

# RETROMAGAZINE

Future days are back



NUMERO 45 - NOVEMBRE 2023 - WWW.RETROMAGAZINE.NET



**LETTURE VIDEO LUDICHE**

Uno sguardo ai libri dedicati al mondo retro

**PUBBLICAZIONE GRATUITA**



## "RENOVATIO" (dal latino "RINNOVAMENTO")

Il retrogaming ed il retrocomputing in generale non sono mai stati così vivi. Qualcuno forse ne dubiterà anche se chi legge la nostra rivista probabilmente è un appassionato come noi, ma potrebbe, ciononostante, non essere d'accordo con me.

Potrei aggiungere anzi che il panorama si rinnova continuamente. Nuovi progetti hardware e software continuano a nascere e tenere accese le nostre passioni, e a volte le rinnovano.

Prendiamo il caso del famigerato Commodore 64. Qualche esempio di moderni progetti hw/sw? Cominciamo dai lettori di schede SD Commodore 64 (SD2IEC), che ci permettono di utilizzare le moderne schede SD come unità a disco per i nostri computer Commodore più veloci e, soprattutto, più capienti.

Difficile immaginare, nei tardi anni '80, che quarant'anni dopo avremmo avuto tra le mani supporti dalle dimensioni così ridotte, per poter immagazzinare programmi e dati contenuti in tutti i dischetti in nostro possesso allora. E, naturalmente, rimarrebbe ancora spazio a disposizione. Vogliamo parlare dei manuali d'uso o della Guida di Riferimento del Programmatore per C64, che vengono riscritti, reimpaginati e messi a disposizione in comodi pdf?

Non sto parlando dei manuali originali ma dei manuali corretti e revisionati! Per chi fosse curioso ecco il link di riferimento:

<https://pickledlight.blogspot.com/p/commodore-64-guides.html>

Oppure preferite che vi parli di nuovi giochi che vengono realizzati ancora oggi, da appassionati programmatori, grafici e musicisti, facilmente reperibili gratis o a costi basissimi? Anche qui un link di riferimento, ma il panorama è davvero vasto e sempre mutevole. <https://itch.io/games/newest/tag-commodore-64>

Vi piacciono i demo, le intro, i trainers e le nuove creazioni software in genere? Brani .sid, immagini grafiche realizzate per il vostro amato Commodore e tanto altro ancora? Ecco il sito, tra i tanti per la verità, dove è possibile attingere sempre a nuovi contenuti. E c'è anche qualcosa realizzato dal sottoscritto! <https://csdb.dk/>

Parliamo di nuovo hardware? Inizierei dal "TheC64" di Retrogames che ha suscitato tanti entusiasmi e, al contempo, tante critiche.

L'entusiasmo nasce dal fatto, ad esempio, che per la prima volta dalla scomparsa dal mercato della Commodore, si fabbricano e si distribuiscono nuovi dispositivi simili al nostro Commodore 64.

Le critiche nascono soprattutto dal fatto che questi dispositivi, al loro interno, emulano via software il Commodore 64 e, pertanto, l'esperienza utente può non essere perfetta. Ecco dove saperne di più: <https://retrogames.biz/products/thec64/>

Andando oltre, qualcuno ha ben pensato di realizzare un Commodore 64 sfruttando le attuali tecnologie FPGA, e su questa scia, da questa idea è nato "The Ultimate 64". Il link di riferimento è il seguente: <https://ultimate64.com/>

Ma perché simulare accuratamente soltanto il Commodore 64? E infatti, sfruttando questa tecnologia, ecco un altro prodotto, noto come "Mister". Link qui: <https://ultimatemister.com/>

I nostri lettori sanno bene che ci occupiamo spesso nella nostra rivista di queste "perle" che spuntano continuamente. Il senso di questo mio editoriale è proprio questo. Tutti questi spunti che vi ho fornito servono a dimostrarvi, se ce ne fosse ancora bisogno, di come il retrogaming ed il retrocomputing siano sempre vivi ed in evoluzione quindi...continuate a seguirci, ne avremo ancora tante da vedere, insieme a voi, amici lettori. Ciao e buona lettura a tutti!

**Marco Pistorio**

## SOMMARIO

♦ FPGA per principianti assoluti - parte 1	Pag. 3
♦ 7800 Game Drive e Mega 7800	Pag. 12
♦ Nintendo Entertainment System (NES)/ Famicom - parte 1	Pag. 14
♦ Il Fortran sul Commodore 64	Pag. 20
♦ Giocare a vite infinite - La sfida finale	Pag. 24
♦ Strani prodotti	Pag. 26
♦ Permutazioni	Pag. 28
♦ Libdragon Development	Pag. 30
♦ Quattro chiacchiere con i Knights of Bytes	Pag. 32
♦ Uno sguardo approfondito su Amiga38	Pag. 34
♦ Giappone 23^ puntata	Pag. 38
♦ GP Cars (C64)	Pag. 40
♦ RMW Consiglia: Letture Videoludiche	Pag. 42
♦ G-Zero (GB)	Pag. 48
♦ Attack of Petscii Robots (MSX)	Pag. 49
♦ Sam's Journey (NES)	Pag. 50
♦ Electronic Dreams the Space Odyssey (SMS)	Pag. 52
♦ Reshoot Proxima 3 (Amiga)	Pag. 54
♦ Rocky & Co (C64)	Pag. 56
♦ SNK VS Capcom (C64)	Pag. 58
♦ Legend of Wukong (MD)	Pag. 60
♦ Pampas & Selene (MSX2)	Pag. 62
♦ Star Wars: Empire Strikes Back (Plus4)	Pag. 64
♦ Super Monza GP2 (VIC20)	Pag. 65
♦ Cornelius (MSX)	Pag. 66
♦ Time Wizard (XL/XE)	Pag. 67
♦ BurgerTime (C64)	Pag. 68
♦ Dracula's Castle (C64)	Pag. 69

## Hanno collaborato alla stesura di questo numero di RetroMagazine World (in ordine sparso):

- |                              |   |
|------------------------------|---|
| • Dave Nardella              | • Takahiro Yoshioka   |
| • Carlo N. Del Mar Pirazzini | • Marta Rossmann  |
| • Daniele Brahimi            | • Mic the Biker Novarina                                    |
| • Francesco Fiorentini       | • Gianluca Girelli  |
| • Leonardo Miliani           | • Marco Pistorio  |
| • Roberto Del Mar Pirazzini  | • Immagine di copertina:<br><b>Giuseppe Mangini</b>         |
| • Giampaolo Moraschi         | • Layout di copertina:<br><b>Carlo N. Del Mar Pirazzini</b> |
| • Giuseppe Rinella           | • Immagine di chiusura:<br><b>Fabio de Renzis</b>           |
| • Filippo Santellocco        |   |
| • Michele Ugolini            |   |
| • Eugenio Rapella            |   |





# FPGA per principianti assoluti: partiamo da zero - prima parte

di Dave Nardella

## Introduzione

Diciamo subito che questo articolo non è un corso di progettazione FPGA. L'obiettivo è individuare un percorso didattico per avvicinarsi a questi sistemi.

Il problema degli FPGA non è la loro complessità intrinseca, al contrario, come constaterete: sono troppo semplici. Questo genera la necessità di acquisire una grande quantità di informazioni per capire come sfruttare questa flessibilità; ed è questo il nocciolo del problema. Queste informazioni spesso ritenute, a torto, "tutte essenziali subito" vanno a sovraccaricare la curva di apprendimento facendo desistere molte persone.

In questo contesto è quindi necessario uno Step 0, puramente introduttivo ma con dei piccoli approfondimenti esplicativi, per poter individuare una direzione, ma soprattutto stimare quanta distanza è necessario percorrere. Per cui, descriverò la loro architettura, la filosofia di programmazione, farò dei piccoli esempi pratici per visualizzare quanto descritto, vi consiglierò il materiale didattico da utilizzare e il primo hardware da acquistare. La lunghezza di questo articolo può sembrare in contraddizione con quanto ho appena detto, ma purtroppo l'argomento è articolato, fermarsi a delle semplici definizioni darebbe solo un'idea superficiale degli FPGA e alla fine ne sapreste quanto prima, vi assicuro zero.

Esiste purtroppo uno strato di roccia da superare in un unico passo dopo il quale realmente è possibile capire se questi sistemi possono avere un futuro nei nostri interessi e di che impegno abbiamo bisogno per affrontarli.

State tranquilli, il piccone lo uso io, voi dovete solo seguirmi.

## Sommario e prerequisiti

Per programmare gli FPGA è necessario avere un'idea precisa, almeno a livello di sistema, di cos'è una CPU ed avere dimestichezza con il concetto di sistema di sviluppo, inteso come software e boards (schede elettroniche che ospitano le CPU). Se termini come Raspberry, Arduino o sistemi embedded non vi dicono proprio nulla, allora è meglio colmare questo gap prima di continuare.

Le competenze di elettronica digitale aiutano tantissimo ma non sono indispensabili, un programmatore Python non è tenuto a conoscere i registri della CPU che eseguirà

il suo codice, come vedrete, esistono vari livelli di astrazione. Le basi di programmazione invece sono necessarie.

I macro-argomenti che affronteremo sono:

- Architettura di un FPGA
- Filosofie di programmazione e simulazione
- Campi di applicazione
- Brand e famiglie
- Approccio consigliato
- Consigli per gli acquisti
- Risorse consigliate

## Architettura di un FPGA

L'FPGA è un chip programmabile. Il suo nome è l'acronimo di Field Programmable Gate Array, cioè una matrice di porte logiche programmabile dall'utente.

Pensate ad un grande arazzo, esso è composto da un insieme di fili orizzontali, chiamato trama, ed uno verticali, chiamato ordito.



Il concetto che sta alla base è molto semplice: utilizzando nodi di diverso colore, all'intersezione della trama e dell'ordito, è possibile realizzare dei disegni strabilianti. L'arazzo è un sistema complesso realizzato mediante la composizione di piccoli elementi, tutti molto simili, mediante regole semplici.

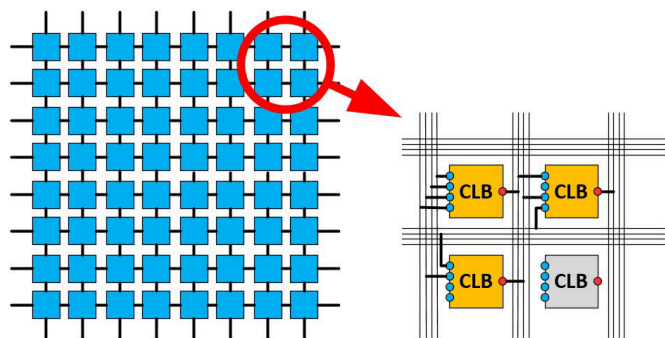
L'FPGA funziona allo stesso modo, non a caso la struttura base prende il nome tecnico di "fabric" che in inglese significa tessuto. I nodi sono rappresentati dai CLB (Configurable Logic Block, che vedremo fra poco), piccole unità logiche programmabili, tutte uguali, che possono essere interconnesse a dei fili orizzontali e verticali.





Programmare un FPGA si riduce a parametrizzare i singoli CLB e stabilire quali connessioni devono essere realizzate (fortunatamente non dovremo occuparci noi di questo livello di dettaglio).

Per questo motivo, spesso si parla anche di progettare un FPGA, in questo contesto i termini sono equivalenti.

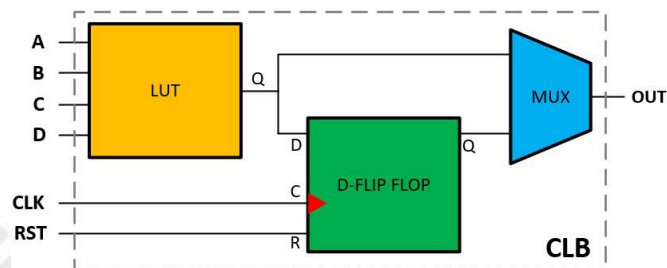


In figura vedete, sulla sinistra lo schema concettuale, e sulla destra, in dettaglio, le connessioni che è possibile realizzare fra i vari CLB. Ovviamente verranno utilizzati solo i CLB che servono.

### CLB

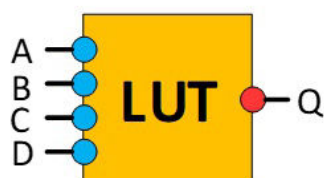
Per poter programmare un FPGA, come vedremo, non è indispensabile conoscere la loro composizione interna, ma è giusto parlarne per completezza di informazione. Se non avete nozioni di elettronica digitale potete saltare questo passaggio.

Nella forma più semplice i CLB sono costituiti da un LUT (Lookup Table) che può essere a 4 o 6 ingressi, un Flip Flop di tipo D ed un Multiplexer collegati come segue.



I LUT sono delle piccole unità logiche che emettono in uscita un segnale 1 o 0 a seconda della combinazione degli ingressi. Esisterà quindi una tabella delle associazioni simile a questa.

D	C	B	A	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
:	:	:	:	:
1	1	1	1	0

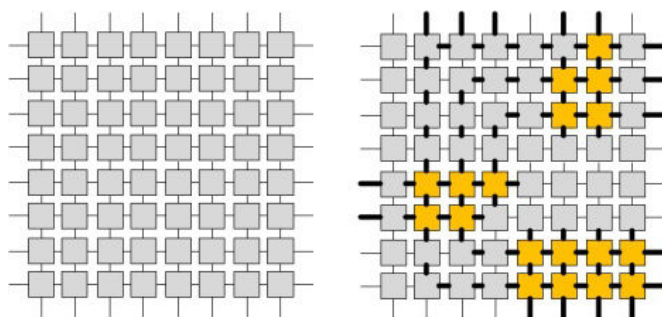


Il Flip Flop permette di memorizzare l'informazione al suo ingresso D sul fronte di salita dell'impulso di clock, possiamo considerarlo quindi una mini-memoria di 1 bit. Il Multiplexer è semplicemente un deviatore, emette in uscita il segnale presente su uno o sull'altro ingresso a seconda di come è parametrizzato. Nel nostro caso possiamo avere in uscita o il segnale diretto che proviene dal LUT oppure quello differito (memorizzato) che proviene dal Flip Flop.

Parametrizzare un CLB, quindi, significa semplicemente definire la tabella interna del LUT e stabilire cosa deve emettere il Multiplexer.

Attenzione, la parametrizzazione interna di un CLB non viene fatta manualmente, di questo si occupa il sistema di sviluppo. Anche quando programiamo una CPU in Assembly, noi non definiamo come spostare il bit di Carry o riempire la Cache.

Quando acquistiamo un FPGA quindi, questo sarà completamente vuoto.



BLANK

PROGRAMMED

A questo punto, ricordando cosa abbiamo detto prima, l'FPGA è un sistema complesso realizzato mediante l'interconnessione e la parametrizzazione di piccoli elementi, tutti uguali, mediante regole semplici.

### Input/Output

Tutti i chip elettronici presentano dei terminali, anche le FPGA; nel nostro caso questi sono rappresentati da molti (ma non tutti) fili che escono dai quattro lati del fabric.

In realtà non escono direttamente, esistono infatti dei blocchi specializzati (chiamati IOB) sul perimetro che consentono, una volta parametrizzati, di adattare il livello elettrico, di stabilire se i terminali sono di Input, Output o InOut e che tipo di configurazione elettrica devono presentare (push-pull, open drain ecc..).

Anche questa parametrizzazione verrà fatta nel tool di sviluppo ed è specifica per il chip che andremo ad utilizzare perché i terminali dovranno seguire la numerazione del produttore.







Chiunque ha programmato Arduino avrà utilizzato la funzione `pinMode()`, il concetto è identico.

### Blocchi speciali

Quella che abbiamo appena visto è un FPGA base, quasi teorica, con l'andar del tempo, questi chip si sono evoluti. Durante l'utilizzo degli FPGA, ci accorgiamo che ci sono dei circuiti di utilizzo molto ricorrente; il grande vantaggio delle FPGA è che, partendo dai blocchi base, virtualmente possiamo realizzare qualunque cosa; possiamo certamente creare delle librerie con dei circuiti base, però non è detto che tutto ciò che andiamo a realizzare sia conveniente (ed efficiente) farlo in questo modo. Sarebbe comodo avere questi blocchi già disponibili; questo è esattamente ciò che avviene negli FPGA moderni.

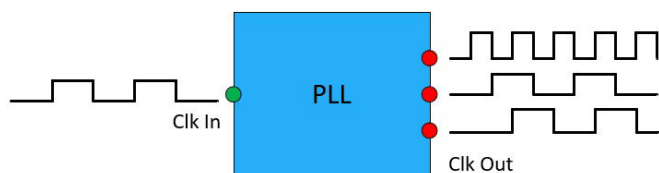
Nel tempo la presenza di questi moduli è cresciuta esponenzialmente; quello che potrebbe sembrare una violazione al principio base in realtà è una grande comodità, perché anche questi moduli sono "slegati" all'interno dell'FPGA e possiamo decidere se e come usarli, in alcuni casi, addirittura, i tools di sviluppo sono abbastanza intelligenti da capire quando è il caso di utilizzarli, in modo completamente trasparente.

I più utilizzati sono senz'altro le memorie, i blocchi DSP ed i PLL.

Le memorie sono dei banchi RAM di varia dimensione e larghezza di parola che i tools di sviluppo riescono a combinare facilmente fra loro.

I DSP sono dei blocchi aritmetici che permettono moltiplicazioni e somme di interi in complemento a 2 e sono dotati di un registro accumulatore, in genere a partire da 18 bit. Sono estremamente efficienti ed oggi non è raro trovarli anche in FPGA low cost.

Continuando, realizzare logica sequenziale prevede inevitabilmente l'utilizzo di un clock di sistema, in realtà, molto spesso, abbiamo necessità di più segnali di clock, a fasi e frequenze diverse. Anche qui, per evitare la loro realizzazione mediante CLB (che inoltre introdurrebbero ritardi poco desiderabili sulle linee di clock) esistono i moduli PLL che, facilmente parametrizzati, a partire da un segnale base, permettono di generare multipli e sottomultipli della frequenza in ingresso.



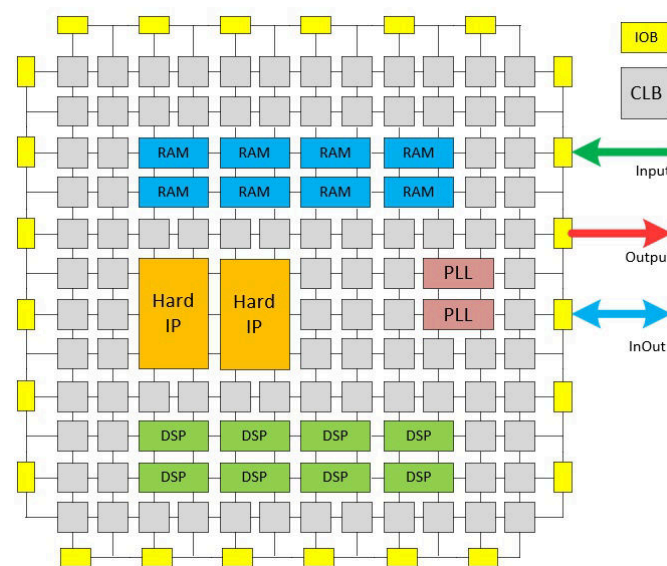
Esistono poi moduli più specifici, presenti nelle FPGA di

fascia medio-alta, ad esempio i serdes (Serializer-Deserializer) che permettono la conversione parallelo/seriale e viceversa a velocità di svariati Gb/s. Vengono utilizzati nelle apparecchiature di telecomunicazione, militari e ovunque sia necessaria un'alta velocità di trasferimento dati. Ma è comune trovare anche controllers per DRAM esterna, moduli Ethernet, convertitori A/D e D/A ecc. Tutto quanto può semplificarci la vita.

In genere, tranne memorie e DSP, tutti i blocchi extra realizzati in modo tradizionale, cioè disegnati direttamente nel silicio, prendono il nome di Hard-IP (Intellectual Property) perché vengono implementati in modo proprietario dai singoli costruttori, verranno istanziati e parametrizzati direttamente nei nostri programmi. Ci sono anche i Soft-IP, ma ne parleremo in seguito.

Avrete già capito che la taglia, e di conseguenza il costo, di un'FPGA dipende dal numero dei CLB e dalla presenza e numero dei blocchi specializzati.

Questa è una FPGA più realistica come struttura, ma non come numero di celle, le quali possono arrivare fino a centinaia di migliaia e anche più.



