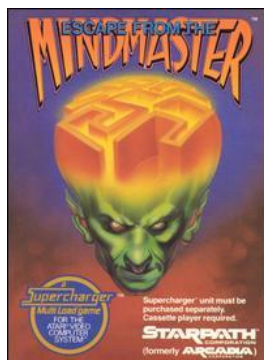
di Dante Profeta



Tutte le azioni che facciamo sono destinate ad avere ripercussioni nel tempo. Chissà se Dennis Caswell nel 1983 ne era cosciente. Probabilmente

no, come gran parte di noi. È certo però che le sue azioni sarebbero state destinate ad avere un eco per lungo tempo, tanto che siamo qui nel 2018 a parlarne ancora.

Era il 1981 quando un giovane Ingegnere



Informatico, fresco fresco di Master, venne assunto dal fratello del suo compagno di stanza, uno dei fondatori della Starpath.

La Starpath, inizialmente Arcadia, per chi non ne avesse memoria ha dato un enorme contributo all'Atari 2600 introducendo sul mercato il Supercharger, una speciale cartuccia in grado di caricare i giochi da musicassetta collegando l'uscita audio mono di un comune mangianastri (nome comune, ormai dimenticato, del famoso registratore a cassette).

Phaser Patrol ed Escape from Mindmaster, che assieme al Supercharger e ad altri titoli fanno bella mostra in uno scaffale del mio studio, basterebbero da soli a osannare Dennis Caswell. Ma Caswell si è spinto ben oltre regalandoci un capolavoro videoludico ormai senza tempo: Impossible Mission per Commodore 64.

Di lì a poco la Starpath, sull'orlo della bancarotta a causa del ben noto collasso del mercato delle console di seconda generazione, venne acquisita dalla mitica Epyx, e Dennis Caswell con lei.

La progettazione di Impossible Mission fu per lo più modulata e affinata durante la codifica, interamente in Assembler come era uso e necessità, in barba alle tecniche di ingegnerizzazione del software che prevedono in primis un'analisi dettagliata del progetto, relegando la programmazione come atto finale del processo di sviluppo.



Neanche il nome era stato abbozzato, tutto ebbe inizio dal disegno su carta e dalla codifica della splendida animazione del personaggio principale del gioco, che avrebbe da quel momento in poi alzato l'asticella e si

sarebbe imposto come pietra di paragone per qualunque animazione futura.

Senza strumenti software di supporto per la realizzazione della grafica, Impossible Mission prese vita dapprima su carta da disegno per essere riportato in Assembler come sequenze di stringhe esadecimali.

I fotogrammi di animazione del personaggio principale che corre prendevano così tanta memoria che non era possibile replicarli ribaltati in memoria per farlo correre nella direzione opposta. Per ovviare a questo inconveniente Caswell scrisse una routine in Assembler che ribaltava i pixel a runtime. Tuttavia, era estremamente onerosa in termini computazionali tanto da lasciare al gioco solo una parte delle potenzialità di calcolo del Commodore 64. Ma poiché il motore del gioco non richiedeva troppi cicli di clock per essere eseguito, il ribaltamento a runtime appariva estremamente fluido, come se i fotogrammi risiedessero in memoria per come venivano visualizzati - "Reversing the running man sprites did take a great deal of the processor's time. Fortunately, Impossible Mission was not otherwise computationally intensive, so there was enough time to spare."

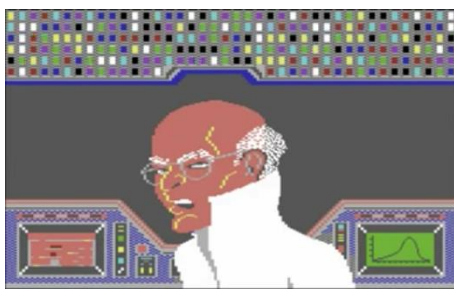
Ho deciso di riportare l'intera sua frase perché è per me emozionante e dovrebbe essere di monito a chi ritiene che un gioco abbia bisogno di un hardware potente per essere divertente, avvincente, ed essere definito un capolavoro. Di certo Impossible Mission è un capolavoro nel suo genere, tanto da ricevere conversioni e adattamenti periodicamente nel corso degli anni fino a quello per Nintendo DS.