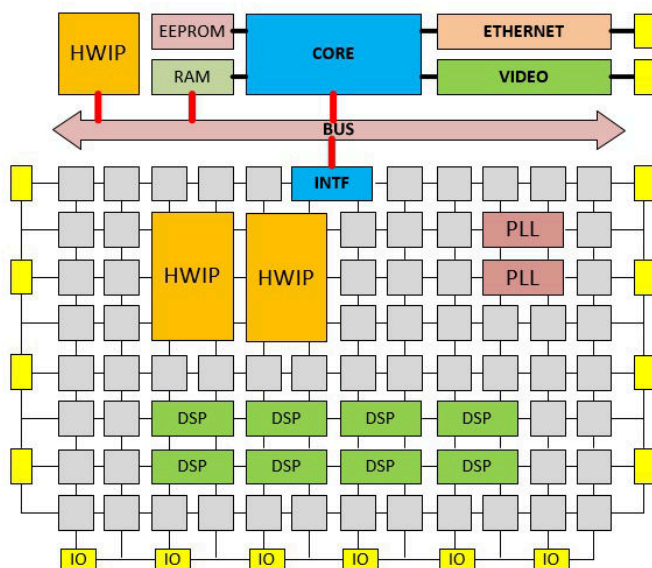
### SoC

Infine, come ultima evoluzione, troviamo i SoC (System on Chip) ovvero chip che contengono uno o più core, in genere ARM o RISC-V, che vengono chiamati HPS (Hard Processor System), ed un fabric FPGA, il tutto connesso mediante bus ad alta velocità.

In questo caso è difficile stabilire se questi chip sono da considerare delle CPU con un'area FPGA per realizzare logiche complesse, oppure delle super-FPGA che contengono uno o più core di servizio per interfacciarsi con l'esterno in modo più agevole.

Dipende tutto dal contesto ovviamente.





Per i nostri scopi ci concentreremo solo su FPGA ed i blocchi extra più comuni. L'utilizzo di SoC e di IP complessi è bene che appartengano ad un secondo step di apprendimento.

Riepiloghiamo quanto visto finora:

- L'FPGA è una matrice di blocchi configurabili che possono essere connessi fra loro.
- Esistono dei blocchi speciali, anch'essi da linkare, che integrano in modo efficiente funzioni standard e collaudate.
- Gli "agglomerati" creati con le connessioni dei singoli blocchi, essendo circuiti distinti, lavorano tutti in parallelo.
- Programmare o (il suo sinonimo) progettare un FPGA significa stabilire la parametrizzazione dei singoli blocchi e le loro interconnessioni

Vediamo ora come disegnare il nostro arazzo.

## Filosofie di programmazione

### Parallelismo

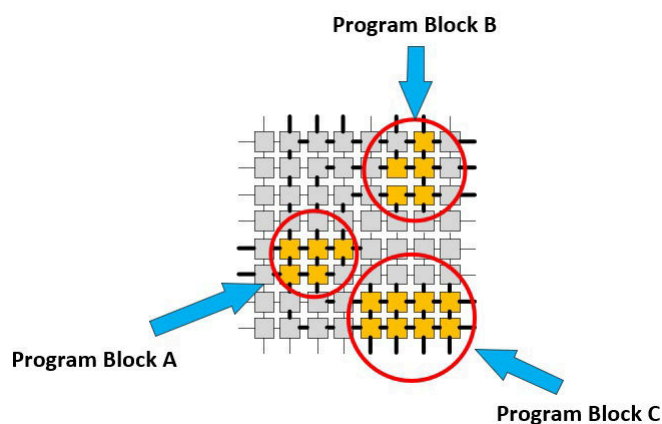
Questo è un concetto che deriva da quanto abbiamo visto e che voglio sottolineare subito perché sostanzialmente è uno dei motivi principali per cui spesso viene preferito un FPGA.

I gruppi di CLB, connessi fra loro, possono avere delle interconnessioni con altri gruppi o essere completamente indipendenti, per cui tutti questi agglomerati logici, deputati ad uno scopo specifico, lavoreranno completamente in parallelo. A differenza di una CPU/MCU dove i micro-task non sono mai completamente paralleli e devono essere eseguiti in time-sharing, ogni task che andremo a realizzare in un FPGA mediante un circuito specifico, lavorerà in modo completamente parallelo agli altri.

### Blocchi di programma

Ancor prima di entrare nei dettagli, ci siamo accorti di come la programmazione degli FPGA debba seguire delle metodologie "particolari".

Per quanto visto finora, potremmo immaginare qualcosa di simile.



Con piccole deviazioni, è esattamente quello che avviene. Questo è abbastanza confortante; come per la programmazione tradizionale, possiamo suddividere il nostro problema in tante parti più piccole, riducendone la complessità.

Un'altra considerazione che possiamo fare è che i nostri programmi non saranno una sequenza di istruzioni da far eseguire ad una CPU, ma ci devono consentire di stabilire parametri e connessioni; quindi, ci aspettiamo uno o più linguaggi orientato a questo scopo.

Anche questo è esatto, infatti i linguaggi per progettare un FPGA vengono chiamati HDL (Hardware Descriptor Language), utilizzeremo una serie di istruzioni che descrivono cosa vogliamo realizzare, esisterà quindi un compilatore che tradurrà tutto in come dev'essere realizzato. All'inizio dell'articolo ho detto che per programmare gli FPGA non è indispensabile avere nozioni approfondite di elettronica digitale, ma finora abbiamo parlato solo di blocchi logici e connessioni. Come si conciliano le due cose? La risposta breve è che, analogamente alla programmazione tradizionale, esistono dei livelli di astrazione che consentono di realizzare programmi in larga misura indipendenti dall'hardware sottostante.

È arrivato il momento di introdurre le quattro metodologie di programmazione, ma prima introduciamo l'ultimo concetto.

### Soft-IP

Analogamente a quanto visto per gli Hard-IP, i Soft-IP, o più brevemente IP, sono sostanzialmente dei moduli di libreria. Blocchi scritti in uno dei linguaggi che andremo







a vedere, che istancieremo nei nostri programmi come delle classi e li conatteremo fra loro.

## Metodologie di programmazione

A partire dal livello di astrazione più alto, troviamo:

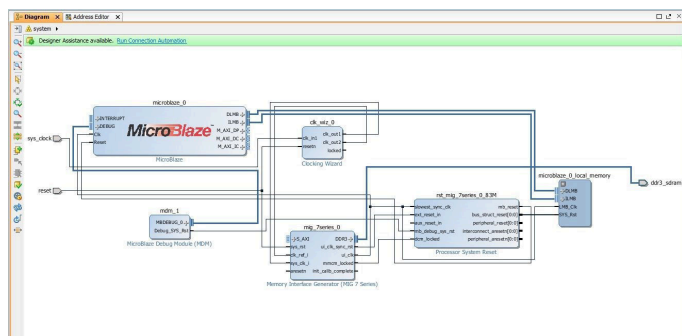
1. System
2. Behavioral
3. RTL (Register Transfer Level)
4. Physical

### System

La programmazione di sistema formalmente non è una metodologia a sé stante, come vedremo, concettualmente ricade nella Behavioral, però è comodo distinguere la soprattutto quando lavoriamo con i SoC, dove HPS e parte FPGA sono due entità distinte.

Sostanzialmente è quella di più alto livello in cui ci occupiamo solo di istanziare, parametrizzare e collegare i nostri IP.

I tools più evoluti permettono di farlo in forma grafica mediante la creazione di schemi a blocchi, e in questo modo, possiamo programmare un FPGA senza scrivere una riga di codice.



In figura è presente un sistema minimale contenente MicroBlaze di Xilinx; MicroBlaze è un softcore, ovvero il core di una CPU realizzato via software, che può eseguire programmi diretti (chiamati bare-metal) o, avendo risorse sufficienti, permette di ospitare Linux o un RT-OS (Realtime-OS). Esistono molti softcore free, soprattutto RISC-V, come vedremo.

Con questo sistema possiamo assegnare la memoria e le periferiche necessarie; avendo a disposizione un FPGA potente possiamo pensare anche di inserire più di un softcore e farli comunicare mediante un bus AXI (anch'esso presente in libreria) tipo quello utilizzato da ARM.

Utilizzando un tool grafico, il processo è molto semplice: si inseriscono tutti gli IP mediante un banale insert, si collegano fra loro, si parametrizzano mediante menù guidati e infine è possibile compilare il tutto.

Ove non esista un editor grafico, utilizzeremo un linguaggio

testuale andando ad istanziare i nostri IP che poi parametrizzeremo e collegheremo via codice.

Voglio far notare che anche programmando in modo grafico, il tool trasformerà i diagrammi in files testuali prima della compilazione.

### Behavioral

In questo livello si descrive il “comportamento” del nostro circuito, cioè quali sono le funzionalità che deve avere, senza entrare in dettaglio di flip-flop, memorie o blocchi logici. Sarà compito del compilatore riconoscere i blocchi base da utilizzare, parametrizzarli ed interconnetterli.

Un programma Behavioral sarà portabile a patto che gli IP “esterni” siano presenti anche nel sistema di destinazione ed abbiano la stessa interfaccia.

### RTL

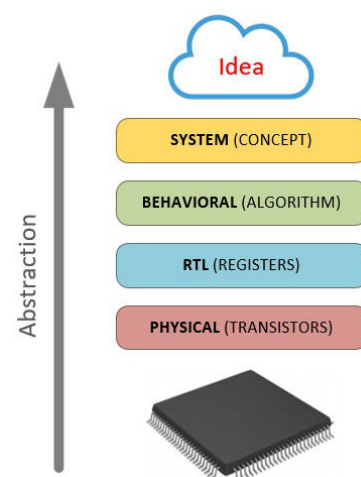
È l'acronimo di Register Transfer Level, e come recita il nome, la nostra programmazione sarà espressa in termini di registri, trasferimenti fra registri e di logica combinatoria. Qui siamo più vicini all'hardware, però sempre ad un livello più alto dei CLB. Quello che scriviamo sarà portabile, perché l'hardware che andremo ad utilizzare sarà generico e presente in tutti gli FPGA a prescindere dalla dimensione e dal Brand.

Per fare un'analogia, quando lavoriamo con i Files (apertura, chiusura, lettura/scrittura), siamo ragionevolmente sicuri che queste entità saranno presenti in tutti i sistemi operativi.

### Physical

Qui lavoriamo direttamente con l'hardware dell'FPGA che stiamo utilizzando, possiamo stabilire la locazione degli agglomerati, indicare dei path preferenziali per le linee di clock ed effettuare tutte le ottimizzazioni, specifiche per quel chip che il compilatore non è in grado di fare.

È necessaria una conoscenza molto approfondita di tutte le tematiche relative al timing nonché una preparazione elettronica notevole. Non pensateci per ora, ricordate che i tools di sviluppo riescono sempre meglio ad ottimizzare il codice FPGA, un intervento physical, se non fatto con





cognizione di causa, potrebbe risultare controproducente.

## Linguaggi di programmazione

Prima di fare qualche esempio è necessario introdurre i linguaggi di programmazione.

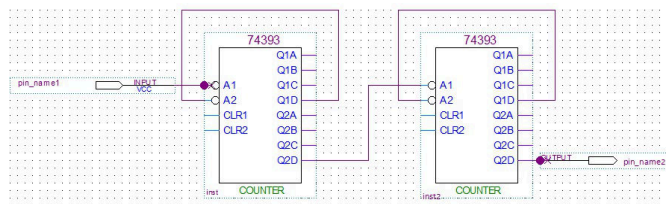
Esistono due categorie di linguaggi per programmare gli FPGA, una grafica ed una testuale.

## Programmazione grafica

La programmazione grafica non esiste in tutti i sistemi di sviluppo e non è standard, ovvero viene implementata in modo diverso dai vari produttori.

Abbiamo già visto la programmazione a livello sistema; l'altro livello in cui possiamo utilizzare un editor grafico è quello RTL.

Se vogliamo utilizzare un FPGA per realizzare delle reti logiche speciali, che realizzate in logica discreta ci costringerebbero ad utilizzare decine di circuiti integrati diversi, questa metodologia è abbastanza vantaggiosa. Andremo a realizzare il nostro schema elettrico inserendo porte logiche, flip-flop, contatori e quant'altro ci necessiti. Il compilatore trasformerà questo progetto in codice FPGA (vedremo un esempio in seguito).



Ovviamente al crescere della complessità questa metodologia presenta enormi problemi di gestione, la sua convenienza di utilizzo deriva direttamente dal background dell'utente; per un progettista elettronico, abituato a lavorare con porte logiche e circuiti integrati, rappresenta sicuramente la comfort-zone.

Un'ultima considerazione: tutto quello che realizziamo graficamente possiamo esprimerlo con un linguaggio testuale standard e portabile, infatti, tutti i tools di programmazione che implementano la programmazione grafica, ne permettono la traduzione in modo testo.

## Programmazione testuale

Per la programmazione testuale i due linguaggi principali, che ricordiamo essere HDL (Hardware Description Language), sono Verilog e VHDL.

Essi sono esattamente equivalenti, utilizzare uno o l'altro è solo una questione di gusti e propensione verso una o l'altra sintassi. Infatti, il Verilog è molto simile al C, mentre VHDL lo è al Pascal/ADA.

La comunità si divide esattamente al 50%, ognuna delle fazioni cerca di convincere l'altra che il suo linguaggio ha dei vantaggi in più: non credeteci, scegliete quello che è più vicino al vostro background di programmazione.

Verilog è più compatto, ha un controllo sui tipi molto basic, possiamo copiare qualunque variabile in qualunque altra, anche se hanno dimensioni diverse, esattamente come avviene in C.

VHDL è più "verboso", per esprimere un costrutto è necessario scrivere più parole, è fortemente tipizzato e quindi l'assegnazione fra tipi diversi può avvenire solo tramite un cast esplicito (che però è solo a livello del parser del compilatore e non produce codice extra).

```

VERILOG
1 module logic_v(input a, b, c, output q);
2   assign q = (a & b) | ~c;
3
4 endmodule

VHDL
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 entity logic_vhdl is
5   port (
6     a : in std_logic;
7     b : in std_logic;
8     c : in std_logic;
9     q : out std_logic
10  );
11 end logic_vhdl;
12
13 architecture RTL of logic_vhdl is
14 begin
15   q <= (a and b) or (not c);
16
17 end RTL;

```

Un grosso vantaggio è che tutti i tools permettono di gestire progetti contenenti moduli misti, Verilog e VHDL, per cui, conoscendo entrambi, possiamo pensare di importare/modificare qualunque IP a prescindere di come esso sia stato scritto.

Esistono anche altri linguaggi o preprocessori che aumentano il livello di astrazione, come SystemVerilog o SpinalHDL, o anche compilatori che permettono di utilizzare direttamente C/C++, ma vi consiglio di evitarli all'inizio, partite dalla base.

Con Verilog e VHDL possiamo programmare ad ogni livello, anche parzialmente a livello physical dato che esistono delle primitive a basso livello.

Prima di fare qualche esempio ricapitoliamo brevemente quello che abbiamo visto.

- Possiamo programmare gli FPGA a quattro livelli di astrazione.
- Possiamo utilizzare linguaggi grafici (proprietary) o due testuali altamente portabili che coprono i tre livelli più utilizzati.
- I linguaggi testuali, funzionalmente sono identici, possiamo scegliere quello che più ci aggrada o addirittura di utilizzarli entrambi nello stesso progetto

## Esempio di programmazione Behavioral/RTL

Questo è solo un articolo introduttivo/orientativo, vi consiglierò dei testi e tutorial adatti per imparare i due







linguaggi; tuttavia, vediamo un esempio per capire la differenza fra la programmazione Behavioral e quella RTL. Problema: vogliamo effettuare la somma di due parole di quattro bit e stabiliamo che questa debba essere modulare, cioè dobbiamo poter ricevere in ingresso un bit di Carry In (riporto aritmetico) proveniente dall'operazione precedente e quindi emettere un bit di Carry Out utile all'operazione successiva. Questo eventualmente ci permetterà di concatenare i blocchi che andiamo a creare per poter addizionare parole di lunghezza maggiore.

Per la cronaca, nella terminologia "digitale", un blocco che addiziona due parole gestendo Carry In e Carry Out viene chiamato full-adder.

Utilizzeremo sia Verilog che il linguaggio grafico di Quartus Prime (il sistema di sviluppo di Intel-Altera).

Partiamo con l'approccio Behavioral. Per sommare due parole scriveremo qualcosa del genere:

```

1 module fadder_4bit(
2     input [3:0] a, // 4 bit (from 3 to 0) operand A
3     input [3:0] b, // 4 bit (from 3 to 0) operand B
4     input cin, // Carry In
5     output [3:0] sum, // 4 bit (from 3 to 0) Sum
6     output cout, // Carry Out
7 );
8
9     assign {cout, sum} = a + b + cin;
10
11 endmodule

```

Questo è un modulo Verilog, concettualmente è molto simile ad una classe: se esso non è il modulo principale del progetto, è necessario che venga istanziato per poterlo utilizzare, e questo, come per le classi, possiamo farlo quante volte vogliamo.

Le parole di ingresso e di uscita sono degli array di quattro bit dichiarati nella forma [Bit finale : Bit iniziale], tutte le variabili, se non specificato, sono di tipo wire cioè dei semplici collegamenti che non memorizzano l'informazione dopo un'assegnazione, al contrario delle variabili di tipo reg, nelle quali il valore permane anche dopo l'assegnazione. Un'altra particolarità che incontriamo guardando questo codice è il raggruppamento; riferirsi al costrutto {Variabile A, Variabile B} equivale a creare in memoria una variabile che diventa la concatenazione geometrica di entrambe le variabili elencate; per definizione: parte alta - parte bassa, da sinistra verso destra.

Nel nostro caso, alla variabile sum, andiamo a concatenare un bit (il Carry out) che verrà riempito se la somma  $a + b + cin$  produce un riporto in uscita, cioè se è maggiore di quattro bit. In definitiva, la concatenazione {cout, sum} è una variabile larga 5 bit.

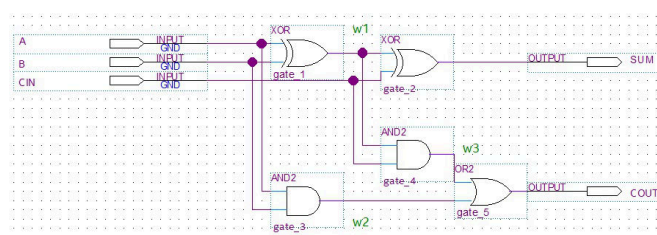
Osservate che se non avessimo la dichiarazione dei tipi, necessaria perché abbiamo creato un modulo, il tutto si ridurrebbe ad un'unica istruzione.

Veniamo alla programmazione RTL. Andiamo creare un

addizionatore a quattro bit come concatenazione di quattro full-adder di un bit, questo ci permette di diminuire la complessità totale utilizzando un approccio gerarchico, come risulterà chiaro soprattutto guardando i diagrammi grafici.

Vedremo come farlo sia in modo grafico, con uno schema elettrico, che in modo testuale.

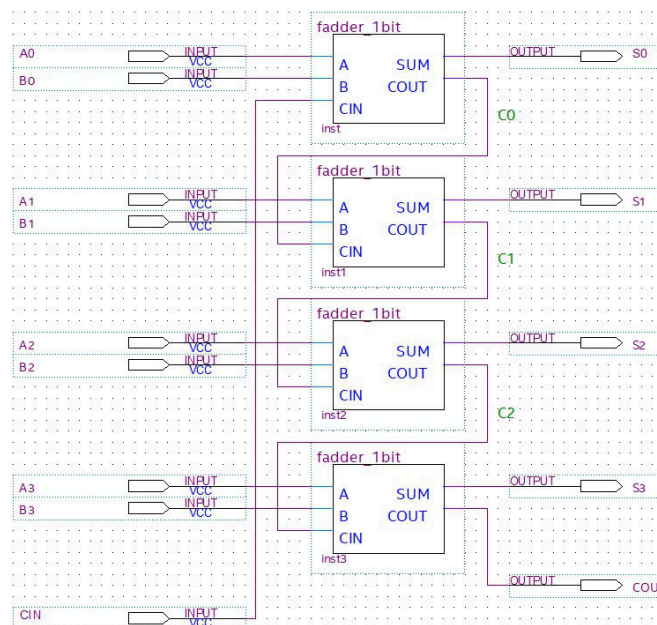
Questo è lo schematic-file di un full-adder ad un bit. (le scritte w1, w2, w3 sono solo label esplicative che ho inserito io il cui motivo sarà chiaro in seguito)



Questo circuito somma due bit, A e B, considerando il riporto in ingresso CIN, ed emettendo, se è il caso, il riporto in uscita COUT.

Se vogliamo estendere la parola dobbiamo usare più addizionatori collegando in cascata CIN e COUT.

A seguire lo schema del full-adder a quattro bit. Per realizzarlo ho creato un IP a partire dal full-adder ad un bit e l'ho inserito (cioè istanziato) quattro volte effettuando poi i collegamenti necessari.



Vediamo ora il corrispettivo testuale in Verilog. Ho utilizzato volutamente le primitive di libreria delle porte logiche per mostrare l'esatta analogia con lo schema elettrico, e adesso avranno senso le label w1, w2, w3 le quali sono delle variabili temporanee che rappresentano dei fili di collegamento.

Se non avessimo voluto utilizzare le porte logiche, molto più semplicemente, al posto di `and gate_3(w2, a, b);`





avremmo potuto scrivere `assign w2 = a & b`; il risultato è lo stesso.

Per definizione, il primo parametro di una porta logica è sempre l'uscita, a seguire ci sono gli ingressi.

```

1 module fadder_1bit_rtl(
2     input a,
3     input b,
4     input cin,
5     output sum,
6     output cout
7 );
8     wire w1, w2, w3; // temporary
9
10    // gate_x is an arbitrary (unique) instance name
11    xor gate_1 (w1, a, b);
12    xor gate_2 (sum, w1, cin);
13
14    and gate_3 (w2, a, b);
15    and gate_4 (w3, w1, cin);
16    or gate_5 (cout, w2, w3);
17
18 endmodule

```

Questo è infine il full-adder completo dove si vanno ad istanziare i quattro full-adder ad un bit.

Come potete vedere, i due approcci sono abbastanza diversi.

```

1 module fadder_4bit_rtl(
2     input[3:0] a, // 4 bit (from 3 to 0) operand A
3     input[3:0] b, // 4 bit (from 3 to 0) operand B
4     input cin, // Carry In
5     output[3:0] sum, // 4 bit (from 3 to 0) Sum
6     output cout // Carry out
7 );
8
9     wire c0, c1, c2;
10
11    fadder_1bit_rtl adder_1 (a[0], b[0], cin, sum[0], c0);
12    fadder_1bit_rtl adder_2 (a[1], b[1], c0, sum[1], c1);
13    fadder_1bit_rtl adder_3 (a[2], b[2], c1, sum[2], c2);
14    fadder_1bit_rtl adder_4 (a[3], b[3], c2, sum[3], cout);
15
16 endmodule

```

### Ricapitoliamo

- Nel primo esempio (Behavioral) abbiamo detto al compilatore cosa volevamo ottenere, lasciando a lui decidere come farlo.
- Nel secondo (RTL) gli abbiamo detto anche come ottenerlo.
- Abbiamo visto che la programmazione grafica equivale a disegnare uno schema elettrico e che questa può essere espressa in modo testuale

Il grosso vantaggio è che queste metodologie possono coesistere nello stesso progetto.

### Simulazione

Debuggare il programma di un FPGA non è un'attività semplicissima, non abbiamo la possibilità di inserire breakpoint perché non esiste una vera e propria esecuzione sequenziale, e non abbiamo nemmeno la possibilità di effettuare il watch delle variabili.

Inoltre, per progetti complessi, quando magari sono presenti softcore, i tempi di compilazione non sono rapidissimi.

Per questi motivi è possibile effettuare delle simulazioni software.

Il nostro modulo principale, chiamato in gergo top, come

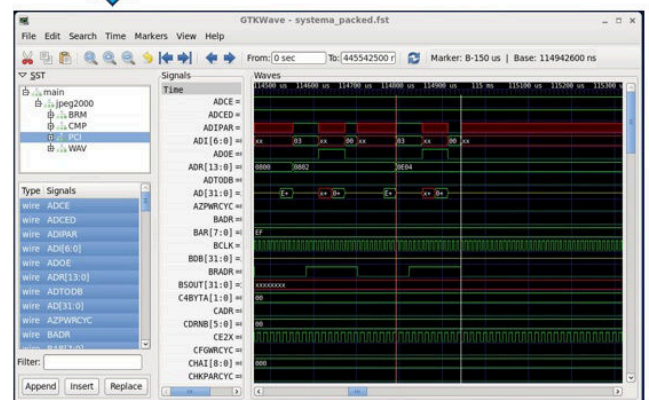
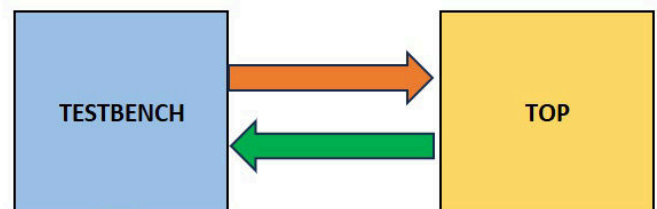
abbiamo visto, avrà, nella dichiarazione un certo numero di segnali di Input/Output che, come già accennato, mediante un file apposito di configurazione, andremo ad associare a dei pin fisici dell'FPGA indicando anche altri attributi (input, Output, Open drain ecc..).

Il concetto di simulazione negli FPGA è molto semplice: creiamo un modulo, in gergo testbench, che istanzia il modulo top e gestisce via software i suoi I/O mediante "stimoli" programmati.

C'è quindi la possibilità di visualizzare i segnali scambiati in un analizzatore logico software oppure salvarli su disco e visualizzarli con un altro tool.

Alcuni sistemi di sviluppo hanno il simulatore integrato, altri no. Esistono tuttavia dei simulatori free, tipo Icarus, che permettono la compilazione e simulazione di un progetto Verilog producendo un file visualizzabile da GTKWave, un tool anch'esso free.

### Campi di applicazione



Grazie alla loro versatilità e alla potenza cresciuta negli ultimi anni, gli FPGA hanno visto aumentare il loro impiego in architetture storicamente ad appannaggio delle CPU tradizionali.

Il primo impiego massiccio è stato nel campo delle comunicazioni; con la possibilità di realizzare protocolli personalizzati e alta velocità di trasferimento dati, oggi troviamo FPGA a partire da apparecchiature "economiche" quali modem e TV, fino ad arrivare a codificatori/decodificatori militari, ponti radio e schede di interfaccia fra server.

A seguire, il campo di applicazione più presidiato è







l'elaborazione numerica dei segnali (DSP). Parliamo quindi di Filtri, FFT, convoluzioni, analisi delle immagini, analisi della rumorosità, ecc.

Un'altra applicazione che vede l'impiego degli FPGA in rapida crescita sono gli acceleratori hardware per computers ad alte prestazioni e la crittoanalisi. Si stima che nei prossimi anni l'hardware basato su FPGA pareggerà come presenza quello delle CPU tradizionali.

Grazie alla natura "parallela" e ad opportuni blocchi speciali, un altro campo in cui gli FPGA stanno emergendo prepotentemente sono le reti neurali e il machine learning. Quindi sistemi per la visione artificiale ed il riconoscimento di patterns.

Infine, gli FPGA vengono utilizzati come sistemi di sviluppo per gli ASIC, chip architetture molto simili che però vengono disegnati direttamente in fabbrica su silicio. Il costo di un chip ASIC è molto più basso di un FPGA, sono pensati per applicazioni ad alto volume, tipo controllers per Hard Disk, schede grafiche ecc. Il problema è che per realizzare il primo chip è necessario un investimento iniziale di qualche centinaio di migliaia di euro, cifra che viene ripetuta in caso di revisioni per errori. Poter simulare un circuito sul campo, e non solo analiticamente, può rappresentare una notevole riduzione del rischio.

### Realizzazioni accessibili a tutti

Abbiamo visto la struttura di una FPGA e il suo parallelismo, inoltre abbiamo appreso che esistono dei blocchi specializzati per aumentare la potenza di fuoco. Abbiamo intravisto le enormi potenzialità che derivano da un'architettura componibile a piacere, ora ci domandiamo: cosa possiamo realizzare?

Shield di interfaccia verso schede commerciali

Immaginate di aver bisogno di dieci interfacce SPI che lavorano in parallelo e pre-elaborano i dati, non ci sono MCU che le contengono. Realizzarle con un FPGA è abbastanza semplice, fatta una interfaccia la si moltiplica. Stesso discorso ovviamente per UART, I2C e CAN.

### Gestori di protocollo

Un altro impiego potrebbe essere un concentratore di segnali provenienti da vari sensori che comunicano con protocolli diversi, tipo 1Wire, CAN, RS485, ecc. L'FPGA potrebbe poi comunicare con un MCU o un Gateway IoT utilizzando un'interfaccia SPI veloce.

Le applicazioni sono tantissime, a molte sicuramente finora non abbiamo mai pensato perché impossibili da realizzare con metodi tradizionali, ma quelle sicuramente

più intriganti, sono i softcore.

### Softcore

Abbiamo già accennato ai softcore blasonati MicroBlaze e Nios, sviluppati e supportati direttamente dai produttori di FPGA, è possibile trovare in rete molti altri core, altrettanto complessi, basati su architettura RISC-V perché è free, ampiamente documentata e componibile, quindi possiamo sia semplificarla che arricchirla.

Un softcore è meno veloce di un hard-core di identica architettura e schema interno, però, avendo controllo completo sull'esecuzione, possiamo ad esempio realizzare dei softcore realtime.

Vi ricordo che realtime non significa veloce o performante, ma deterministico. Ho scritto un articolo che vi invito a leggere se volete approfondire l'argomento.

Possiamo riprodurre CPU esistenti o crearne di nuove con architetture e set d'istruzioni arbitrari, magari orientate alla soluzione di problemi specifici.

C'è tutto il mondo retrogame per il quale è possibile trovare repliche esatte di intere console, cioè CPU, memoria e chip di supporto. I più famosi, MiSTer e MiST, utilizzano hardware non economicissimo, ma c'è anche NESTang che riproduce un Nintendo NES su una board FPGA molto economica (Tang Nano 20k).

Questi progetti sono molto interessanti dal punto didattico, è possibile vedere come sia semplice replicare uno Z80, 6502 o 68000 in un FPGA.

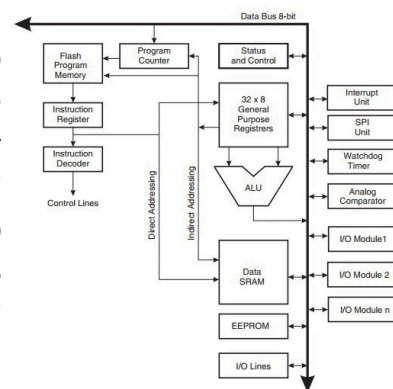
Nella figura è mostrata l'architettura interna di ATmega328p, l'MCU presente in Arduino che è possibile trovare come softcore in rete.

Infine, possiamo pensare di progettare dei micro-core per eseguire task specifici ed allocarne un numero elevato (avendo CLB disponibili) per farli lavorare tutti in parallelo.

Abbiamo la libertà di realizzare da core didattici a otto bit "Fetch-Decode-Execute", fino a RISC-V o POWERPC con prefetch, cache e memoria TMC. È solo una questione di CLB disponibili e, ovviamente, di skill.

*Articolo pubblicato con il permesso dell'autore.*

*Appuntamento al prossimo numero per la seconda parte di questo interessantissimo articolo!*





# 7800 Game Drive e Mega 7800

di Filippo Santellocco – [www.santellocco.com](http://www.santellocco.com)

È dal 2016 che James, il tecnico inglese che realizza i prodotti RetroHQ ([www.retrohq.co.uk](http://www.retrohq.co.uk)), pensa ad una multicart per la console Atari 7800. All'epoca la situazione non era delle migliori, non essendo facilmente reperibile la vecchia e costosa Cuttle Cart 2 uscita nel 2003.

Lo sviluppo è iniziato nel 2019 ma, complici COVID e mancanza di componenti, solo nel 2021 James ha concretizzato il suo progetto e ne ha dato notizia. Due anni fa il panorama era migliore: erano disponibili le multicart Concerto e Dragonfly, però, sia per alcune carenze di queste ultime, sia per le funzionalità avanzate via via mostrate da James, sia infine per la reputazione guadagnata con ottimi prodotti quali Lynx Game Drive e Jaguar Game Drive, si è creata una grande attesa tra gli appassionati.

Naturale quindi che i primi 200 esemplari commercializzati a luglio siano andati esauriti nel giro di poche ore. Niente paura, al momento della pubblicazione di questo articolo probabilmente il prodotto sarà di nuovo disponibile.



## Selezione giochi 2600

Oltre al chip Pokey singolo, la cartuccia emula anche doppio Pokey, Yamaha YM2151, BupChip e scheda COVOX. A proposito di software homebrew, apro una parentesi per sottolineare come negli ultimi anni siano stati sviluppati giochi che mettono in luce le potenzialità di una console che se commercializzata tempestivamente nel 1984 invece che nel 1986 avrebbe sicuramente avuto maggior successo. Penso ad esempio a titoli come 1942, A.R.T.I., Attack of the Petscii Robots, Ghosts 'n Goblins, Keystone Koppers, Millie and Molly, PentaGo e Popeye.

Come noto la console 7800 è retrocompatibile con l'Atari 2600. La Game Drive permette il caricamento di tutti i suoi giochi, ad eccezione di quelli che si avvalgono del processore ARM contenuto nella multicart Harmony e nelle schede Melody. Qualcuno potrebbe storcere il naso perché tra gli homebrew si tratta di quelli tecnicamente più avanzati ma bisogna dire che a distanza di 14 anni dalla sua disponibilità, sono pochi gli amanti del VCS che non posseggono una Harmony; l'unico rammarico è quello di non poter giocare a questi giochi su 7800 sfruttando



Ma quali sono le funzionalità che hanno suscitato così grandi aspettative?

Trattandosi di una multicart, il suo scopo principale è quello di caricare i programmi (giochi e demo), nel caso specifico collocati in una scheda microSD. Ebbene, la Game Drive offre una compatibilità totale non solo con i giochi classici (compresi quelli con l'audio del chip sonoro Pokey senza bisogno di reperirne e installarne uno), ma anche con tutte le produzioni homebrew moderne, che a volte risultano ostici per altre multicart (vedi Rikki & Vikki).





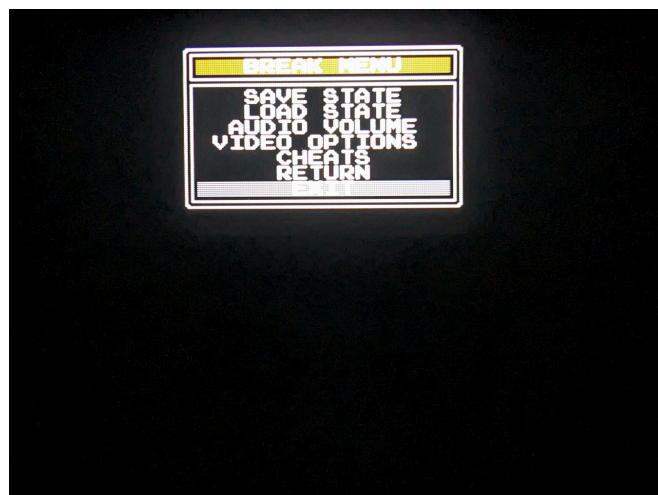


**Selezione giochi 7800**

l'uscita RGB. Uscita RGB? Esatto, tramite un cavetto in standard Sega Mega Drive II collegato alla cartuccia si può godere, in contemporanea con la normale uscita RF, della qualità assicurata dal segnale RGB. Naturalmente l'ideale è usare uno schermo CRT ma anche gli upscaler traggono giovamento dall'RGB della Game Drive.

Tornando a funzionalità più usuali per una multiscart, il menù a schermo visualizza le informazioni sui giochi e permette il loro caricamento. Molto comoda la possibilità di interromperli e di caricarne altri senza dover spegnere né premerne i tasti posti sulla console (al momento solo giochi 7800). Per farlo basta premere i due pulsanti del pad e spostare il joystick in basso; il gioco viene messo in pausa e appare il cosiddetto Break Menu.

Tale menù permette di accedere a varie funzionalità, quali il salvataggio degli stati (nove per ogni gioco) e le cheat. In questo modo anche i giocatori meno abili o con meno tempo possono godersi i giochi fino in fondo sull'hardware reale. Dal menù si può anche regolare il volume dei singoli chip audio (alcuni giochi usano più chip



**Break Menu**

contemporaneamente), cambiare palette (per simulare il cambiamento che interviene quando la console si scalda) e abilitare effetti video. Inoltre la Game Drive consente il salvataggio di highscore e dati nei giochi predisposti. Utile infine la possibilità di collegarvi il PC per testare velocemente i propri programmi senza doverli trasferire sulla microSD; è già predisposto il 7800Basic, il linguaggio moderno che a dispetto del suo alto livello sta contribuendo allo sviluppo di giochi straordinari. RetroHQ ha pure commercializzato un adattatore denominato Mega 7800, grazie al quale si possono adoperare pad e stick Master System/Mega Drive e light phaser. È raccomandato il pad del Mega Drive con 6 pulsanti, che rende velocissimo il caricamento e il salvataggio degli stati.

In conclusione, dopo averla provata in anteprima per due mesi e fornito feedback al produttore, non posso che consigliare la 7800 Game Drive. Il prezzo la pone nella fascia alta del mercato ma credo che sia corretto se si considerano la compatibilità, le funzionalità e la presenza di un'uscita RGB che altrimenti necessiterebbe di modifiche hardware non proprio alla portata di tutti.



**Vista frontale e posteriore della multiscart**







# Nintendo Entertainment System (NES)/ Famicom - parte 1

di Leonardo Miliani

Il 2023 non è stato solo il 40° compleanno di due importanti computer quali il Commodore 64 e lo ZX Spectrum. C'è un altro sistema che festeggia gli stessi anni, un sistema i cui numeri annichiliscono le macchine su citate: stiamo parlando del "Nintendo", così come era chiamato dai ragazzini dell'epoca, la console che ha segnato la rinascita del mercato dei videogiochi domestici, facendo riprendere il settore da quella profonda crisi che aveva portato grossi nomi dell'intrattenimento elettronico al fallimento o quasi, con una spinta così imponente da rappresentare un volano che ha poi trainato tutti i produttori di giochi verso un nuovo boom creativo ed economico. Il Nintendo Entertainment System (fig. 1), noto anche come NES, nato come Famicom in Giappone e poi venduto in tutto il mondo, è la console che andremo ad analizzare in questo articolo.

## Dalle carte da gioco

Nel lontanissimo 1889, a Kyoto, Fusajiro Yamauchi fonda la Nintendo Karuta (a volte indicata anche come Koppai, ma è solo una diversa lettura degli stessi ideogrammi), una società dedita alla produzione di carte da gioco tradizionali giapponesi dette "hanafuda". Karuta è la traslitterizzazione del portoghese "carta", carta da gioco, perché i Portoghesi introdussero nel paese le carte da gioco nel XVI secolo. Durante i decenni successivi Nintendo si costruisce una buona reputazione e gli affari vanno discretamente bene, operando in un settore quasi di nicchia. Le cose cambiano durante la Seconda Guerra Mondiale: le restrizioni dovute al conflitto e le nuove necessità delle persone fanno sì che l'azienda attraversi un periodo di crisi. Nel 1949/1950, a causa delle condizioni di salute dell'allora presidente in carica Sekiryo Yamauchi



**Fig. 2 - Hiroshi Yamauchi, il terzo presidente di Nintendo sotto la cui guida è nato il NES (fonte: NYTimes.com)**

(figliastro di Yamauchi, nato come Kaneda ma adottato nel 1907), chiama il nipote Hiroshi Yamauchi a condurre l'azienda di famiglia (fig. 2). Quest'ultimo inizia un piano di ristrutturazione e diversificazione degli investimenti per cercare di aumentare il fatturato di Nintendo.

È nell'ottica di questi cambiamenti che Nintendo stringe accordi con Walt Disney per offrire carte da gioco con i personaggi Disney. La risposta del mercato è immediata e positiva: le nuove carte vanno letteralmente a ruba. Ma, come spesso accade in questi frangenti, le cose non durano molto perché Nintendo si ritrova a dipendere da un prodotto destinato ai soli bambini dato che le vendite delle carte da gioco per adulti continuano a diminuire: questo perché la società giapponese in quel periodo sta cambiando velocemente e gli adulti stanno iniziando a divertirsi con altre forme di intrattenimento, ad esempio le sale "pachinko" (un tipo di gioco d'azzardo meccanico nato da poco), oppure passano il tempo nelle sale bowling. Il presidente cerca di diversificare ulteriormente gli affari di Nintendo, arrivando ad investire anche nel mercato del riso istantaneo e delle compagnie di taxi, ma senza



**Fig. 1: Il Nintendo Entertainment System, meglio noto come NES o, semplicemente, Nintendo (autore: Evan-Amos – fonte: Wikimedia Commons)**







**Fig. 3 - Gunpei Yokoi, l'ingegnere ideatore di numerosi successi targati Nintendo (autore: Gobonobo - fonte: Wikimedia Commons)**

successo. Decide perciò di creare una divisione di Ricerca e Sviluppo (o R&D, dall'inglese Research & Development) ed assume un giovane e brillante ingegnere, Gunpei Yokoi, con il quale ideare nuovi sistemi di intrattenimento (fig. 3).