Sebbene l'idea del gioco, in particolare quella del super computer che controlla la casa-laboratorio, venne in mente a Dennis Caswell guardando il film War Games, l'ambientazione è delle più classiche jansbondiane, con tanto di super cattivo, qui noto come Elvin Atombender, dedito alla conquista o alla distruzione del mondo, a scelta. L'arcifamosa voce campionata che dà inizio al gioco venne recitata da un

dipendente della Electronic Speech Systems, che per volere dello stesso Caswell doveva essere di mezz'età, simile a un nemico mortale di 007. Scusatemi, ma non riesco a resistere alla tentazione di trascriverla... "Another visitor. Stay awhile... Staaaaay FOREVER!".

La Electronic Speech Systems non solo fornì la voce, ovviamente, ma fornì anche il software per la riproduzione dei campioni, che venne inglobata nel gioco solo verso la fine dello sviluppo.

Anche il nome Impossible Mission venne attribuito alla fine dello sviluppo, e non è un mistero che sia stato preso in prestito dalla ben nota serie televisiva Mission: Impossible, con le due parole che lo compongono invertite di ordine per evitare di essere messi sotto accusa di violazione di copyright.



#### La trama

Il giocatore impersona dunque un agente segreto senza nome né numero di matricola, che infiltratosi nella fortezza sotterranea che funge da casa-laboratorio del professor Atombender, dovrà recuperare la parola chiave d'accesso del super computer per disinnescare la bomba e porre fine alle mire folli di Elvin.

La chiave d'accesso è scritta su un foglio smembrato in più parti, che a detta di Caswell, rappresenta una scheda perforata. Il rimettere insieme la scheda perforata è un classico gioco di puzzle, come quelli da tavolo, con tanto di azioni di rotazione e ribaltamento dei pezzi al fine di farli combaciare nel verso giusto. I pezzi del puzzle sono stati dunque sparsi casualmente nel mobilio della struttura, e occorrerà una completa e attenta ricerca, mobile per mobile, per recuperarli e riassembliarli.

**"Destroy him, my robots"**



Il gioco è un classico platform a schermata fissa, e ogni quadro è rappresentato da una stanza sempre diversa, sia come locazione nella struttura, sia come mobilio, sia come numero e tipo di nemici di guardia.

Le stanze sono collegate tra loro per mezzo di classici ascensori a corsa verticale e sono piantonate da solerti e attivi robot guardiani pronti in ogni momento a fulminare il nostro eroe con una bella scossa elettrica.

Alcune stanze sono composte da pareti conduttive, cosicché se un robot decide di erogare la sua mortale scarica elettrica verso una parete, il nostro agente speciale verrà fulminato all'istante se a contatto col pavimento. È ben noto infatti che gli agenti speciali si infiltrano nelle superstrutture nemiche in giacca e cravatta e con mocassini in cuoio tutt'altro che isolanti. Ma come diceva Shakespeare, il teatro funziona solo se c'è la complicità dello spettatore, e noi non solo siamo spettatori ma siamo soprattutto giocatori e la nostra complicità è totale.

Per fortuna durante le nostre attente ricerche è possibile recuperare delle password da immettere nei terminali sparsi nelle varie stanze per disabilitare temporaneamente i robot per una manciata di secondi, in modo da dare il tempo al nostro eroe di scansionare pazientemente le suppellettili alla ricerca dei pezzi di puzzle senza venir fritto nel frattempo.

Le password possono anche essere collezionate in speciali stanze in cui un computer, probabilmente a valvole e con uno schermo enorme a scacchiera, ci attende per un minigioco del tutto simile a Simon: il gioco consiste nel ripetere nel giusto ordine in scala cromatica crescente le note casuali che vengono proposte; ad ogni vittoria il minigioco si complica aggiungendo un'ulteriore nota fino ad esaurimento delle configurazioni disponibili.

Ogni volta che l'agente segreto viene folgorato, oppure cade in una botola senza fondo, non perderà una vita, ma gli verranno sottratti 10 minuti dal tempo massimo a disposizione per fermare lo stolto professore.