### I primi giochi elettronici

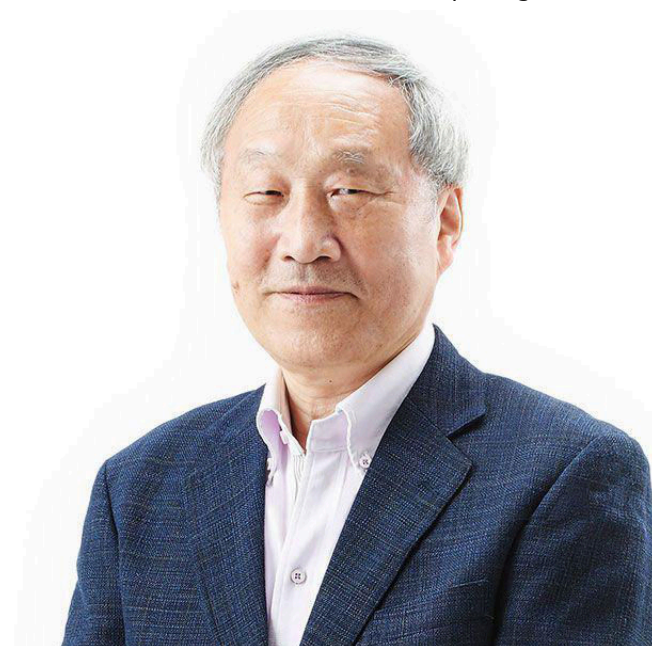
Come primo giocattolo Yokoi crea la "Ultra Hand", una specie di mano estensibile operante con un'impugnatura a mo' di forbici e con una parte centrale estensibile costituita da segmenti incrociati. Stringendo l'impugnatura, la parte centrale si allunga e contemporaneamente la mano all'altra estremità si stringe permettendo di afferrare piccoli oggetti (delle palline, nella confezione del gioco). Il successo è clamoroso: presentata nel 1971, ne vengono vendute circa 1 milione di unità, risollevando Nintendo dalla crisi finanziaria in cui latitava. C'è anche un altro ingegnere che lavora per Nintendo, Masayuki Uemura (fig. 4): nello stesso periodo ha sviluppato una serie di giochi elettronici basati su armi finte dotate di sensori ottici per rilevare se il bersaglio è stato colpito o meno,

le "Beam Gun". Anche di questi giochi ne vengono venduti più di 1 milione di pezzi.

Nel 1974 Nintendo rilascia uno dei primi esemplari di giochi arcade, "Wild GunMan", un sistema elettro-meccanico basato su un proiettore che proietta le immagini di un pistolero del Vecchio West su uno schermo. Quando i suoi occhi brillano è segno che il pistolero sta per sparare: se il giocatore è più veloce di lui e lo colpisce con la sua pistola finta, vince il duello (il gioco viene rilasciato anche nel Nord America nel 1976 tramite Sega). Vengono prodotte anche diverse serie di giochi per ragazzi basate su GunMan e Beam Gun (fig. 5). Yamauchi capisce che il futuro sono i videogiochi elettronici e firma un accordo con Magnavox per la distribuzione della console Odyssey in Giappone.

Intorno al 1977 la divisione R&D viene divisa in due: nascono la R&D1, diretta da Yokoi, e la R&D2, diretta da Uemura. La R&D1 inizia a lavorare ad un gioco portatile a batterie basato su un display a cristalli liquidi, che viene presentato nel 1980 come "Game&Watch". Il successo è mondiale: ne vengono presentate diverse varianti, ognuna con un gioco differente, e, nel suo primo anno di vita, ne vengono venduti 14 milioni. Alcune versioni, basate su famosi arcade, vanno ancora meglio: quello basato su "Donkey Kong" vende da solo 8 milioni di unità. Nel Game&Watch viene introdotto anche un elemento che ritroveremo più avanti: il controllo a croce (fig. 6). La R&D2 produce in collaborazione con Mitsubishi Electric delle console domestiche simili al Pong, la prima delle quali è la Color TV-Game 6, a cui segue a breve la Color TV-Game 15 (fig. 7): entrambe permettono di giocare a diverse varianti di Pong.

La R&D1 rilascia nel 1978 e nel 1979 i primi giochi arcade



**Fig. 4 - Masayuki Uemura, l'ideatore del NES/Famicom (fonte: Memoriabit.com.br)**





**Fig. 5 - Una versione di GunMan con una Beam Gun inclusa nella confezione (fote: Beforemario.com)**

elettronici, rispettivamente “Sheriff” e “Radar Scope”: quest’ultimo, nonostante riesca a rivaleggiare bene contro “Galaxian” sul suolo nipponico, non si afferma al di fuori del Giappone, risultando nel complesso un fallimento commerciale. Per riprendersi dalle perdite economiche, Yokoi mette al lavoro Shigeru Miyamoto su un altro arcade, che esce nel 1981 diventando immediatamente un successo planetario: “Donkey Kong”. Nel contempo, nel 1980, viene creata la Nintendo of America, una sussidiaria dedita alla distribuzione dei cabinati Nintendo in Nord America: anche questo è un elemento che ritroveremo più avanti.



**Fig. 6 - Un Game & Watch: da notare il particolare controllo a croce che verrà poi riutilizzato anche nei controller NES/Famicom( autore: Federico Leva - fonte: Wikimedia Commons)**



**Fig. 7 - Il Nintendo Color TV-Game 15, precursore della futura console programmabile di Nintendo (autore: MagentaGreen - fonte: Wikimedia Commons)**

### Da Kong al Famicom

Il successo di “Donkey Kong” nelle sale giochi porta molta gente a volerlo giocare anche nelle proprie case. Dopo una controversa trattativa a tre fra Nintendo, Atari e Coleco, quest’ultima si aggiudica la conversione ed offre la cartuccia del gioco abbinato alla sua nuova console ColecoVision. Il successo è immediato: le persone acquistano il ColecoVision perché vogliono “Donkey Kong”. Nintendo capisce che il mercato dei videogiochi domestici può essere un nuovo, fruttifero, settore dove investire e incarica la R&D2 di sviluppare una propria console. Vengono però messi dei paletti: al momento del debutto deve risultare tecnologicamente superiore rispetto alla concorrenza e lo deve rimanere per almeno un anno, e deve anche risultare economica da produrre.

Uemura inizia a studiare il progetto. Pensa inizialmente di utilizzare come base l’hardware di “Donkey Kong”, inarrivabile anche dal ColecoVision, nonostante la conversione del gioco sia qualitativamente buona, ma il costo dei vari componenti, come la CPU Zilog Z80 dell’arcade, non permettono di produrre una console economica. Vengono anche scartate, per lo stesso motivo, le CPU a 16 bit che stanno comparando sul mercato in quel periodo, molto più avanzate ma molto più costose di quelle ad 8 bit. Alla fine, si opta per un microprocessore in commercio da alcuni anni, economico, molto diffuso, e discretamente performante, ossia il MOS 6502. Per ottimizzare i costi, viene deciso di far produrre la CPU da Ricoh, un grosso produttore di componenti elettronici ed



**Fig. 8 - La CPU prodotta da Ricoh per il Famicom/ NES, qui in versione PAL (autore: PantheraLeo1359531 - fonte: Wikimedia Commons)**







**Fig. 9 - Il Nintendo Family Computer, più noto come Famicom (autore: Evan-Amos - fonte: Wikimedia Commons)**

integrati, con il quale viene rivisto il core del processore: dal 6502 viene rimossa la logica di calcolo BCD e viene integrato un generatore sonoro programmabile, o PSG, da "Programmable Sound Generator" (fig. 8). Per aiutare la CPU nella gestione video, le viene affiancato un altro integrato denominato PPU ("Picture Processing Unit"), capace di generare un'immagine con risoluzione di 256x240 pixel e di poter gestire fino a 64 sprite differenti: un vero salto di qualità rispetto ai chip grafici della concorrenza. Viene scartata l'idea di dotare la console di periferiche di massa quali i floppy disk in favore delle più economiche e diffuse nonché affidabili e sicure cartucce ROM. Per far capire che si tratta di un dispositivo da gioco vengono scelte di proposito forme e colori tali che nessuno, vedendo la console, possa confonderla con un altro tipo di prodotto.

Il primo prototipo viene prodotto nel mese di ottobre del 1982: la console è un piccolo oggetto di plastica bianca con inserti rossi, con un grosso sistema di rilascio delle cartucce posto al centro della parte superiore. Questo dispositivo viene suggerito da Yokoi, che pensa possa essere un elemento di divertimento per i ragazzi, e Uemura adotta l'idea. I joystick, collegati in modo fisso alla console, sono di colore rosso/oro e presentano un'altra innovazione di Yokoi, il D-Pad: questo dispositivo di controllo a 4 direzioni viene mutuato direttamente dai Game&Watch. La console è pronta, deve essere solo "battezzata" e poi presentata alla stampa. Inizialmente si pensa al nome GameCom ma poi si opta per Famicom, una contrazione di "FAMIlY COMputer", ossia un dispositivo destinato a tutta la famiglia. C'è però un problema: Sharp produce già una linea di forni elettrici chiamati "FamiCon" (con la "n" finale). Per evitare possibili azioni legali a causa di eventuali violazioni dei diritti sul nome, il 15 luglio 1983 la console viene presentata come "Family Computer" (fig. 9), e questa denominazione viene utilizzata in tutte le comunicazioni di Nintendo e le pubblicità, anche se la console è nota a tutti come Famicom. Nintendo in seguito si accorda con Sharp sul termine "Famicom" per l'uso

esclusivo sui sistemi da gioco elettronici ma "Family Computer" resta comunque il nome ufficiale della console, così come sarà sempre riportato sul dispositivo stesso nonché sulla sua scatola.

### Dal Famicom al NES

Presentata lo stesso giorno dell'SG-1000 di Sega, la console inizia subito benissimo, favorita da un prezzo di vendita di 14.800 yen, che non è così basso come desiderato da Yamauchi ma è comunque di 200 yen inferiore a quello del dispositivo di Sega. Insieme alla console sono disponibili, al momento del lancio, anche 3 giochi che altro non sono che altrettante conversioni di arcade Nintendo: "Donkey Kong", "Donkey Kong Jr." e "Popeye". Nei primi 2 mesi il Famicom vende 500.000 unità ma pian piano calano le vendite a causa di un problema: sempre più console



**Fig. 10 - Vs. Dr. Mario, uno degli arcade Nintendo basati sul VS. System, derivato dall'hardware del NES (autore: 17daysolderthannes - fonte: Wikimedia Commons)**







**Fig. 11 - Il caricamento delle cartucce sul NES è stato studiato per ricordare quello delle videocassette nei videoregistratori dell'epoca (autore: Yagamichega- fonte: Wikimedia Commons)**

vengono mandate in riparazione per via del fatto che con alcuni giochi la console non si avvia. Viene riscontrato un difetto di fabbricazione che affligge la scheda madre. Yamauchi prende una decisione molto forte: nonostante l'operazione sia economicamente onerosa, viene deciso di sostituire in garanzia tutte le schede madri difettose. Questo serve, secondo il suo pensiero, a creare nei confronti della società una reputazione di serietà ed affidabilità, e il pubblico apprezza e ripaga lo sforzo di Nintendo: il Famicom riprende a macinare numeri di tutto rispetto diventando, alla fine del 1984, la console più venduta in Giappone. Agli inizi del 1985 la console ha venduto 2,5 milioni di unità: tale successo spinge Yamauchi a decidere di commercializzare la console anche negli Stati Uniti. Quel mercato, benché potenzialmente remunerativo, è molto rischioso: il Nord America ha appena affrontato quella che è considerata la più grossa crisi del settore dei videogiochi, con il fallimento di molte aziende ed il ritiro dal commercio di diversi sistemi. C'è però un problema: Nintendo of America ha già provato l'anno prima ad introdurre il Famicom nel mercato USA, prima cercando un accordo con Atari (saltato per la storia della conversione di "Donkey Kong"), e poi cercando di piazzarlo direttamente

nei negozi ma i rivenditori non ne hanno voluto sapere di una console da gioco. D'altronde è il 1983 ed il "botto" dei videogiochi fa ancora rumore. Non volendo rinunciare a quel mercato, e vedendo come, nonostante la crisi delle console domestiche, il mercato dei cabinati da bar va sempre bene, Nintendo decide di sviluppare una piattaforma per arcade basata sull'hardware del Famicom, il "Nintendo VS. System": questo sistema usa la stessa CPU del Famicom a cui viene affiancato, a seconda del gioco, un kit composto da una ROM ed una PPU dedicata (fig. 10). Il VS. System diviene in breve il sistema arcade più venduto, e resta in produzione per diversi anni, totalizzando alla fine della sua vita commerciale il considerevole numero di 100.000 unità vendute. Questa cosa è importante perché il VS. System è, a tutti gli effetti, il primo debutto, nel 1984, del Famicom sul suolo americano.

Mentre le vendite del Famicom e del VS. System procedono spedite, Uemura viene incaricato di studiare un modo per rendere il Famicom vendibile in Nord America. Dato che la crisi dei videogiochi ha di fatto creato una repulsione verso le console da gioco, viene deciso di cambiare abito al dispositivo: via le plastiche colorate ed appariscenti, via il caricamento delle cartucce sul lato superiore. Il







dispositivo non deve ricordare una console da gioco ma un comune elettrodomestico: e per questo viene scelta la forma di un... videoregistratore! Sì, Uemura decide di camuffare la console da un dispositivo elettronico che in quegli anni si sta diffondendo molto nelle famiglie. Anche il caricamento frontale delle cartucce mima proprio l'inserimento di una videocassetta in un videoregistratore: l'utente deve avere l'impressione di usare un comune apparecchio elettronico e non una console da gioco. Il centro stile di Nintendo of America aiuta nello sviluppo del contenitore in modo da realizzare qualcosa di appagante per i gusti dei nord-americani: le due tonalità di grigio e la forma squadrata servono proprio a completare il "feeling" di elettrodomestico (fig. 11).

Si decide anche di abbandonare il nome di "Family Computer" e di optare per "Nintendo System", in modo da non richiamare in nessun modo il legame con un computer o una console e di evitare questi termini incriminati. Viene in seguito deciso di inserire nel mezzo del nome la parola "Entertainment", in modo da far passare il messaggio che il dispositivo sia per l' "intrattenimento" e non qualcosa con cui giocare. Ed i nomi sono scelti accuratamente anche per le varie componenti: il corpo centrale della console viene denominato "Control Deck" mentre le cartucce gioco assumono il nome di "Game Pak", l'unico richiamo all'origine ludica del dispositivo.

Il Nintendo Entertainment System, abbreviato in "NES", viene presentato al Consumer Electronic Show che si tiene nel mese di giugno del 1985. Viene però messo in commercio solo alcuni mesi dopo, ed a zone: si parte ufficialmente il 18 ottobre da New York e poi, agli inizi del 1986, da Los Angeles, mentre la distribuzione su tutto il territorio americano parte solo il 27 settembre 1986.

Anche il lancio nel resto del mondo avviene per zone e con diversi distributori. La prima zona comprende la Scandinavia e la maggior parte dell'Europa continentale come Francia, Germania, Spagna, in cui inizia a partire dalla fine del 1986. Per la seconda zona, che comprende Italia, Irlanda, Gran Bretagna, Australia, Nuova Zelanda ed India, la distribuzione inizia nel 1987 e viene presa in carico per la maggior parte da Mattel. Il NES arriva poi in Corea del Sud, ma solo nel 1989, ed addirittura nel 1993, circa 2 anni prima della sua dismissione, in Brasile e Sud Africa.

### La vita commerciale

In Nord America, il mercato principale, il NES viene proposto al momento del lancio nel solo "Deluxe Set", che costa 179,99 dollari e che comprende il robot "R.O.B.", che verrà usato solo in 2 giochi tra cui "Gyromite" incluso nella confezione, e la pistola ottica "Zapper", che verrà

usata in una ventina di giochi fra cui "Duck Hunter" incluso con la console. In seguito la sola console viene messa in vendita a 89,99 dollari, oppure, se si opta per la versione con la doppia cartuccia di "Super Mario Bros." e "Duck Hunt", a 99,99 dollari.

La potenza della console nonché la qualità dei suoi giochi sono entrambi fattori che spingono le vendite del NES: si stima che nel solo 1988, ad un anno dall'inizio della distribuzione su tutto il suolo americano, il NES abbia venduto più di 7 milioni di unità, lo stesso numero che il Commodore 64 ha fatto registrare nei suoi primi 5 anni di vita. Nel 1990 il NES è presente nel 30% delle case americane ed il Famicom nel 37% di quelle giapponesi. Tra la fine degli anni '80 e l'inizio degli anni '90 iniziano a comparire le prime console a 16 bit, tra cui il TurboGrafx-16 ed il Sega Mega Drive. Per restare al passo coi tempi, Nintendo aggiorna il NES e presenta, alla fine del 1990, il Super NES, o Super Nintendo Entertainment System. Nonostante la concorrenza di queste macchine tecnicamente superiori, il NES regge il colpo e continua a vendere molto bene, grazie anche all'offerta di giochi che, con il passare del tempo, spremono sempre più le capacità della console. Nintendo si vede costretta a mettere in vendita il Super NES a soli 10 dollari in più rispetto al NES così che la poca differenza di prezzo fra il sistema ad 8 bit e quello a 16 bit invogli le persone a comprare quest'ultimo.

Il NES resta in commercio fino al 1995, quando Nintendo decide di toglierlo dal mercato. Il Famicom ha invece una vita molto più lunga: in Giappone resta in vendita fino a 2003 e viene ritirato dal mercato perché la scarsità di pezzi di ricambio rende ormai impossibile la riparazione delle eventuali unità guaste.

Il NES/Famicom è stata la prima console da giochi a segnare un record incredibile di vendite: si stima che ne siano state vendute quasi 62 milioni di unità, di cui 42,5 milioni di NES. È stata anche una delle più longeve, essendo rimasta in commercio (in Giappone) per 20 anni consecutivi.

*Per questo numero è tutto amici, appuntamento al prossimo numero per la seconda e conclusiva parte dell'articolo. (NdFF).*





# Il Fortran sul Commodore 64

di Francesco Fiorentini

Chi segue i miei articoli su RetroMagazine World sa che preferisco recensire applicazioni e linguaggi, seppur di tanto in tanto non disdegni di provare e commentare anche i giochi. In questo mio ennesimo articolo voglio però rimanere fedele alla mia passione parlandovi di un linguaggio di programmazione: il Fortran.

Il Fortran è uno dei primi linguaggi di programmazione ad essere stato sviluppato, la sua progettazione risale addirittura alla prima metà degli anni '50, quando, verso la fine del 1953, **John W. Backus** propose ai suoi superiori dell'IBM di sviluppare un'alternativa più pratica al linguaggio assembly per la programmazione del computer mainframe IBM 704.

Da allora sono passati esattamente 70 anni, ma questo linguaggio è tuttora utilizzato dalle comunità scientifiche e ingegneristiche; ovviamente in tutti questi anni il

linguaggio ha subito diverse modifiche per cercare di adattarsi alle rinnovate esigenze del mercato e degli sviluppatori.

## Il Fortran ed il Commodore 64

Come abbiamo già avuto modo di toccare con mano, il Commodore 64 ha ospitato qualsiasi tipo di software, ludico e non, ed ovviamente, nella lista dei linguaggi di programmazione disponibili per questo home computer, non poteva mancare il Fortran.

Dalle mie ricerche risultano disponibili almeno due versioni di Fortran per l'8-bit più venduto di casa Commodore, il **Nevada Fortran** e l'**Abacus Fortran-64**. La mia scelta, per motivi di praticità, è ricaduta appunto su quest'ultimo, dato che il Nevada Fortran richiede la cartuccia CP/M per funzionare. Ho quindi ipotizzato che il prodotto Abacus fosse più accessibile a chi volesse provare a testarlo su macchina reale. Questo non vuol dire che in futuro non farò anche un test del Nevada Fortran.

Ah, chiunque fosse a conoscenza di altre versioni di Fortran è pregato di farmelo sapere. Adoro mettere le mani su software semi-sconosciuti. :-)

## Abacus Fortran-64

Il linguaggio Fortran di Abacus è stato rilasciato nel 1988 ad opera di Bob Stovere e Tim Adams. Il pacchetto si compone di un disco contenente il compilatore, il linker, le librerie, un comodo traduttore (vedremo successivamente a cosa serve) ed un manuale che, in perfetto stile Abacus del periodo, è ridotto all'essenziale.

Una volta caricato il Fortran-64 verremo accolti da un menù che contiene tutte le opzioni per creare il nostro primo programma in Fortran, compilarlo, linkarlo e generarne un eseguibile. Quello che balza subito agli occhi però è la mancanza di un editor di testo per creare il file sorgente: come fare quindi?