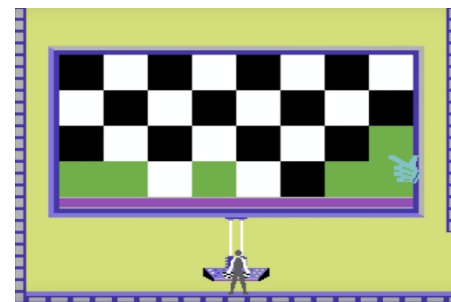
Per completare il gioco infatti non serve trovare solamente i pezzi di puzzle e metterli assieme, ma occorre anche recuperarli per tempo, prima che il timer scenda a zero e tutto esploda in un'ondata luminosa di pixel bianchi. E per tempo occorrerà recarsi nella stanza dove si trova un'apertura inaccessibile senza la preziosa parola chiave.

Ad ogni nuova partita le stanze, il mobilio, i robot, e i pezzi di puzzle, vengono rilocati casualmente dando longevità e piacevolezza al gioco. Io ancora a distanza di 35 anni lo trovo divertente e piacevole da giocare e finire.

**"No! No! Nooo! - Mission Accomplished. Congratulations!"**

Impossible Mission è uno di quei pochissimi giochi stimolanti, appaganti e che si lascia giocare ripetutamente senza invecchiare mai. È difficile riuscire a mettere su carta la bellezza e la freschezza di Impossible Mission, per cui se non lo avete mai giocato, perché magari siete atterrati solo adesso su questo pianeta, sappiate che avete una missione da compiere.

Una missione impossibile.



#### Gioco online

Questo capolavoro può essere giocato anche online all'indirizzo:

<http://impossible-mission.krissz.hu/>

# USCITA DAL RETRO

Una rubrica a cura di Adriano Avecone



## BIONIC GRANNY

La perfida vecchiarda cerca di blastare i poveri giovani con un bastone a forma di ammasso fecale infestato da termiti.



## DANGEROUS STREET

Una combattiva battona da porto cerca di blastare un golem vestito con una tovaglia simile a quelle in vendita nei mercatini cinesi.



## ROBOBOLT

Un robot per pavimenti a forma di ciambella gonfiabile per malati di emorroidi che emette il suono di un citofono. Bel gioco.

## BIONIC GRANNY

Anno: 1984. Sviluppatore: Mastertronic.  
Piattaforma: Commodore 64 e Amstrad CPC

Immaginate di acquistare un Commodore 64, portarlo a casa, accenderlo e caricare lo splendido gioco che l'onestissimo negoziante vi ha regalato: LA NONNA BIONICA, sarà divertente! Entusiasmo a mille, chissà che spasso. Dopo una scarna presentazione, su schermo viene visualizzato un totale orrore catodico, tanto brutto da far ammalare di depressione post-traumatica estrema anche il più allegro dei ragazzini. Lo scopo del gioco consiste nel controllare una malefica vecchiarda che intende mazzolare con un laido bastone pixeloso i giovinastri che impazzano fuori dalla scuola, evitando i leccalecca (!) da questi scagliati. Il gioco è uno dei parti più osceni prodotti da una mente umana: graficamente è realizzato in modo agghiacciante usando i caratteri standard del Commodore 64, gli "sprite" sembrano ammassi fecali di una colonia batterica di qualche fiume dalle parti di Calcutta, i movimenti ricordano gli spasmi dei malati terminali di colera e il gameplay sembra mutuato dalle torture inflitte nelle prigioni cambogiane. Il tutto condito da una musicchetta che ricorda le più agghiaccianti composizioni dei Gazosa, il gruppo musicale di adolescenti stonati in playback, riprodotte con un ukulele e al triplo della velocità. Unico aspetto positivo del gioco: il prezzo di 1,99 misere sterline, anche troppo per un orrore del genere.

## DANGEROUS STREETS

Anno: 1994. Sviluppatore: Micromania, Flair Software. Piattaforma: Amiga, CD32, MS-DOS, picchiaduro

Già dalla copertina, la cosa migliore del gioco, è possibile intuire lo squallore videoludico di questo titolo. Si tratta di un osceno picchiaduro in cui occorre mazzolare l'avversario eseguendo mosse completamente casuali. Il gioco è scattoso all'inverosimile, evidenzia una serie immensa di errori tecnici e prospettici, costringe a giocare a caso premendo i pulsanti senza alcun senso logico, utilizza sprite disegnati malissimo e animati con 1-2 frame anche nelle "mosse" più articolate, un effetto parallasse completamente cannato e un gusto atroce nel

design dei lottatori, che spaziano da un clone del baffuto attore hard Ron Jeremy alla classica bagasciona da porto vestita con indumenti spessi quanto il filo interdentale. Il gusto generale del gioco è comparabile alle illustrazioni presenti nei bordelli thailandesi, le musiche sono tanto pessime e stonate da far sembrare quasi melodiose le atroci musiche dei rapper drogati. Le mosse ricordano le convulsioni dei malati di delirium tremens o i riti propiziatori eseguiti dai selvaggi amazzonici in caso di stitichezza cronica. Il tutto è terribile, angosciante, troppo brutto per essere vero: Commodore ebbe anche il coraggio di proporre questo orrore in bundle con lo sventurato CD32. Da profanazione rettale immediata.

## ROBOBOLT

Anno: 1986. Publisher: Alpha Omega.  
Sviluppatore, grafico e musicista: Gary Leach.  
Piattaforma: Commodore 64

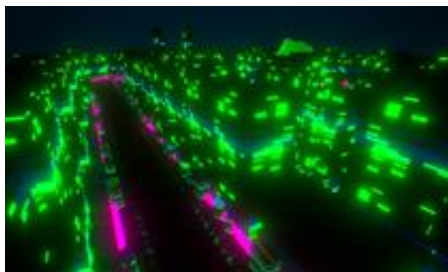
Robobolt è un gioco che anticipa uno dei prodotti più in voga nelle nostre case: il robottino per le pulizie ROOMBA. Un robottino simile a una ciambella gonfiabile per malati terminali di emorroidi scivola in un osceno livello grigio chiaro nel tentativo di eliminare altri robot disegnati in maniera altrettanto pessima. L'attacco disponibile consiste nel lancio di una palletta simile a un testicolo espunto dallo scroto di un gibbono. La schermata iniziale è realizzata talmente male da nascondere il titolo del gioco, come a voler mitigare la vergogna per l'immondo lavoro del programmatore Gary Leach, probabilmente segregato da un comando libico e torturato con brodo di mucillagini avariate. Le stonate musiche ricordano un jingle per detersivi nordcoreano e gli straziati effetti sonori sono simili al cicalino dei citofoni guasti, spingendo il giocatore alla totale pazzia dopo pochi minuti. La ciambella è priva di qualsiasi animazione, così come i ridicoli nemici. Tre quarti dello schermo sono occupati dal punteggio e da altri inutili dati e i livelli sono talmente confusi da rendere impossibile l'individuazione della direzione da intraprendere. L'AI dei nemici ha un'intelligenza paragonabile a quella di uno scarafaggio guatemalteco e il gameplay è di fatto inesistente. Da schiacciamento immediato.

# Intervista a Fabrizio Radica

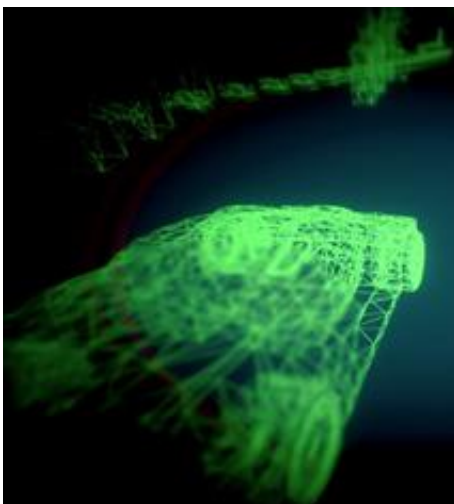
## creatore di Sprint2076

di Michele Ugolini

Quando si parla di talento, cari amici, tutto risulta lampante, soltanto se d'acchito si avverte la magica vibrazione di quel soffio vitale. Oggi respiriamo una frizzante ventata d'aria virtuale, che in realtà, proviene da una Italia talentuosa, che non ha assolutamente nulla da invidiare a nessuno.



Poi, se per un attimo, dovessimo complicare il calcolo algebrico, scrivendo sullo stesso foglio numerose variabili, tutte insieme, una in codice vettoriale, una EncomOS-V12, una in SolarOS, miscelandole in un mondo Arcade, con il comun denominatore del Retrogaming, sommando un lodevole lavoro di ambientazione audio, secondo voi, cosa potrebbe uscirne? Se il fautore di questo "mare della simulazione" ha amato il film cult Tron allora, sicuramente, saprà orchestrare egregiamente tutti i bit che gli passeranno sotto le mani e ci farà magicamente svanire il timore, derivato dall'enorme mole di dati.