Il file sorgente può essere creato in due modi differenti. Il primo consiste nell'utilizzare un qualsiasi elaboratore di testi che produca file di output ASCII sequenziali, mentre il secondo è quello di utilizzare l'editor integrato

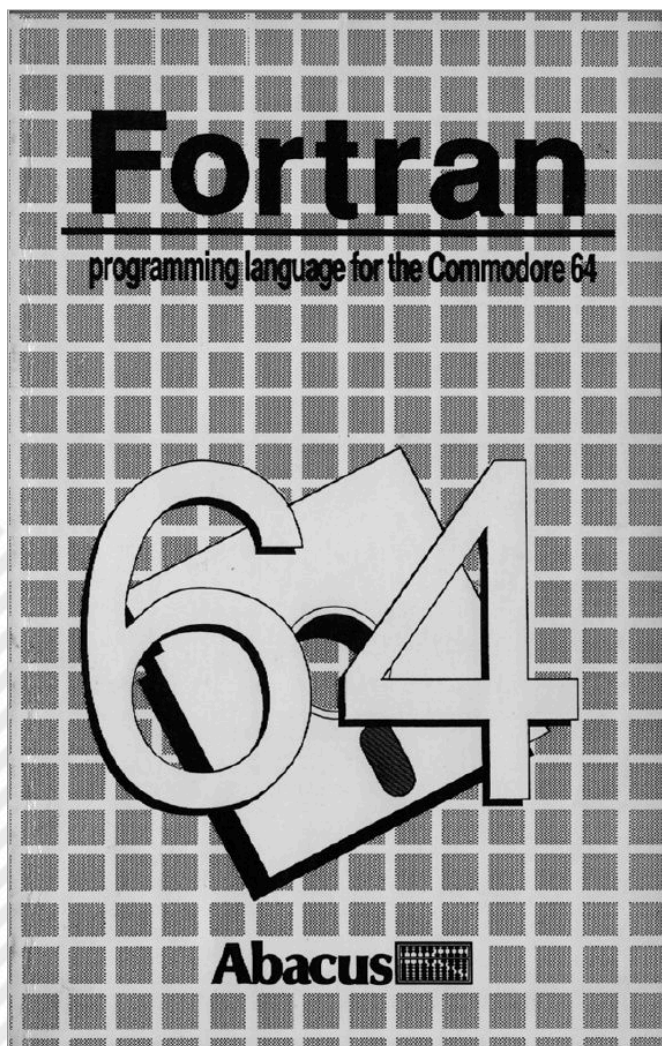


Fig. 1 - Abacus Fortran-64







Fig. 2 - Caricamento di Abacus Fortran-64

nel BASIC V2. Utilizzando l'opzione F2 – Back to Basic, torneremo al prompt del BASIC e potremo cominciare a scrivere il codice. Ogni riga deve essere inserita come nel BASIC, ma con una piccola eccezione: i due punti (:) devono apparire nella riga tra il numero di riga e l'istruzione Fortran. Dopo che il programma è stato scritto potrà essere salvato su disco con il classico comando SAVE del BASIC.

A questo punto, tornando al menù dell'Abacus Fortran-64 tramite il comando SYS49152 (attenzione, leggete la nota sotto), potremo procedere a trasformare il file BASIC PRG in un file SEQ (sequenziale) tramite il comando F6 – Translator.

**Nota bene:** per consentire il rientro in Fortran-64 senza ricaricare, è necessario eseguire un SYS49152 subito dopo il ritorno al BASIC. Dopo aver salvato il file, è possibile eseguire un secondo SYS49152 per tornare al menù principale.

### Il nostro primo programma in Fortran-64

Data la possibilità di utilizzare l'editor in Basic, ho scelto



Fig. 3 - Il menù principale

proprio quest'opzione per scrivere il mio primo programma in Fortran con il Commodore 64.

Una volta tornato al Basic ho digitato quindi SYS49152 e dopodiché ho scritto il seguente programma:

```
10 : program test
15 :* this program will write a
16 :* list of the odd numbers
16 :* from 1 to 10 to screen
20 : integer a,b
30 : open 3,3
40 : do 10 i = 1, 10, 2
50 : write (3,100) i
60 : 10 continue
65 : close 3
70 : stop
80 : 100 format(i5)
90 : end
```

e l'ho poi salvato con il comando: **SAVE "ODDNUM", 8**

In seguito ho digitato di nuovo SYS49152 per tornare al menù dell'Abacus Fortran-64 e l'ho trasformato in un file sequenziale tramite il comando F6 – Translator.

Il programma Translator ci richiede il nome del file che abbiamo creato e salvato in ambiente BASIC:

**enter program file name: oddnum**

Poi ci viene chiesto con quale nome vogliamo salvare il file sequenziale:

**enter sequential file name: oddseq**

A questo punto siamo pronti per compilare il nostro programma.

### E adesso compiliamo...

Dal menù principale sceglieremo la voce F1 – Compiler.

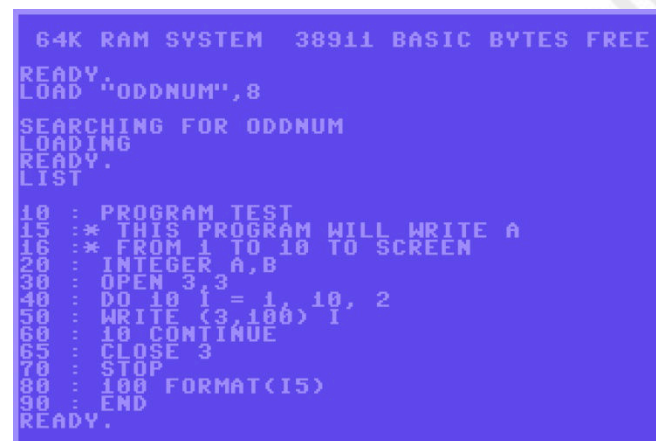


Fig. 4 - Il programma appena digitato





```

*****
***** TRANSLATOR *****
*****
ENTER PROGRAM FILE NAME: ODDNUM
ENTER SEQUENTIAL FILE NAME: ODDSEQ
SEQUENTIAL FILE EXISTS.
REPLACE? (Y/N CR=Y)
TRANSLATION COMPLETE.
PRESS RETURN WHEN READY:

```

Fig. 5 - Il programma 'tradotto' con il translator

Il compilatore ci chiederà di fornire il nome del nostro programma, cioè il file sequenziale che abbiamo creato con il traduttore e poi di fornire il nome del file oggetto che vogliamo creare:

```

enter source file name: oddseq
enter object file name: oddobj2

```

A questo punto inizierà la fase di compilazione del programma e verranno segnalati a video tutti gli eventuali errori che il compilatore avrà individuato. Ho trovato veramente interessante questa opzione perché ogni errore viene indicato esattamente sotto la riga che lo contiene, facilitando decisamente la vita del programmatore.

Gli errori verranno segnalati usando il seguente formato:

```
***** ERROR xxx yyy *****
```

xxx = codice dell'errore

yyy = posizione del carattere nella riga che ha causato l'errore (esclusi gli spazi)

Per la lista dei 64 codici errore, si veda il manuale alla pagina 47: Appendice D - Error Description.

```

0000  PROGRAM TEST
0001  * THIS PROGRAM WILL WRITE A
0002  * FROM 1 TO 10 TO SCREEN
0003  INTEGER A,B
0004  OPEN 3,3
0005  DO 10 I = 1, 10, 2
0006  WRITE (3,100) I
0007  10 CONTINUE
0008  CLOSE 3
0009  STOP
0010  100 FORMAT(I5)
0011  END

```

```

SOURCE FILE NAME: ODDSEQ
OBJECT FILE NAME: ODDOBJ2

```

```

0000 ERRORS DETECTED.
0012 SOURCE LINES READ.

```

```
PRESS RETURN WHEN READY:
```

Fig. 6 - La compilazione

Tutto fatto? Certo che no, manca ancora un passaggio fondamentale prima di poter testare l'esecuzione del nostro programma. Utilizzando il linker dovremo collegare il codice oggetto, cioè la traduzione del codice sorgente in linguaggio macchina, con le librerie del linguaggio necessarie per poter eseguire il programma.

...e finalmente il Linker!

Per utilizzare il linker non dovremo fare altro che scegliere F3 - Linker dal menù principale e poi comunicare il nome del nostro programma oggetto, nel nostro caso ODDOBJ2.

Il linker automaticamente collegherà tutti i moduli necessari all'esecuzione del nostro codice e questi, a scelta, saranno anche visualizzati. Infine il linker ci informerà a quale indirizzo si trova il nostro codice eseguibile:

#### modules loaded:

```

test      $0850
clwbuf    $0903
set-up    $0915
in2as     $09d3
fiwbuf    $0abf
wformat   $0c2d
faterr    $11f6

```

#### runable code resides

```
from $0850 to $1230
```

Al termine del processo di linking, il programma eseguibile sarà residente in memoria. Utilizzando il comando BASIC LIST, vedremo sulla prima riga un comando REM con il nome del programma Fortran, mentre nella seconda riga un comando SYS con l'indirizzo di esecuzione del programma. Il resto del programma è in linguaggio macchina. A questo punto il programma può essere salvato su disco o essere eseguito con il comando BASIC RUN.

#### L'output del comando LIST:

```

10 rem program test
20 sys02128

```

Da notare che la SYS punta all'indirizzo 2128, in esadecimale \$850. Sì, proprio lo stesso indirizzo di partenza indicato dal linker.

#### Un'occhiata al codice







```

LINKER
ARE YOU SURE? (Y/N CR=Y)
ENTER OBJECT FILE NAME: ODDOBJ2
LIST MODULES? (Y/N CR=Y)
HARDCOPY? (Y/N CR=N)

MODULES LOADED:
TEST      $0850
CLWBUF    $0903
SET-UP    $0915
IN2AS     $09D3
FIWBUF    $0ABF
WFORMT    $0C2D
FATERR    $11F6

RUNABLE CODE RESIDES
FROM $0850 TO $1230
LINK COMPLETE
READY.

```

Fig. 7 - Il programma linkato!

```

FATERR $11F6
RUNABLE CODE RESIDES
FROM $0850 TO $1230
LINK COMPLETE
READY.
SAVE "ODDFTN",8
SAVING ODDFTN
READY.
RUN
  1
  3
  5
  7
  9
READY.
LIST
10 REM PROGRAM TEST
20 SYS02128
READY.

```

Fig. 8 - Il programma in esecuzione

10 : **program test**

Semplicemente il nome del programma

15 :**\*** **this program will write a**

16 :**\*** **list of the odd numbers**

16 :**\*** **from 1 to 10 to screen**

Righe di commento

20 : **integer a,b**

Definisce due variabili di tipo intero (non utilizzate)

30 : **open 3,3**

Apre una connessione con un device. Personalmente per identificare il video ho fatto riferimento a questa tabella: Dispositivo numero 0 | tastiera (non contato come primo dispositivo!)

Dispositivo numero 1 | datasette | numero secondario 0 = per il caricamento; 1 = per la scrittura; 2 = per la scrittura di un file seguito da un segno EOT (End Of Tape)

Dispositivo numero 2 | modem / RS-232 | numero secondario 0

Dispositivo numero 3 | schermo | numero secondario 0 o 1

Numero dispositivo 4 o 5 | stampante | numero secondario 0 = lettere maiuscole/simboli grafici 7 = lettere maiuscole/minuscole

Numero dispositivo 8 fino a 15 | unità disco | numero secondario 2-14 = canali dati[1]; 15 canale per i comandi

40 : **do 10 i = 1, 10, 2**

DO label varnam = is,if [,ic]

dove: label è un numero intero senza segno

varnam è una variabile semplice

is è una costante o una variabile intera (valore iniziale)

if è una costante o variabile intera (valore finale)

ic è una costante intera firmata (incremento)

Ciclo DO da 1 a 10 con step 2

50 : **write (3,100) i**

Stampa a video la variabile i (formattata alla riga 80)

60 : **10 continue**

Fine del loop DO

65 : **close 3**

Chiude la connessione con il device (vedi linea 30)

70 : **stop**

Termina l'esecuzione del programma e torna al Basic

80 : **100 format(i5)**

Formatta la stampa di un intero a 5 cifre

90 : **end**

Indica la fine di un'unità di programma

Adesso tocca a voi! Provate a scrivere il vostro primo programma e compilatelo allo stesso modo.

Buon divertimento!

## Links Utili

### Download di Abacus Fortran-64 e del manuale:

- <https://www.lyonlabs.org/commodore/onrequest/collections.html>

### Pagina WIKI sul Fortran:

- <https://it.wikipedia.org/wiki/Fortran>





# Giocare a vite infinite - La sfida finale

di Marco Pistorio

Un saluto a tutti i lettori di RetroMagazine World. Torno ancora su questo argomento perché credo valga la pena ed interessi a tanti "crackers in erba" :)

Vi suggerisco di rileggere gli articoli precedenti che riguardano questa tematica, se non li avete ben presenti, in quanto in questo articolo andrò a sviluppare alcuni elementi di cui ho già parlato proprio nei precedenti articoli, pubblicati rispettivamente nel numero 03 di RetroMagazine ("Vite infinite con il commodore 64") e nel numero 23 ("Giocare a vite infinite sul C64 - "La sfida continua").

Faccio un breve excursus. Ho presentato un tool che permette di esaminare istantanee diverse della memoria del C64 e che facilita l'individuazione della locazione contenente il numero di vite a disposizione all'interno di un generico gioco.

Ho poi mostrato come, attraverso delle istruzioni NOP, disabilitare la variazione del contenuto della locazione individuata, cosa che altrimenti avverrebbe, di norma, con un decremento.

Infine, dopo la modifica del codice del gioco con gli opportuni cambiamenti, ho mostrato come salvare tutto su disco ed ottenere quindi una versione "personalizzata" del gioco modificato.

Questa è una strada che, in effetti, è percorribile. Tuttavia molto spesso si crea un "trainer", un programma per ottenere un risultato analogo.

Ci concentreremo quindi su questa strada e cercheremo di comprendere cosa è esattamente un trainer, come opera e quali vantaggi comporta il suo utilizzo.

## IL TRAINER

Un trainer è un ausilio, un semplice programma che modifica per noi il codice del gioco per il quale è realizzato e che ci consente, ad esempio, di giocare a vite infinite. Qual è il vantaggio di adoperare un trainer? Il vantaggio è essenzialmente quello di caricare il gioco e lasciare all'utente la scelta di giocare al gioco in modalità normale, oppure giocare in modalità 'facilitata', ad esempio evitare di perdere le vite a disposizione durante il gioco.

Il vantaggio quindi mi sembra piuttosto chiaro.

Ma come si fa a realizzare un trainer?

Innanzitutto bisogna avere ben chiaro quale aspetto si

intende modificare del gioco in esame. In genere è proprio la possibilità di giocare a vite infinite, ma in altri casi un trainer permette di modificare altri comportamenti o dettagli del gioco. Il numero delle spade, delle risorse a disposizione durante il gioco, oppure la velocità del gameplay oppure ancora la possibilità di terminare un livello e passare a quello successivo immediatamente.

Se intendiamo lavorare solo sulle vite infinite, resta valido tutto ciò che abbiamo già visto nei due articoli precedenti per arrivare a determinare la locazione di memoria che contiene tale informazione durante l'esecuzione del gioco. Ma non cambieremo in maniera diretta il codice del gioco nè salveremo una copia così modificata del gioco stesso. Invece, scriveremo un trainer.

Il gioco NON partirà dalla sua sys iniziale, bensì dal nostro trainer. Il trainer ci porrà la domanda: "Vuoi giocare a vite infinite?" e se risponderemo "No", il trainer non farà assolutamente nulla, terminerà e ci farà partire il gioco. Se invece risponderemo "Sì", verrà eseguita la modifica opportuna del codice del gioco dall'interno del trainer, quindi il trainer terminerà e, alla partenza del gioco, avremo le vite infinite!

## CONSIDERAZIONI IMPORTANTI

Il trainer occuperà parte della memoria del computer, e per agire, dovrà essere presente insieme al codice del gioco che andrà, eventualmente, a modificare.

Per fortuna, di solito i trainer occupano poca memoria.

Altro concetto importante, già espresso per la verità, è che il gioco non può partire direttamente, ma dovrà essere lanciato dal trainer. Solo così avremo la possibilità di scegliere se giocare al gioco senza alcuna modifica, oppure se lo giocheremo con qualche "facilitatore" (ad esempio vite infinite).

Se viene scelto di giocare senza alcuna modifica, il codice del gioco dovrà rimanere immutato.

## ANATOMIA DI UN TRAINER REALE

Ho realizzato recentemente, a firma del gruppo "HOKUTO FORCE" (che saluto, approfittando di queste righe) un trainer per il gioco "TSI\_MAZE\_MAN". Il mio trainer agisce su ben 5 aspetti di questo gioco.







Vi mostro qui una sua versione semplificata che consente soltanto di giocare a vite infinite con questo gioco oppure no.

```
.pc=$0801 "basic starter"
:BasicUpstart(start)

.pc=$7A00 "start code trainers"
start:

    lda #0
    sta $d020
    sta $d021

////////////////////////////////////

    ldx #0
su_0:
    lda testo0,x
    jsr $ffd2
    beq cont
    inx
    jmp su_0

cont:

////////////////////////////////////

    ldx #0
su_1:
    lda testo1,x
    beq leggi_tasto_premuto_1
    jsr $ffd2
    inx
    jmp su_1

leggi_tasto_premuto_1:
    jsr $FFE4

    cmp #89                //Y
    beq si_1
    cmp #78                //N
    beq no_oppure_return_1
    cmp #13                //RETURN
    beq no_oppure_return_1
    jmp leggi_tasto_premuto_1

si_1:
    jsr $ffd2

    lda #$EA                //opcode NOP
    sta $832c
    sta $832d
    sta $832e

    sta $832f
    sta $8330

    lda #13
    jsr $ffd2
    jmp continua_1

no_oppure_return_1:
    lda #78
    jsr $ffd2
    lda #13
    jsr $ffd2
    jmp continua_1

////////////////////////////////////

continua_1:

    jmp $7FE0
```

```
testo0:
.byte 147,05
.text "-----"
.text "TSI MAZE MAN (C) 1983 TRAINED BY MP SOFT"
.text "-----"
.byte 13
.byte 13,158
.byte 0
```

```
testo1:
.text "UNLIMITED LIVES <Y/N> "
.byte 0

.pc=$7FE0 "start dump game data"
.import binary "dump2.bin",2
```

Si tratta di poche righe di codice, con sintassi KICK ASSEMBLER.

Subito dopo aver impostato il colore nero per schermo e sfondo, viene scritto a video un semplice testo che presenta il gioco e questo suo trainer "TSI MAZE MAN (C) 1983 TRAINED BY MP SOFT"

Successivamente, viene posta la faticosa domanda, sempre sullo schermo: "UNLIMITED LIVES <Y/N> " e viene controllata la scelta effettuata dall'utente.

Sfruttando la routine del kernal \$FFE4 siamo in grado di conoscere quale tasto viene premuto.

Se è stato premuto N oppure è stato battuto return viene scritto a video il tasto premuto (vale N in entrambi i casi), poi diamo un return ed infine il gioco parte.

Se invece è stato premuto Y, il trainer entra in funzione modificando le opportune locazioni di memoria, ovvero caricando nell'accumulatore il valore #\$EA (che corrisponde al buon vecchio "NOP") e poi scrivendolo nelle locazioni \$832C, \$832D, \$832E, \$832F, \$8330.

Lasciando invece tali locazioni ai valori originari le vite verrebbero decrementate normalmente durante il gioco. Anche in questo caso, diamo un return sullo schermo (con le istruzioni lda #13 e jsr \$ffd2) ed infine il gioco parte (con jmp \$7FE0).

So che questa ultima parte è tosta ma vedere il codice del trainer era irrinunciabile :)

Un caro saluto a tutti i lettori di RetroMagazine World ed alla prossima!

Per chi fosse curioso, riporto qui di seguito il link del gioco TSI MAZE MAN con il mio "vero" trainer:

<https://csdb.dk/release/?id=235283>





# Strani prodotti

## (per C64 e per principianti)

di Eugenio Rapella

Francesco sta imparando la moltiplicazione tra le frazioni. Ivy, l'insegnante, nota che Francesco ha scritto:

$$\frac{1}{4} \cdot \frac{8}{5} = \frac{18}{45}$$

"Non puoi eseguire la moltiplicazione in questo modo, affiancando le cifre" dice Ivy; Francesco sostiene che l'operazione è corretta: semplificando, in entrambi i membri si ottiene 2/5.

Il "quesito" proviene dal libro "Professor Stewart's Cabinet of Mathematical Curiosities"; l'autore invita il lettore a trovare per quali combinazioni di cifre (lo zero non viene considerato in questo problema) il "metodo di Francesco" funziona, a parte le situazioni banali in cui il numeratore e il denominatore di ciascuna delle due frazioni sono uguali tra loro. Nel capitolo dedicato alle soluzioni, Ian segnala che vi sono esattamente 14 soluzioni (vi sono sette scelte per le quattro cifre più le altre sette che si ottengono ... capovolgendo le frazioni).