Vi confesso che da tempo sto seguendo il diario di bordo sulla creazione di un videogame moderno, sebbene vestito di elementi prettamente Arcade e Retro. Vi presento il "Creativo" Fabrizio Radica, proprietario della RetroAcademy (e FutureAcademy). Un ragazzo dotato di

talento nella programmazione tanto da averne ritmato lo scandire della propria vita, un "leit motiv" di notevole ambizione, nonché dalle ampie prospettive. Pensate che sto esagerando? Lascero' dunque il giudizio a voi lettori, in questa intervista, preludio dell'uscita di una OMI (opera multimediale interattiva) intitolata Sprint 2076.

### PREMESSA

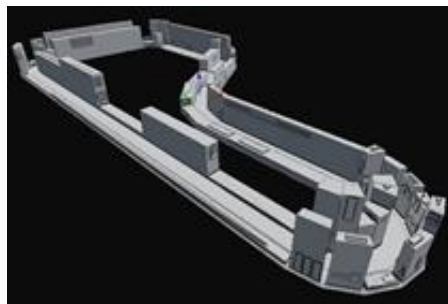
La Direzione di RM mi ha donato il grande privilegio di interagire nel porting con Fabrizio Radica e il suo attraente operato. Idem, Fabrizio ha ceduto all'ammirazione che provo per il suo lavoro e mi ha concesso l'intervista. Il mio sesto senso mi ha suggerito a gran voce di scappare finché ero in tempo, ma si sa, la passione è l'unico motore reale del mondo, quindi suppongo che l'interazione nata tra queste due ammirevoli realtà, sarà ricca di aggiornamenti, rubriche, aneddoti, gossip. Sicuramente saranno necessarie numerose puntate. Ebbene, RM, Fabrizio, RetroAcademy, grazie di cuore. Apriamo il Log di questo cammino e parliamo del bellissimo Sprint 2076!

### L'INTERVISTA

*RM: Ciao Fabrizio, un caloroso benvenuto da tutto lo staff, la prima domanda, la più ovvia: chi sei/siete e cosa c'è dietro questo capolavoro annunciato, che stai sviluppando?*

Ciao Michele e grazie per l'opportunità, mi sento veramente onorato. Professionalmente nasco nel 1995 come developer Web. Ai tempi programmavo in CGI+HTML 1.0 su IE e Netscape.. e non c'era Google che ti aiutasse.

Ho vinto vari premi nel web dal 1995 al 2011 circa, "finendo" anche su MTV con un progetto dedicato agli artisti musicali (un pre-social).



"Artisticamente" ho iniziato nel 1986 a programmare sul Commodore16, giochini e

retroacademy

programmini per poi cimentarmi sull'Amiga dal 1990 facendo musiche.

Nel 1994 circa, sviluppai assieme ad amici LA BIONDA di Franco Saudelli edito da Granata Press... sì, sviluppavamo giochi per Granata Press... prima che fallisse! Una sfiga guarda...

Nel Novembre del 2013, con Alessandro Bulgarelli, fondiamo **RetroAcademy**, nata per passione, per "gioco" e, personalmente, come una sfida.

RetroAcademy ha all'attivo numerosi eventi e vari collaboratori oltre che progetti rilevanti ed importanti già pianificati per il 2019. Dopo 5 anni di "avvio" ho deciso di creare il "lato oscuro" con FutureAcademy, nata prettamente per lo sviluppo di giochi arcade INDIE, sulla base di un modello da me elaborato (che non voglio raccontare qui) e vista come un'evoluzione non cronologica ma di stile, di RetroAcademy.



....così nel luglio del 2018, nasce Sprint 2076 ovvero il secondo gioco FutureAcademy (il primo è WiredPONG). Oggi FutureAcademy è costituito da me, Igor Imhoff ( l'artista modellatore e docente ) Marco Ricci ed Andrea Bovo, magnifici coder che, spero, facciano presto parte attivamente del team.



*RM: Realmente stai dando vita ad una "Opera Multimediale Interattiva" poliedrica, dotata di numerosi intriganti aspetti, da dove nascono le*

scintille di regia, interattività, animazione e quindi il tuo "OMI"?

Tutto nasce dalla passione per la DemoScene Commodore64 ed Amiga. Effetti e trucchi partono tutti da lì... dove l'arte digitale non ha tempo né luogo...