Riuscite a trovarle anche senza l'aiuto del nostro fidato C64?

Il mio Commodore 64 era quasi offeso dalla semplicità del quesito, ma gli ho spiegato che si trattava di fare un po' di stretching prima di impegnarlo in compiti più complessi.

Ecco allora come ha affrontato il problema:

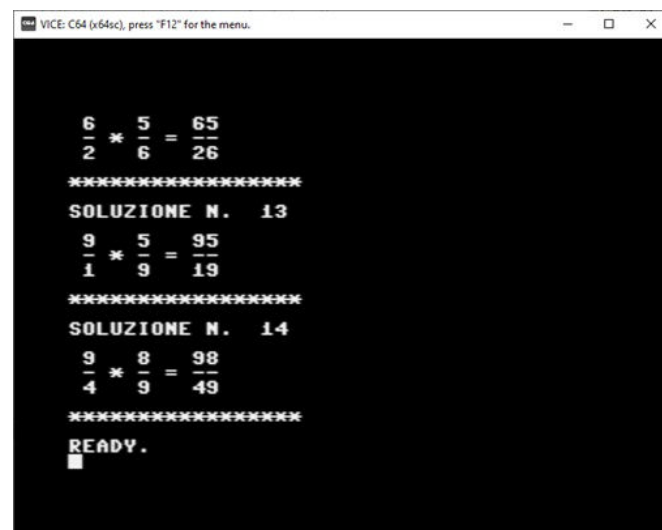
```
100 poke 53280,0:poke 53281,0:print chr$(5)
110 for a=1 to 9:for b=1 to 9
120 for c=1 to 9:for d=1 to 9
130 if a=b and c=d then 220
140 n=10*b+d:m=10*a+c
150 if a*c*n<>b*d*m then 220
160 k=k+1:print:print"soluzione n. ";k:print
170 print a;" ";c;" ";m
180 print" - * - = --"
190 print b;" ";d;" ";n
200 print:print"*****"
220 next:next:next:next
```

Se A e B sono il numeratore e il denominatore della prima frazione e C e D il numeratore e il denominatore della seconda, il "metodo di Francesco" ha successo se la frazione  $(A*C)/(B*D)$  ha lo stesso valore di  $(10*A+C)/(10*B+D)$ : "affiancando" la cifra A alla cifra C si ottiene il numero  $10*A+C$ ; lo stesso avviene per il denominatore. Ecco dunque che, dopo le colorazioni di rito (alla 100), si aprono ben quattro cicli "for-next" (istr. 110 e 120) che si chiudono alla 220.

Il C64 si permette il lusso di saggiare tutte le  $9*9*9*9=6561$  possibilità: alla 130 si controlla se si tratta di una soluzione banale (in questo caso si va direttamente alla 220).

In caso contrario, alla 140 si calcolano numeratore M e denominatore N della frazione al secondo membro. Se  $(A*C)/(B*D)$  è diverso da  $M/N$ , ovvero se  $A*C*N \neq B*D*M$  (istr. 150) il "metodo di Francesco" fallisce e si va alla 220, in caso contrario si è trovata una delle 14 soluzioni che viene stampata.

Ecco qui la schermata a fine lavoro:



Il metodo di Francesco funzionerà anche con tre frazioni?

Il programma precedente può essere facilmente modificato:

```
100 poke 53280,0:poke 53281,0:print chr$(5)
110 for a=1 to 9:for b=1 to 9
120 for c=1 to 9:for d=1 to 9
```







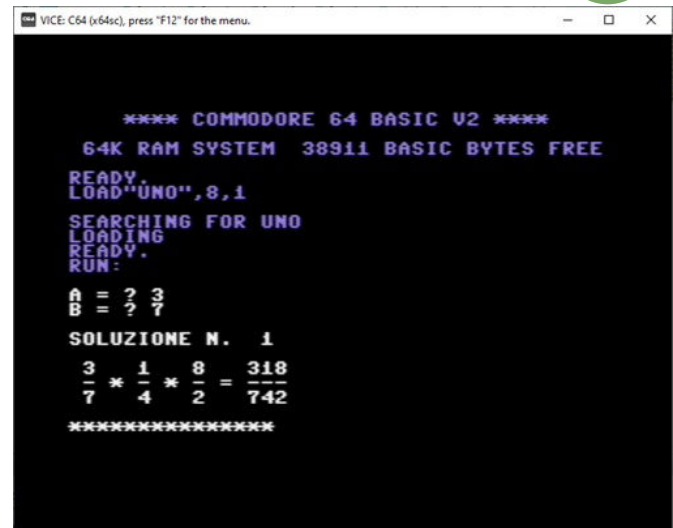
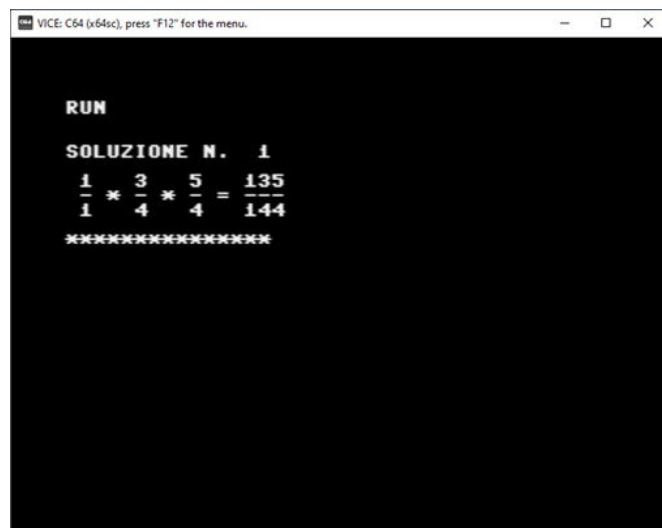
```

130 for e=1 to 9:for f=1 to 9
140 if a=b and c=d and e=f then 230
150 n=100*a+10*c+e:m=100*b+10*d+f
160 if a*c*e*m<>b*d*f*n then 230
170 k=k+1:print:print"soluzione n. ";k:print
180 print a;" ";c;" ";e;" ";n
190 print" - * - * - = ---"
200 print b;" ";d;" ";f;" ";m
210 print:print"*****"
230 next:next:next:next:next:next

```

ma ben sei cicli nidificati da 1 a 9 significa la bellezza di  $9^6=531441$  iterazioni e tempi d'attesa improponibili.

Ecco comunque la prima soluzione trovata dal C64 e una soluzione ottenuta impostando manualmente i valori di A e B (tanto per controllare l'esistenza di altre soluzioni senza dover aspettare un'eternità).



Per la verità, ho provato a modificare il programma per vedere se il nostro C64 trovava qualche soluzione anche con quattro frazioni... sì, sì, ce ne sono, eccone alcune:

$$\frac{2}{1} \cdot \frac{2}{8} \cdot \frac{8}{2} \cdot \frac{5}{8} = \frac{2285}{1828}$$

$$\frac{2}{1} \cdot \frac{3}{3} \cdot \frac{1}{2} \cdot \frac{7}{4} = \frac{2317}{1324}$$

$$\frac{2}{1} \cdot \frac{3}{8} \cdot \frac{6}{9} \cdot \frac{5}{2} = \frac{2365}{1892}$$

E bravo il nostro Commodore 64! Se lo merita davvero.





# Permutazioni (per C64)

di Eugenio Rapella

Le lettere che formano la “parola” EFBDC A sono le prime sei lettere dell’alfabeto. Qual è la parola, sempre formata da queste lettere, che le è immediatamente successiva in ordine alfabetico? Visto quanto richiesto, cercheremo, per quanto possibile, di mantenere inalterate le prime lettere per cui ... partiremo dal fondo.

Le ultime due lettere sono CA; poiché C segue la A nell’ordine alfabetico, se le scambiassimo di posto, avremmo una parola (EFBDAC) precedente a quella in esame. Lo stesso vale per la coppia DC. Continuiamo a risalire esaminando le coppie di lettere contigue; ora è la volta di BD e questa volta, poiché B precede D avremo la possibilità di trovare una lettera a destra di B che ne potrà prendere il posto (al limite sarà la stessa D).

Naturalmente, se la parola di partenza fosse FEDCBA, non troveremmo la coppia adatta in quanto questa parola è l’ultima in ordine alfabetico formata dalle lettere considerate.

Con chi va scambiata la B? Tra tutte quelle a destra di B, ovvero DCA, sceglieremo quella che, in ordine alfabetico, la segue per prima. Siccome queste sono già disposte in ordine alfabetico inverso, basterà partire dall’ultima e risalire fino a trovare la prima che segue la nostra in ordine alfabetico. La A non va bene (A precede B), ma la C passa il test per cui scambieremo di posto la C e la B. Per ora abbiamo EFC seguito da DBA. Siamo agli sgoccioli: ci rimangono da sistemare le lettere dopo la C in modo che la “coda finale” sia ... in ordine alfabetico. Lo scambio che abbiamo effettuato non ha variato l’ordine alfabetico di questa “coda” che risulta essere in ordine alfabetico inverso. Basterà semplicemente rovesciarla: DBA > ABD. Morale: l’anagramma di ABCDEF che segue EFBDC A nell’ordine alfabetico è EFCABD.

Il tutto ha l’aria un po’ complicata, ma è solo apparenza. Rivediamo i passaggi in modo più sintetico con un nuovo esempio: qual è l’anagramma di “CALZONI” che segue questa parola in ordine alfabetico? Usiamo il simbolo “<” per “precede” e “>” per “segue”. Abbiamo C>A<L<Z>O>N>I. Risalendo troviamo che la lettera da scambiare è la “L”. Tra le lettere a destra di L, qual è quella che la segue per prima? Loro sono già disposte in ordine decrescente, per cui, partendo dal fondo, I>L? No! N>L? Sì! Scambiamo L con N: CAN-ZOLI e ora invertiamo l’ordine delle ultime quattro lettere: CANILOZ. In ordine alfabetico, l’anagramma immediatamente seguente a CALZONI è CANILOZ.

Ecco come il nostro fidato C64 affronta e risolve il problema:

```
100 input"scrivi una parola";n$:d=len(n$):dim
a(d)
110 for k=1 to d:a(k)=asc(mid$(n$,k,1)):next
120 print"premi un tasto per iniziare e"
130 print"per passare al successivo"
140 get r$: if r$="" then 140
150 for k=1 to d:print chr$(a(k));:next:print
160 i=d-1
170 if a(i)>=a(i+1) then i=i-1:goto170
180 if i=0 then print:print "fine":end
190 j=d
200 if a(j)<=a(i) then j=j-1:goto200
210 t=a(j):a(j)=a(i):a(i)=t
220 i=i+1:j=d
230 if i<j then t=a(j): a(j)=a(i):
a(i)=t:i=i+1:j=j-1: goto 230
240 goto 140
```

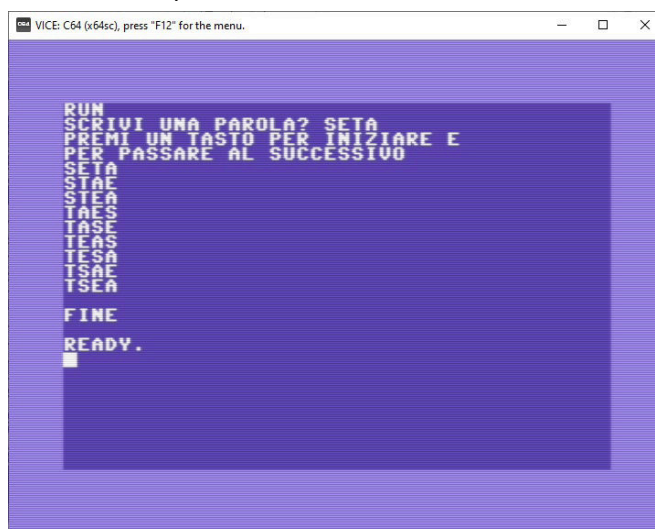
I codici ASCII delle lettere che formano la parola, richiesta alla 100, vengono caricati nel vettore A(..); la dimensione di A(..) è D, pari alla lunghezza della parola (se N\$=“amo”, è D=3 e A=(65,77,79)).

Alla 150 vengono stampate le lettere ora contenute in A(..); il programma vero e proprio inizia alla 160.

A partire dalla fine si confrontano le coppie di lettere contigue fino a trovare la prima coppia per cui A(I) precede A(I+1) (istr. 170): A(I) sarà la lettera da scambiare. Se l’anagramma è l’ultimo in ordine alfabetico, la variabile I raggiunge il valore 0 e il programma ha termine. Alle istruzioni 190-200 si riparte dal fondo alla ricerca della prima lettera A(J) che segue A(I): A(J) sarà la lettera da scambiare con A(I).

Alla 210 si esegue lo scambio e alla 220-230 si inverte l’ordine della coda; alla 240 si torna alla 140 per la stampa e per la ricerca dell’anagramma successivo.

Ecco un esempio di utilizzo:







Ma ... a cosa può servire un programma del genere? A non molto ... però, se la parola iniziale è formata dalle prime N lettere in ordine alfabetico (ad esempio "abc" per N=3), il C64 ci fornisce tutti i possibili anagrammi, ovvero l'elenco delle permutazioni.

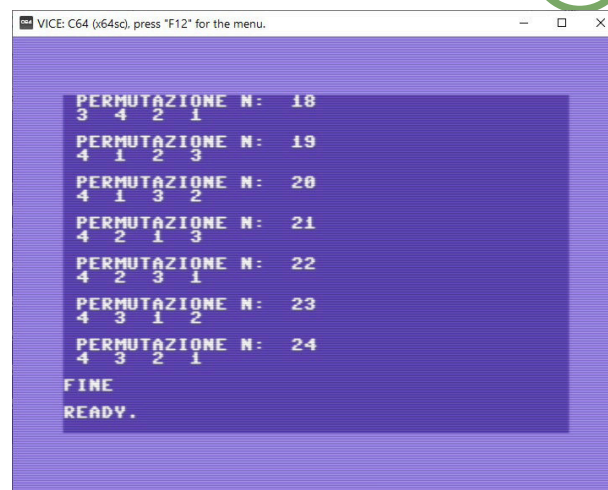
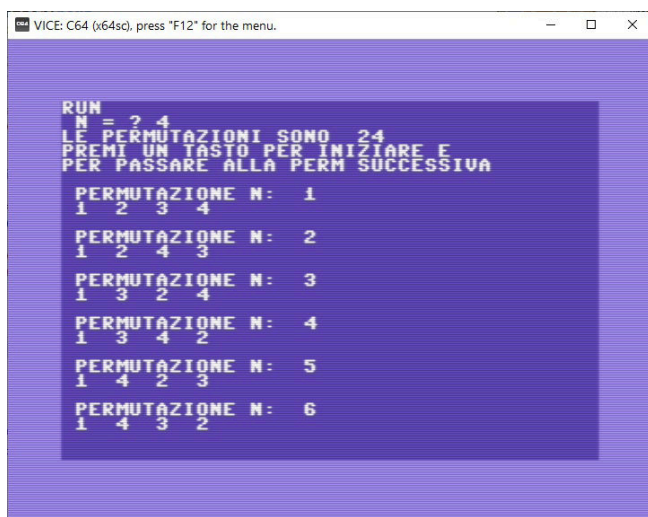
Abbandoniamo l'esempio letterale e riformuliamo il programma utilizzando i numeri da 1 a N. Ecco come:

```
100 input" n = ";n:dim a(n):f=1:c=0
110 for k=1 to n:a(k)=k:f=f*k:next
120 print"le permutazioni sono ";f
130 print"premi un tasto per iniziare e"
140 print"per passare alla perm successiva"
150 get r$: if r$="" then 150
160 c=c+1
170 print:print" permutazione n: ";c
180 for k=1 to n:print a(k);:next:print
190 i=n-1
200 if a(i)>=a(i+1) then i=i-1:goto200
210 if i=0 then print:print "fine":end
220 j=n
230 if a(j)<=a(i) then j=j-1:goto230
240 t=a(j):a(j)=a(i):a(i)=t
250 i=i+1:j=n
260 if i<j then
t=a(j):a(j)=a(i):a(i)=t:i=i+1:j=j-1:goto 260
270 goto 150
```

Dopo aver richiesto N, il nostro C64 carica il vettore A(..) con i numeri da 1 a N e ne approfitta per calcolare il fattoriale di N ( $F=N!=1 \times 2 \times 3 \times \dots \times N$ ) che è il numero delle permutazioni che dovrà generare.

Il resto è identico al programma precedente, la variabile C funge da contatore delle varie permutazioni.

Ecco la schermata iniziale e quella finale di un esempio di utilizzo:



Naturalmente il codice può essere utilizzato come subroutine in programmi in cui è richiesto di generare tutte le permutazioni di N oggetti. Ecco allora un quesito che avevo proposto in una verifica per gli studenti del terzo anno di liceo:

**Permutando in tutti i modi possibili le cifre 1,2,3,4,5 si ottengono 120 numeri naturali. Qual è la somma di questi 120 numeri?**

Ovviamente agli studenti (e ai lettori di RMW) è richiesto di risolvere il problema con una ... scorciatoia, ma il C64 può permettersi di eseguire la somma per esteso. Mettetevi al lavoro, confrontate poi la vostra soluzione con quella ottenuta dal programma qui sotto dove la permutazione contenuta nel vettore A(..) viene trasformata in numero nell'istruzione 130. Ogni cifra contribuisce alla creazione del numero V a seconda della sua posizione nel vettore A(..): A(1) "cifra delle unità", A(2) "cifra delle decine" e così via. Quindi, se  $A=(2,5,1,4,3)$ , avremo:

$$V=2 \cdot 10^0 + 5 \cdot 10^1 + 1 \cdot 10^2 + 4 \cdot 10^3 + 3 \cdot 10^4 = 2 + 50 + 100 + 4000 + 30000 = 34152$$

Alla 140 nella variabile S si aggiorna la somma dei vari valori per cui, alla fine, S contiene la soluzione.

```
100 n=5:dim a(n):c=0
110 for k=1 to n:a(k)=k:next
120 c=c+1:v=0
130 for k=1 to n:v=v+a(k)*10^(k-1):next
140 print v;"addendo n.";c:s=s+v
150 i=n-1
160 if a(i)>=a(i+1) then i=i-1:goto160
170 if i=0 then print:print"totale: ";s:end
180 j=n
190 if a(j)<=a(i) then j=j-1:goto190
200 t=a(j):a(j)=a(i):a(i)=t
210 i=i+1:j=n
220 if i<j then
t=a(j):a(j)=a(i):a(i)=t:i=i+1:j=j-1:goto 220
230 goto 120
```

Direi che il nostro Commodore ha senz'altro superato la verifica, voi che ne dite?





# Libdragon Development: il kit per creare giochi su Nintendo 64 - parte 1

di Takahiro Yoshioka – trad: Carlo Nithaiah Del Mar Pirazzini

Libdragon è un kit di sviluppo (SDK) che permette di realizzare giochi su Nintendo 64.

Consente di sviluppare utilizzando il linguaggio di programma C e il compilatore GCC fino al C11.

Per gli utenti inesperti, può essere molto intimidatorio e difficile da utilizzare, quindi è consigliata una conoscenza approfondita del C prima di imbarcarsi nel viaggio di sviluppo su N64 attraverso Libdragon.

In questo articolo vedremo come iniziare a preparare la nostra postazione di lavoro.

Questa guida è il minimo indispensabile per realizzare un gioco basato su Libdragon. Avremo solo un'interfaccia testuale, poco entusiasmante, ma è un trampolino di lancio necessario per creare progetti più complessi in futuro.



## Installazione di Libdragon

Esistono molti modi per iniziare a sviluppare su Pc. Una guida più completa può essere quella sulla wiki ufficiale (<https://github.com/DragonMinded/libdragon/wiki/Installing-libdragon>), ma cercherò di fornire un approccio step by step della configurazione di Libdragon mantenendo uno stile "semplice".

Dato che utilizzo Windows come piattaforma, mostrerò il processo con questo SO. Per le altre piattaforme vi invito a consultare la stessa wiki che vi ho lasciato sopra. Il processo dovrebbe però essere piuttosto simile.

## METODO DOCKER

Metodo consigliato dal team di sviluppo che consiste nell'utilizzare Docker, un software che consente di eseguire "contenitori", piccole macchine virtuali auto-chiuse. Viene utilizzato per eseguire programmi che funzionano in modo coerente su piattaforme diverse poiché l'ambiente effettivo su cui è eseguito è virtualizzato.

## PASSAGGIO: 1 Installazione Docker

Possiamo scaricare Docker da questo indirizzo <https://www.docker.com/> ed installarlo.