*RM: Nel tuo lavoro, l'anima del retrogaming permane come una solida piattaforma nella cerimoniosa presenza della parola, sia futuristica, sia eterna, sia tabù, cioè: Tron! Tiva di raccontaci questa complicata simbiosi?*

Folgorato da Tron fin dalla sua prima apparizione (sono del 1975)... innamorato della storia e dei colori.



Ogni ragazzino degli anni 80 desiderava vivere un'avventura digitale paradossale e distopica come quella di Tron. Ma vorrei citare altri miei miti; War Games



o D.A.R.Y.L.



*RM: Sei anche docente di sviluppo dei videogames presso la Scuola Internazionale di Comics di Padova. Quali sono stati gli elementi che più ti hanno arricchito durante le lezioni?*

I miei allievi.. ho imparato molto da loro; ho visto creazioni interessanti, idee geniali e spesso sono rimasto lì ad ascoltare la loro passione nell'imparare l'arte del Coding. Io non finirò mai di imparare... anche dai miei allievi.



*RM: Durante la docenza, quali elementi teorici, quali metodologie didattiche e su quali leve di interesse agisci per arricchire i tuoi giovani seguaci?*

Sulla passione... cerco di farli divertire.. di fare in modo che non abbiano ali tarpate nell'immaginazione, io dò gli strumenti... li porto alla soglia... Un pò come Morpheus in Matrix.

*RM: L'ambiente e il marketing reali, come si relazionano alla programmazione virtuale dell'ambiente "Indie"?*

Un macello.

*RM: Qualcosa mi dice che nel tuo lavoro hai una visione vettoriale in "High Dynamic Range", per caso bollono in pentola idee sovraesposte e/o sottoesposte?*

Ho un calderone di idee sì.. ma le scopriremo col tempo.

*RM: Te la sentiresti di tenerci aggiornati riguardo il tuo diario di bordo, per i prossimi numeri della rivista?*

Perchè no, mi farebbe piacere.

*RM: Fabrizio, un sentito ringraziamento da me e tutto lo staff di RM, oramai a partire dai numerosi lettori fino alla Direzione siamo entrati "in the Grid" !*

Che la sfida inizi allora... è tempo di mettere in moto la fantasia, come un tempo... alla fine, nessuno di noi deve dimenticare il bimbo che è stato.

Grazie a voi per avermi pensato e per avermi dato modo di raccontare la mia passione.

## APPROFONDIMENTI

Un "Creativo" chiaramente non è solo un

mero programmatore, idem Fabrizio. Ecco alcune tracce musicali che ha composto: <https://soundcloud.com/fabrizioradica>

Ecco anche alcuni video del canale Youtube [www.youtube.com/user/Fabbroz975](http://www.youtube.com/user/Fabbroz975) e FB [www.facebook.com/groups/retroacademy/](http://www.facebook.com/groups/retroacademy/) e [www.facebook.com/groups/futureacademyentertainment/](http://www.facebook.com/groups/futureacademyentertainment/) per vivere le varie tappe di sviluppo di Sprint 2076 e non solo.

Nelle prossime puntate avremo sicuramente più notizie dello sviluppo di questo videogioco.

Prossimamente ci addenteremo nel mondo di Fabrizio, quindi RetroAcademy [www.retroacademy.it](http://www.retroacademy.it) e FutureAcademy, interfacciando il suo mondo del codice [www.radicadesign.com](http://www.radicadesign.com) al nostro mondo editoriale per divulgarlo ad utilità del pubblico.

Sapremo quali eventi organizza ed in quali ambienti si muove per giustificare una tale agilità ed abilità nella programmazione. Questo porting nasce, non solo con l'intento di fornire ad un vasto pubblico la possibilità di ammirare la creazione di **Sprint 2076**, ma soprattutto per creare interazione con i suoi "degustatori" e permettere suggerimenti e/o richieste che potrebbero influire positivamente sullo sviluppo stesso.

Già alcune discussioni costruttive sono state elaborate internamente nel suo gruppo, direzionando la vettorialità e le gamme dinamiche di colore in una direzione piuttosto che un'altra.

Le discussioni comparative e non aversati ve al momento hanno dato vita ad una intrigante vettura che richiama K.i.t.t. (il telefilm Supercar degli anni 80). Chissà se porteranno addirittura alla creazione del quad pilotato da Quorra (Olivia Wilde in Tron) o dei Light Cycle?