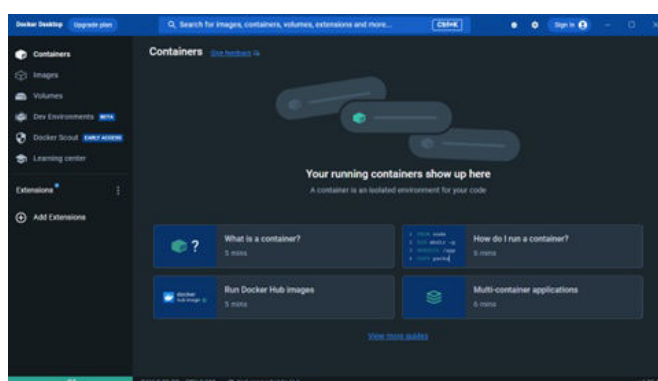
Docker può essere eseguito come applicazione da riga di comando, ma Docker Desktop può essere utile grazie ad un'interfaccia più intuitiva.

Alcune cosette da sapere:

- Su Windows, potrebbe essere necessario aggiornare il sottosistema Windows per Linux (WSL) alla versione più recente.
- Se si monta una VPN, potrebbe influenzare ed interferire con Docker.

Detto questo partiamo

La prima schermata che verrà fuori sarà questa:

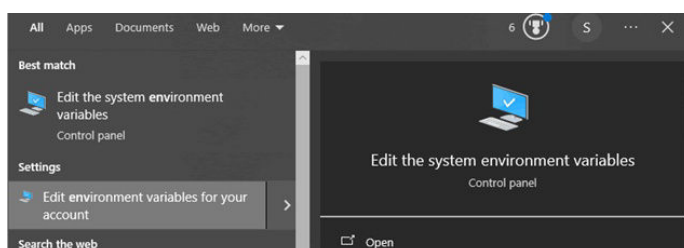


## PASSAGGIO 2: scaricamento eseguibile

L'eseguibile è disponibile da qui: <https://github.com/anacierdem/libdragon-docker/releases/tag/v10.9.1>.

Una volta lo ZIP, devi posizionarlo da qualche parte dove la variabile PATH possa trovarlo.

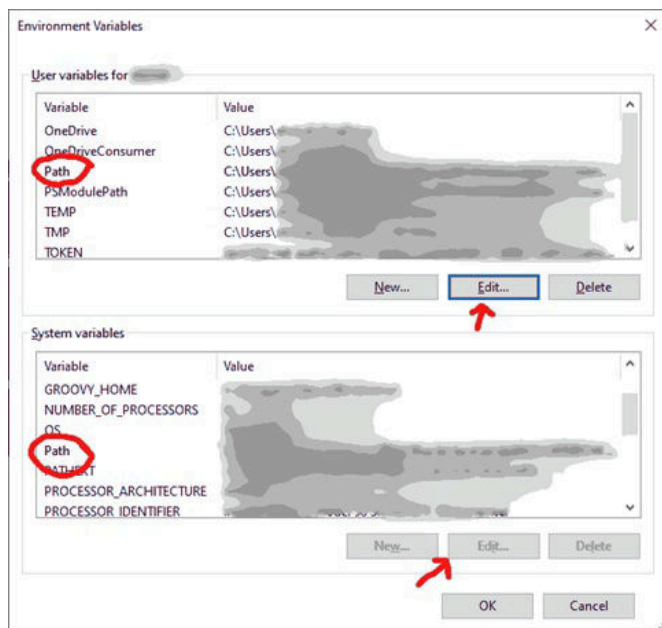
Una volta messo al sicuro modifica le variabili di ambiente in questo modo:



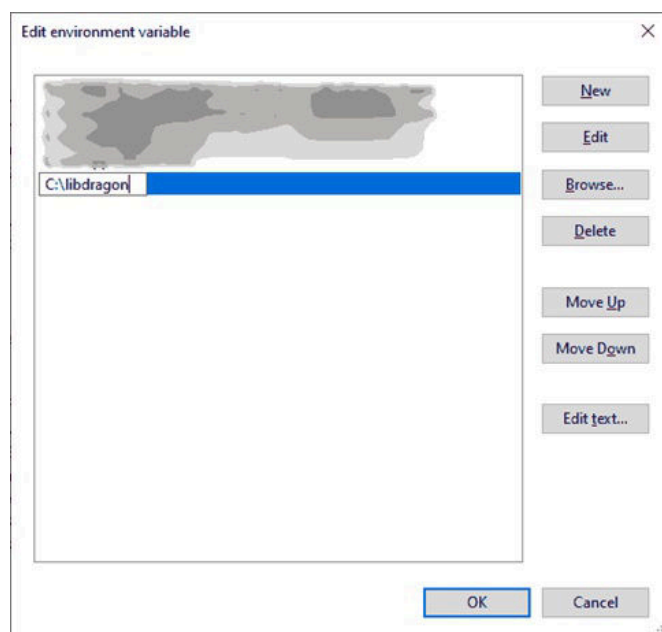




Modifica la variabile d'ambiente chiamata "path" o "PATH" per il tuo utente.



Aggiungi una nuova voce con la cartella in cui si trova l'eseguibile



### Passaggio 3: inizializzare un progetto Libdragon

Apri il tuo terminale e vai alla cartella del progetto chiamata usando il CMD in questo modo:

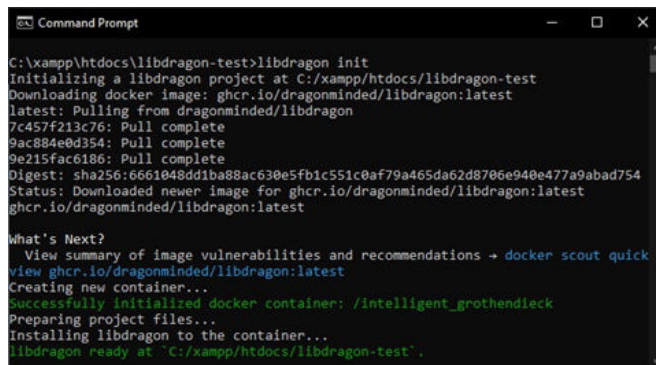
```
cd C:\percorso\della\tua\cartella
```

Quindi esegui il comando per inizializzare in questa cartella:

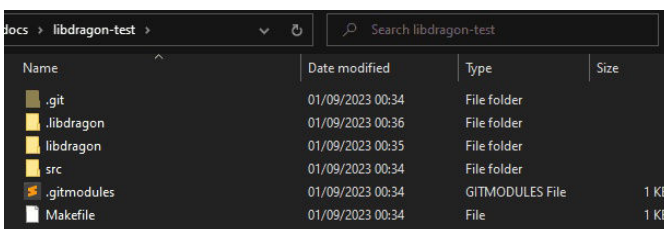
```
libdragon init
```

Ci vorranno alcuni minuti per scaricare i file necessari,

ma si otterrà qualcosa del genere:



Ora sarai pronto per iniziare a sviluppare con Libdragon!

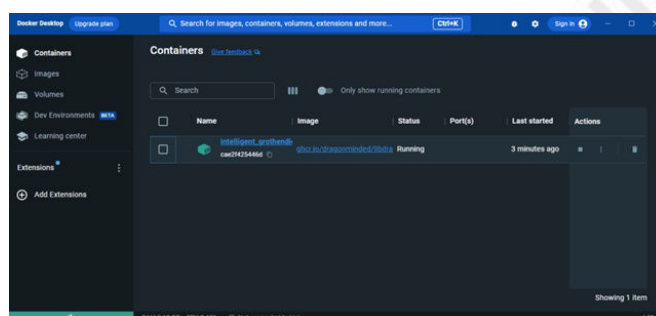


### IL TUO PROGETTO BASE

A questo punto avrai finito di configurare il Libdragon e avrai una cartella con un repository GIT che puoi utilizzare per il controllo della versione. Ci sono diverse guide su come usare GIT quindi non entrerò nei dettagli qui. Le cose principali da sapere però sono queste:

- GIT – La cartella contenente tutti i dati per il tuo repository
- .libdragon – contiene la configurazione Libdragon
- libdragon – il dispositivo contenente il sotto modulo. Fondamentalmente, tutto il codice di Libdragon.
- src – Qui è dove inserisci il codice per il tuo progetto.
- .gitmodules – Contiene informazioni sui moduli utilizzati dal progetto.
- Makefile – un Makefile di base che contiene le istruzioni per costruire la tua ROM.

Inoltre, se guardi la finestra del desktop Docker, vedrai che è impostata un'immagine Libdragon. Il nome dell'immagine viene generato in modo casuale.



Il prossimo articolo parlerà della Libdragon Console.

Daremo un'occhiata a come può essere modificata la console, manipolata e come sviluppare e debuggare.





# Quattro chiacchiere con i Knights of Bytes

di Carlo Nithaiah Del Mar Pirazzini

Ragazzi, grazie per questa intervista e grazie per i vostri fantastici giochi (Sam's Journey NdFF).

**Cominciamo con la prima domanda: come è nato Knights of Bytes.**

Chester Kollschen: Creo videogiochi dalla metà degli anni '80, quando ho acquistato il mio primo computer di casa e ho iniziato a farlo con la programmazione. È iniziato come un hobby con semplici avventure testuali e giochi d'azione di base, ma è diventata rapidamente una passione quando i progetti sono diventati più grandi, hanno coinvolto più persone e hanno dato i loro frutti risultando nettamente migliori.

A metà degli anni '90 ho deciso di trasformare la mia passione in una professione e ho fondato Knights of Bytes. Sebbene l'era dei 16 bit fosse già in pieno svolgimento, presi di mira l'universo del Commodore 64, computer con cui sono cresciuto. Il mio primo gioco Ice Guys è uscito nel 1995 e ha attirato l'attenzione dell'aspirante editore retrò Protovision. In collaborazione con loro, ho anche creato e pubblicato Bomb Mania e poi Metal Dust.

Nel 2005, Knights of Bytes ha subito un momento di pausa, quando ho iniziato a sviluppare e pubblicare giochi per dispositivi mobili telefoni con un'etichetta diversa. Tuttavia, l'ho ripreso dieci anni dopo, dopo aver osservato l'aumento interesse del pubblico per il retrogaming.

Ho chiesto a Stefan Gutsch, che ho incontrato durante il mio periodo con Protovision dove abbiamo lavorato insieme per la prima volta, di unirsi al mio team di progetto. Si è rivelata una collaborazione molto fruttuosa poiché avevamo un progetto molto simile, e io ho sempre cercato un pixel artist e lui ha sempre cercato un programmatore.

Alex Ney è salito a bordo nel 2015 mentre stavamo progettando un enorme gioco platform per il C64 e avevamo bisogno di un mago dell'audio multi-stile che era pronto per quel compito.

Quell'enorme gioco platform è diventato noto come Sam's Journey. Quando lo abbiamo pubblicato nel 2017 Ha ricevuto elogi estremamente alti sia dalla critica che dal pubblico. Questo successo ci ha ispirato a farlo abbiamo portato il gioco su un'altra piattaforma a 8 bit e alla fine abbiamo deciso per il NES.

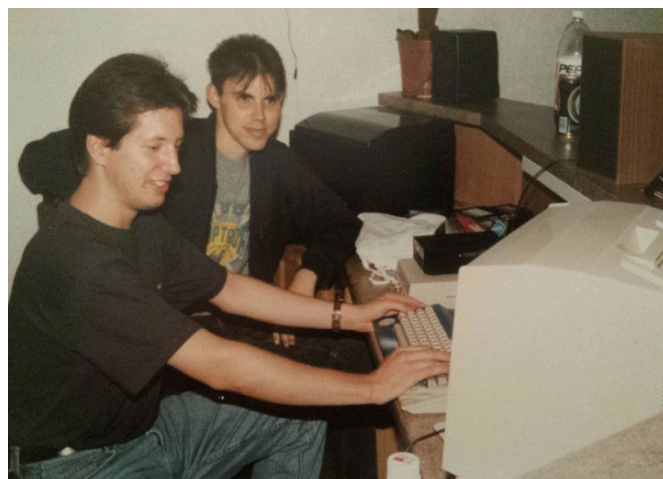
**Puoi parlarci del piccolo Sam? Come si sviluppa la storia di questo piccolo eroe paffuto e come si è sviluppato nel corso di questi anni.**

Chester Kollschen: Sono un grande fan di Super Mario Bros. 3 (NES), Kirby's Adventure (NES) e Donkey Kong Serie Country (SNES), e mi sono sempre chiesto perché giochi del genere non siano mai arrivati sul C64.

All'inizio degli anni '90, ho deciso di provarci e ho disegnato un piccolo personaggio sprite che potesse cambiarlo dall'aspetto da carino a tosto. Ma i miei tentativi di creare un gioco completo come quello sono falliti a causa della mia mancanza di competenze in quel momento. E così, il disco con quel personaggio sprite è tornato nella sua scatola per attendere tempi migliori,

Avanti veloce al 2015... Quando Stefan e io ci siamo resi conto che anche lui voleva realizzare un gioco platform per C64, abbiamo iniziato la nostra ricerca durante la quale mi sono imbattuto proprio in quel disco. A entrambi piaceva quel personaggio e fummo subito d'accordo che questo sarebbe stato il nostro eroe. Abbiamo anche colto l'idea di dargli il potere di trasformarsi in diversi alter ego. Dopo una serie di sessioni di brainstorming, abbiamo compilato un elenco di funzionalità che volevamo aver nel nostro gioco e questa volta, fortunatamente, siamo riusciti a dargli vita.

Il nostro Sam ha conquistato molti fan nella community di C64 e siamo abbastanza sicuri che questo gli abbia dato la fiducia necessaria per farlo viaggiare anche in altre piattaforme.



**Una foto dei gloriosi anni 90 con Stefan Gutsch (a destra), game designer e pixel artist**







**Chester Kollschen nel suo studio di progettazione**

#### **Qualche progetto futuro in arrivo?**

Chester Kollschen: Ci sono alcune idee e concetti che continuano a tornarci in mente ogni volta che parliamo dei prossimi progetti. È comunque un bouquet piuttosto colorato, che include un altro episodio di Sam's Journey, un gioco di corse, un grintoso action-adventure e uno sparatutto.

Per decidere un candidato, stiamo attualmente creando modelli e prototipi per vedere cosa è interessante e cosa no. Naturalmente, vi terremo aggiornati sul nostro sito

web Knights of Bytes e sui canali dei social media.

Grazie per le vostre risposte, non vediamo l'ora di vedere altri vostri titoli.

Il sito dove seguire, acquistare e scaricare i prodotti dei Knights of Bytes:

- <https://www.knightsofbytes.games/>



**Alex Ney, Compositore e artista Chiptune**







# Uno sguardo approfondito su Amiga38

Articolo originariamente pubblicato su [blog.amigaguru.com](http://blog.amigaguru.com)

di *Gianluca Girelli*

Lunga vita all'Amiga, il computer che non si può uccidere, quello che si rifiuta di morire...

Quante volte avete sentito queste parole? Quante volte vi siete chiesti se il concetto alla base di queste parole fosse reale o semplicemente illusorio? Quante volte avete desiderato di essere, almeno una volta nella vita, parte dello spettacolo, ma non ne avete mai avuto la possibilità? Tutte queste domande, e molte altre, hanno invaso la mia mente per anni fino a quando non ho avuto la possibilità di partecipare a un evento Amiga. L'anno era il 2019, il luogo era Neuss (Germania) e l'evento era il 34° anniversario dalla prima presentazione di Amiga nel 1985.

I miei lettori di vecchia data sanno come mi sia sentito per anni, a causa del fatto che la vita mi ha costretto a saltare eventi importanti della storia di Commodore e Amiga. Sanno anche che quel "vuoto" nella mia anima è stato finalmente colmato dopo la riunione del 2019. Dopo Amiga34, mi ero ripromesso di partecipare regolarmente a questi incontri, ma il COVID e questioni di vita personale (ancora una volta) hanno deciso diversamente. Avendo saltato l'Amiga37, ero così entusiasta quando, a sorpresa, è stata annunciata la presente edizione; sono riuscito a conciliare vita e lavoro, ho prenotato il volo, comprato il biglietto per l'evento, fatto le valigie e sono partito. Ecco come è andata questa volta.

Organizzare un evento del genere non è un compito facile (grazie, Markus!), soprattutto con poco preavviso, per cui quest'anno la fiera si è svolta in un solo giorno invece dei soliti due, ed esattamente come nell'edizione precedente la location scelta è stato il Kunstwerk di Monchengladbach-Wickrath. Rispetto a quello di Neuss, il complesso è molto più grande e permette di accogliere un numero maggiore di visitatori, con quasi 800 presenze. Raggiungere il luogo non è difficile, grazie alla rete di treni e autobus che collega le città vicine. Venendo dall'Italia, ho prenotato personalmente un volo per Dusseldorf (via Monaco di Baviera) e ho deciso di noleggiare un'auto, compito che invece è stato assolto da Enrico Vidale (A-Cube Systems), che è arrivato in Germania via Colonia e ci è venuto a prendere all'aeroporto di Dusseldorf. Con sé ha portato un nuovo lotto di schede madri Sam460LE insieme a una

scheda madre per il laptop PPC ormai popolata, e al relativo case. Il progetto ha ancora un po' di strada da percorrere prima di arrivare a destinazione, ma comunque, dopo anni di attesa, ora abbiamo qualcosa di molto tangibile con cui confrontarci.

Anche se l'inizio dell'evento era previsto per sabato 7, la festa era già iniziata nelle prime ore del venerdì mattina, quando Amighisti da tutto il mondo hanno iniziato a riunirsi all'aeroporto. Nonostante fossero già passati 4 anni, leggende come Markus Tillmann (capo organizzatore dell'evento), Dave Haynie (ex ingegnere Commodore) e Mike Battilana (Cloanto) mi hanno accolto come un amico di vecchia data e come se ci fossimo lasciati solo pochi giorni prima.

Dopo un pranzo veloce, ci siamo recati al Kunstwerk per

===== 12:15 =====	
10:00 10:15	Opening Speech with Trevor Dickinson
10:20 10:50	Presentation by Trevor Dickinson / A-Eon
10:55 11:20	Presentation by Michael Schultz / PiStorm
11:30 12:15	Presentation by Camilla Boemann / AmigaOS Dev Team
BREAK	
13:15 13:45	Hans Ippisch, Horst Brandl, Richard Löwenstein interviewed by Nico Barbat
13:50 14:20	Interview with Gail Wellington / Commodore (Remote)
14:25 14:55	Interview with Beth Richards & Dave Haynie / Commodore
15:00 15:30	Surprise Guest!
BREAK	
16:30 17:30	Apollo Team - Evaluation Apollo Demo Contest and Presentation of the Best Entries
17:30 18:00	Raffle and Amiga 38 Community Awards 2023
18:30	END of Amiga 38







allestire i nostri stand. Nelle immagini che seguono, potete farvi un'idea di come si presentava. Il primo giorno (venerdì 6) è stato dedicato principalmente alla preparazione degli stand e all'incontro con gli espositori, anche se i visitatori sono stati autorizzati a entrare per l'ultima parte della giornata. La serata si è conclusa con la festa di lancio e la presentazione dell'eccezionale gioco "Reshoot Proxima 3" di Richard Lowenstein e del suo team, con la musica di Martin Ashman e Siegfried Karcher e con gli interventi di Andre Kudra sulla demoscene. L'atmosfera era già al massimo, nonostante ufficialmente la fiera non fosse ancora iniziata!