## CONCLUSIONI

Il diario di bordo è ormai nato, concludiamo questo Log aprendo "the Grid" a tutti gli sfidanti.

Nelle prossime interviste sicuramente toccheremo la "suburbia" creativa, gli indispensabili tagli di regia che dolorosamente tutti i grandi registi devono apportare per donare scorrevolezza al film, quindi se i lettori vorranno, si potrà anche chiedere sommariamente quale "epurazione delle ISO" è stata affrontata nello sviluppo! Sarà un'occasione da non perdere per conoscere ulteriori dettagli di questo capolavoro italiano!

# RM sei tu... chi puo' darti di piu'?

di Francesco Fiorentini

Si', lo so, ho scocciato con queste frasi fatte e stornelli pubblicitari. Prometto che smetto. Ma anche in questo caso il ritornello di una conosciuta catena di supermercati si adattava perfettamente a quanto volevo scrivere nell'articolo di chiusura e non ho saputo resistere alla tentazione... ☺

## Nuove firme in RetroMagazine

Come avrete notato, in questo nono numero, ci sono almeno 3 nuove firme negli articoli della rivista. Nuovi collaboratori, saltuari o regolari che siano, ma che hanno deciso di farci pervenire il loro contributo per parlare di Retrocomputing e Retrogaming.

Tutto questo non puo' che farci piacere, per almeno due motivi. In primis perche' e' evidente che i nostri appelli non sono caduti nel vuoto e sempre piu' persone credono nel nostro approccio della divulgazione retroinformatica. Ma il vero motivo di orgoglio e' perche' possiamo cosi' offrire ai nostri lettori un prodotto sempre piu' eterogeneo e diversificato.

E' infatti evidente a tutti che il Commodore 64 abbia dominato la scena italiana negli anni 80/90 ed e' quindi piu' che naturale trovare persone che hanno questo tipo di background (anche tra i redattori), ma, come redazione, abbiamo deciso di accettare la sfida cercando gia' da questo numero di diversificare l'offerta ampliando il discorso a macchine che hanno avuto una minore penetrazione in Italia (parlo dell'**Amstrad CPC**) o addirittura alle fantasy console (come il **TIC 80**).

## ZX Spectrum

Discorso a parte lo merita lo ZX Spectrum. Da questo numero siamo felici di annoverare tra i

nostri collaboratori **Alessandro Grusso**, autore di **Spectrumpedia** e di alcuni giochi commerciali per lo ZX Spectrum. Le sue competenze saranno indispensabili per far trovare anche a questa leggendaria macchina lo spazio che merita nella nostra rivista.

## Copertina

Questo mese **Flavio** ci ha stupiti creando una copertina a tutta pagina. Stiamo crescendo e... da ora in poi la vorremo sempre! :-P

## Uscita dal retro

Non tutte le ciambelle riescono col buco (Ma non avevi detto che avresti smesso con le frasi fatte? Vi ho mentito... ☺), ne sa qualcosa **Adriano Avecone** che quando ci propose il nome di questa rubrica fece cadere qualcuno dalla sedia. Ovviamente il tutto senza cattiveria e cercando soltanto di strappare un sorriso tra un articolo tecnico e la recensione di un gioco. Se volete difendere un titolo, fatevi avanti e proponetene una recensione.

## Chi ha detto sorpresa?

Vi preannuncio che il prossimo numero sara' qualcosa di diverso, qualcosa a tema... Non vi anticipo niente per non rovinarvi la sorpresa, ma non rimarrete delusi!

Chiudo, come ormai di consueto, con i **ringraziamenti** a tutti i **gruppi Facebook** ed ai siti **OldGamesItalia** ed **IlVideoGioco.com** che ci aiutano a condividere la rivista ad ogni uscita e con un ringraziamento particolare a **Vincenzo Scarpa** che ha deciso di riservare nel suo ottimo sito **EmuWiki** uno spazio dedicato a RetroMagazine. Alla prossima!

## Disclaimer

RetroMagazine (fanzine aperiodica) e' un progetto interamente no profit e fuori da qualsiasi circuito commerciale. Tutto il materiale pubblicato e' 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 9

Direttore Responsabile  
**Francesco Fiorentini**

Immagine di copertina  
**Flavio Soldani**

Settembre 2018