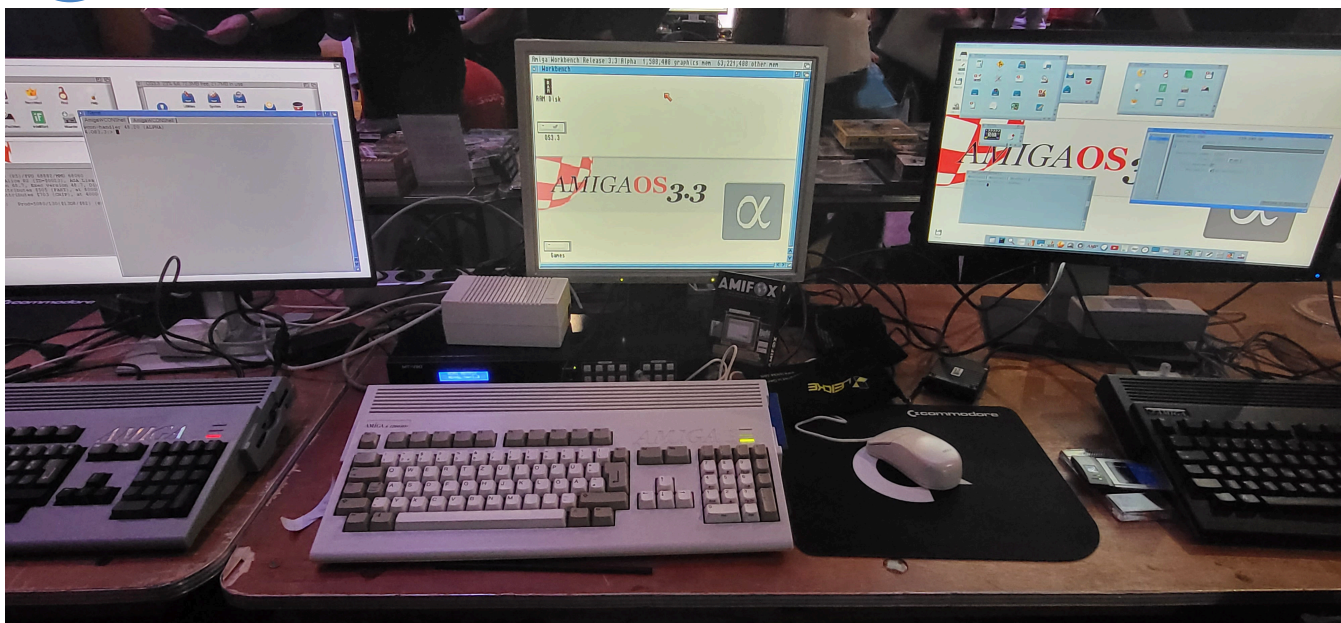
Il sabato è arrivato in tutto il suo splendore, con i visitatori in fila fuori dal centro congressi fin dalle prime ore del mattino. Era passato solo un anno e gli espositori e i visitatori erano ansiosi di sapere quanti passi avanti avesse fatto la comunità in un lasso di tempo così breve. Il discorso di apertura è stato tenuto da Trevor Dickinson,

che ha anche fatto il punto della situazione sui progetti di A-Eon. In particolare, ha condiviso alcuni dettagli su come la realizzazione della scheda madre Tabor A1222+ sia finalmente giunta al termine, grazie a un'entusiasmante collaborazione tra aziende di tecnologia Amiga: ACube Systems, AAA Technology e la stessa A-EON Technology. Gli Amiga equipaggiati con questa scheda erano finalmente completamente operativi presso lo stand A-Eon.

È stato poi il turno di Michael Schultz (PiStorm), presto seguito da Camilla Boemann. Camilla fa parte del Dev Team di AmigaOS e ha fornito interessanti informazioni sulla correzione dei bug per l'attuale versione di AmigaOS per i classici (v3.2.2) e sui miglioramenti previsti per la v3.2.3. Ha inoltre dichiarato che AOS3.3 è in parte in fase di realizzazione (prima fase alpha e solo per la versione per i classici) e che arriverà non prima di 2-4 anni da ora. Per il momento, quindi, godetevi quello che avete: è già un sogno così!







Il pomeriggio è stato dedicato a una serie di interviste: Hans Ippisch, Horst Brabdl, Richard Lowenstein, Gail Wellington, Beth Richards e Dave Haynie si sono alternati sul palco e, dopo una presentazione del team Apollo, l'evento si è concluso con la tradizionale lotteria e la premiazione.

Nel frattempo, questa volta indossando anche i panni di redattore di Retro Magazine, ho avuto la possibilità di visitare quasi ogni singolo stand e di chiacchierare con espositori, leggende di Commodore/Amiga e/o amici di vecchia data. Si tratta di: Dave Haynie, RJ Mical e David Pleasance (la presentazione non è necessaria, LOL! ), Trevor Dickinson e Matthew Leamann (A-Eon Technologies), i miei amici Enrico Vidale e Philippe Ferrucci (A-Cube Systems), HunoPPC (sviluppatore di giochi), Sebastian Eggermont (Commodore Users Europe), Oliver Graf (sviluppatore MEGA65), Camilla Boemann (AmigaOS Dev Team), Michele "Mike" Battilana (Cloanto), Daniel Mussener

(GoldenCode.de), Davide Palombo (con me, cofondatore di GDG-Entertain), Kymon e il suo gruppo di sviluppatori, mio figlio Simone. Mi scuso con tutti gli altri con cui non ho avuto modo di interagire. Non è stato intenzionale e prometto di fare molto meglio la prossima volta.

Come al solito, ci sono state troppe cose degne di nota, quindi ne riporterò solo un paio: il terzo libro della trilogia di David e Trevor sulla storia di Commodore/Amiga (From Vultures To Vampires) e un'implementazione open-source di una scheda madre Commodore 128 revision 9 da parte di Johan Grip. Questa macchina è stata il mio primo computer ed è ancora la mia piattaforma preferita, quindi sono sicuro che mi perdonerete per questa menzione d'onore. Il sito web del progetto contiene molte informazioni sulla realizzazione della scheda, la più degna di nota delle quali è, a mio modesto parere, l'integrazione di un modulatore RF sostitutivo.







Questo modulatore si inserisce interamente nell'ingombro di quello originale. Se si desidera utilizzare un modulatore RF originale o un'altra scheda sostitutiva, è possibile lasciare questi componenti non popolati, aggiungere uno strato di isolante ed installare un modulatore originale.

Tra le altre cose, anche questa volta non poteva mancare un divertente cosplay di Ravenlordess e del suo amico. I cosplay di solito non sono molto presenti nelle fiere di informatica, ma questo era speciale, riprendente una nota scena del mitico "Monkey Island".



Questa volta non ho avuto la possibilità di partecipare all'after party (festa privata solo per gli espositori che si tiene alla fine della kermesse), ma le vie di Amiga sono

infinite e così ci siamo ritrovati (beh, almeno alcuni di noi), il giorno dopo, di nuovo riuniti sotto la bandiera di Amiga mentre aspettavamo di imbarcarci sugli aerei che ci avrebbero riportato a casa.

Ancora una volta, non potrò mai ringraziare abbastanza Markus e il suo staff (Torsten Raudssus, Paul Kitching e Vitus Zeel) per aver organizzato un tale spettacolo e averci fatto sentire a casa. Il loro amore e la loro passione sono notevoli e si vedono. Ora, nonostante la tristezza per un fine settimana che mi è sembrato troppo breve, aspetto con ansia il prossimo grande evento: il 40° anniversario di Amiga! Restate sintonizzati.

#### RISORSE

<https://www.acube-systems.biz/>  
<https://www.amigaforever.com/>  
<https://www.c64forever.com/>  
[https://www.amigashop.org/product\\_info.php?products\\_id=391](https://www.amigashop.org/product_info.php?products_id=391)  
<http://a-eon.biz/>  
<https://www.facebook.com/apolloaccelerators/>  
<https://c128.se/>  
<https://twitter.com/Ravenlordess>





## Giappone 23^puntata:

### Nintendo e Microsoft: c'erano una volta dei semplici droni...

di Michele Ugolini

Cari lettori, quali notizie di retrogaming arrivano dal nostro amato Giappone?

Cosa sta succedendo nelle grandi case di creazione dei videogames?

E' tutto fermo oppure è tutto segretamente in movimento? La polvere si sta depositando sui grandi onori del passato, oppure ci sono grandi invenzioni e novità in procinto di arrivare sul mercato? Le loro grandi case videoludiche stanno riciclando importanti titoli del passato, oppure stanno creando nuove partnership strategiche per apportare migliorie su alcuni grandi titoli recenti? Ecco a voi svelate alcune notizie del passato, evolute in un incerto presente, con grandi intuizioni per il futuro: una filosofia circolare tipica della terra nipponica, come frequentemente vi ho accennato negli articoli passati!

Facciamo una breve sosta in casa big "N".

Come sappiamo la Gamescom è terminata e c'è grande fermento riguardo l'hardware montato nella Nintendo Switch 2, console ancora non ufficialmente annunciata da Kyoto, ma destinata a presentarsi come l'erede tanto attesa. Probabilmente non la vedremo sotto l'albero di Natale ma il tempo sta stringendo, forse la vedremo a metà 2024 e le sue prestazioni saranno simili a Xbox Series S e ASUS ROG Ally, rispettivamente in modalità docked e in portabilità.

Sembra essere confermato il supporto della

console al DLSS di NVIDIA in un annuncio di lavoro pubblicato dalla stessa Nintendo. Rivolta a un aspirante Data Engineer, la posizione aperta è dedicata a esperti nel campo del machine learning, desiderosi di mettere alla prova la tecnologia su "piattaforme integrate a basso consumo".

Il futuro ingegnere Nintendo, si legge nell'annuncio, avrà il compito di esplorare i processi di "ottimizzazione del machine learning", al fine di adeguare questi ultimi a "limiti in termini di memoria, potenza e CPU". Perciò a quanto pare l'architettura di questa erede sembra essere stata confermata dai grandi capi della grande "N".

Nel frattempo che verrà alla luce Super Mario Bros Wonder abbiamo anche notizie riguardo l'Xbox Game Pass: subirà un aumento di prezzo vista l'inflazione a livello planetario.

Come mai questa breve sosta in casa big "N" sta puntando il proprio sguardo verso casa Microsoft?

Phil Spencer come sappiamo è il capo della divisione Gaming della Microsoft ed oltre ad aver recentemente dichiarato tale futuro aumento, nonchè aver ribadito la crisi dei giochi targati AAA, ha anche annunciato una cosa sorprendente, un terremoto dal sapore antico e dalle visioni future.

Sebbene per ora non può ancora essere annunciato nulla ufficialmente, gli Xbox Game Studios sono al lavoro su diversi titoli in collaborazione con studi giapponesi. Immaginiamo quindi si tratti di titoli







sviluppati nella sezione Xbox Game Studios Publishing, che si occupa di produrre titoli First Party realizzati da sviluppatori terzi. Nessun titolo è stato annunciato, le previsioni di uscita sono quindi a data da destinarsi poichè probabilmente Microsoft vuole evitare di fare lo stesso errore fatto con Scalebound, alimentando tante voci per poi farle scoppiare come una grande bolla di sapone.

Non solo.

La volontà di Microsoft di espandersi tramite acquisizioni di importanti studios è notizia ormai ben chiara per tutti.

America e Giappone voglio stringere sempre più accordi. Questa visione di partnership non va vista solo dal punto di vista dei videogames ma soprattutto dal punto di vista di condivisione di conoscenza dell'elettronica di altissimo livello, quella top secret, con un velato retrogusto militare, rimanendo sempre al vertice delle produzioni vista la repentina ascesa di altri concorrenti come la Cina inerenti a qualsiasi campo, a livello mondiale.

Inoltre l'unione della conoscenza riguardo l'elettronica porterebbe un upgrade generale di tutte le medio-piccole realtà industriali che ruotano attorno all'intero processo dello sviluppo delle CPU/GPU: dalla semplice chimica per la stampa delle PCB fino alla generazione di nuovi brevetti vista la stagnazione attuale di una Taiwan che concentra gli investimenti sempre più per la propria difesa anzichè sviluppare nuovi componenti più performanti.

Speriamo che le ambizioni sovraniste di alcuni stati non compromettano ulteriormente la stabilità mondiale. Speriamo presto di poter parlare solo di progresso e benessere per tutti e magari ricordare la parola "drone" unicamente come un giocattolo dotato di telecomando che ci permetteva di osservare un pallone di cuoio incastrato sul tetto di casa.

Ritornando a Microsoft... In un recente scambio di e-mail, che è venuto alla luce come parte della proposta dell'acquisizione di Activision-Blizzard, Phil Spencer avrebbe scritto ad un collega, dicendo che l'acquisizione di Nintendo rappresenterebbe un "momento che definisce una carriera" per lui. Pare dunque che Spencer abbia corteggiato Nintendo, ma che la grande "N" abbia (apparentemente nonchè momentaneamente) rifiutato di essere comprata dal colosso americano.

Ecco una traduzione di quanto scritto nell'e-mail emersa :

"Takeshi, sono totalmente d'accordo sul fatto che Nintendo sia L'ASSET principale per il Gaming e ad oggi il Gaming è il sentiero più probabile da battere per rilevanza verso i consumatori. Ho avuto numerose conversazioni con il team a capo di Nintendo riguardo ad una collaborazione più stretta e sento che se un'azienda degli Stati Uniti avesse qualche possibilità con Nintendo, noi siamo probabilmente nella posizione migliore. La sfortunata situazione (o fortunata per Nintendo) è che Nintendo è seduta su una montagna di denaro, e che hanno un consiglio di amministrazione che non ha spinto per ulteriori incrementi nella crescita di mercato o nell'aumento del valore delle proprie azioni, se non in tempi più recenti. Dico "in tempi più recenti" in quanto il nostro ex-Value Act del consiglio di amministrazione Microsoft ha acquisito pesanti quote di Nintendo e mi sono mantenuto in contatto con Mason Morfit mentre procedeva con questa acquisizione.

È probabile che continuerà a spingere per ottenere più quote Nintendo, cosa che potrebbe creare delle opportunità per noi. Senza questo catalizzatore non vedo una possibilità per una acquisizione a breve termine di Nintendo in cui entrambe le parti siano concordi, e Microsoft ed io non crediamo che un'azione aggressiva sarebbe quella indicata, perciò stiamo giocando sul lungo termine. Tuttavia il nostro consiglio di amministrazione ha visionato l'intero rapporto scritto su Nintendo (e Valve) ed è totalmente a supporto su ambedue se si presenterà l'opportunità, così come lo sono io.

Mi piace questa discussione ed apprezzo che tu ci veda delle opportunità. Ad un certo punto, acquisire Nintendo rappresenterebbe un momento che definirebbe una carriera e credo fermamente che sarebbe un'ottima mossa per entrambe le aziende. Nintendo ci sta soltanto mettendo tantissimo tempo a vedere che il loro futuro esiste al di fuori del loro hardware."

Per ora concludiamo qui ragazzi, come potete ben vedere in Giappone il passato, il presente e soprattutto il futuro coincidono sempre come un movimento spiraliforme. Mai sottovalutare neppure un'ansa di questa spirale: potrebbe nascondere grandi sorprese maturate dall'esperienza nel non commettere nuovamente i medesimi errori del proprio passato.

Alla prossima lettura cari amici!





# GP Cars – Commodore 64

*di Mic the Biker Novarina*

**Release Date:** 19 Marzo 2023

**Released by:** LHS

Il gioco in questione è uscito sul mercato nel mese di Marzo del 2023, ad opera della LHS. Il team che ha curato il comparto tecnico di GP Cars è di assoluto livello: LSH al codice; Almighty God alla grafica, coadiuvato da Nico Galis, e NecroPolo che si è occupato del sonoro.

Prima della sua uscita ufficiale il titolo fece già parlare di sé, visto che si è classificato 2° al WILD Demo Competition FOReEVER 2023 – ROBOT. Si tratta di un gioco di guida con vista dall'alto, un genere che nella golden age andava decisamente di moda. Basti ricordare capolavori senza tempo come Micro Machines e Supercars sul biscottone, e Indy Heat su Amiga. Ma fin dalle prime schermate grafiche di presentazione ho sussultato, perché ho avuto un flashback chiamato "Hot Rod", gioco di corse targato Sega che non tutti ricorderanno. Iniziando a smanettare le prime partite ho avuto la conferma che questo GP Cars pesca a piene mani dall'arcade Sega, uscito nel 1988 nelle sale giochi, e convertito nel 1990 per le principali piattaforme domestiche.

Siamo fortunati in fin dei conti, perché oggi abbiamo accesso a giochi notevoli, come questo, a costo zero. Un

gioco che all'epoca sarebbe costato svariate mila lire, o un deca in versione pirata. Facciamo Load e poco dopo siamo accolti dallo sgasare del motore riprodotto con efficacia dalle onde sonore del SID. La schermata ci regala una bella grafica colorata, con una muscle car azzurra inseguita da quella che pare una Porsche rossa.

Eccoci arrivare alla schermata principale, dove vedremo le cinque ambientazioni in cui potremo cimentarci. Ogni location ha 5 piste, per un totale di ben 25 gare disponibili. Not so easy, perché bisogna vincere le gare della prima zona per accedere alla successiva e via scorrendo. Partiamo dal primo slot di gare, e veniamo accolti da un conto alla rovescia 3,2,1, GO con un buon vocale digitalizzato. Le gare possono essere contro il cronometro o contro altri veicoli, con un totale di quattro. Inizialmente le madonne volano come droni nel cielo, visto che i bordi che delimitano la strada sono tipo dei muri in cemento armato e ci rallentano al minimo contatto. La pista non è per niente larga e le curve spesso sono strette. Tradotto in soldoni i primi dieci minuti li ho passati a battere da tutte le parti prima di capire che nelle curve è saggio regolare la velocità. Terminato dignitosamente un giro iniziano a sbloccarsi gli altri scenari, nei quali ci si trova spesso a fare ricche sportellate con gli avversari. Resto spiazzato dalla collisione tra gli sprite inesistente: le macchine possono passare l'una sopra l'altra senza problemi. Probabilmente se alla difficoltà di non colpire i limiti della strada avessero unito la collisione tra sprite il gioco sarebbe stato ingiocabile.

Le auto guidate dalla CPU inizialmente se ne vanno letteralmente via, ma col passare delle partite il feeling migliora e me la gioco curva dopo curva. Ci vuole comunque un po di tempo per padroneggiare l'inerzia dei comandi,







con la vettura che aumenta la velocità di sterzata tenendo la direzione. Particolare attenzione si deve fare ai cantieri sparsi per le strade: urtarne uno danneggia pesantemente il nostro mezzo a quattro ruote, molto più velocemente che colpendo solo i guard rail assassini. Qua e là si può raccogliere della preziosa Nitro che ci dà quel tot di grinta in più. Procedendo nel gioco potremo scegliere diversi tipi di auto e cambiarne il colore. Un orpello puramente estetico, perché le prestazioni tra le varie carrette presenti nel gioco sono sempre uguali.

### Grafica e Sonoro

Il comparto grafico spacca, non c'è che dire. La palette del Commodore 64 è sfruttata a dovere e le scelte cromatiche sono sempre ottime. Abbiamo anche qualche animazione nel fondale, una vera chicca per gli occhi. Gli sprite sono dettagliati e ben colorati: ogni tipologia di auto è ben riconoscibile e caratterizzata ad hoc. Lo scrolling è al top, fluido in ogni direzione, preciso e veloce.

Il comparto audio è da TOP SID. A partire dagli effetti, come citato ad inizio articolo, a finire con i molteplici brani in game, l'audio è uno spettacolo. Le canzoni danno la giusta dose di carica, sono ben composte e "suonano bene". Gli effetti sonori in game sono giusto dove servono: sgommate e urti. In questo modo non si sprecano risorse e canali del SID in maniera evidente.

### Riflessioni del Biker

Mi ha incuriosito molto questo GP Cars, e appena ho avuto qualche minuto di tempo lo ho scaricato e ho iniziato a

giocarci. I pregi sono molti, nonostante la difficoltà non indifferente. Grafica e sonoro ci fanno emozionare, e presa la mano con la dinamica dei comandi ci si diverte alla grande. Da appassionato di questo genere ho però qualche rammarico: con poco di più questo GP Cars sarebbe stato fenomenale. Mi sarebbe piaciuto ad esempio scegliere un pilota all'inizio e magari vedere un podio con premiazione alla fine delle gare. Sarebbe stato uno spettacolo una zona officina dove riparare l'auto e magari potenziarla nella tenuta di strada e nelle prestazioni.

Certo, sono tutte cose già viste in altri giochi ma proprio perché li ho giocati tutti noto che dettagli come questi lo avrebbero portato ad un altro livello. Ho adorato, e adoro tutt'ora, i buoni vecchi arcade con vista top o iso-top, e GP Cars per me, per dirlo "alla Zzap!", è decisamente un "gioco caldo".

### GIUDIZIO FINALE

**GIOCABILITA':** 85% Inizialmente ostico, una volta presa la mano ci si inizia a divertire seriamente!

**LONGEVITA':** 80% Le sfide non mancano, le piste nemmeno perciò la sfida è assicurata. Probabile che se non siete davvero appassionati del genere possa arrivare la frustrazione da "Non posso salvare", ma quando si giocava a Super Sprint era o no la stessa cosa?





# RMW Consiglia: Letture Videoludiche

di Giuseppe Rinella

Non so voi ma io con l'arrivo dell'autunno e, ancora di più, del successivo inverno, passo gran parte del tempo libero tra le mura domestiche. Insomma il clima fuori non invita certo a passare ore all'aperto.

Tra le attività che amo svolgere in casa c'è ovviamente l'abuso di videogiochi retro e non (famiglia permettendo, che viene sempre prima, è ovvio), inoltre divento un divoratore di libri per cui molto spesso metto da parte il joystick o il joypad e mi immergo nelle pagine scritte.

Da sempre sono estremamente interessato a tutto ciò che riguarda la storia del nostro passatempo preferito e dunque sono sempre alla ricerca di libri che trattino l'argomento.

Ho pensato quindi di suggerire una serie di libri a mio modesto parere da leggere assolutamente, come vedremo in alcuni casi si tratta di oggetti davvero meravigliosi che vi ritroverete ogni tanto a sfogliare, magari con l'occhio lucido.

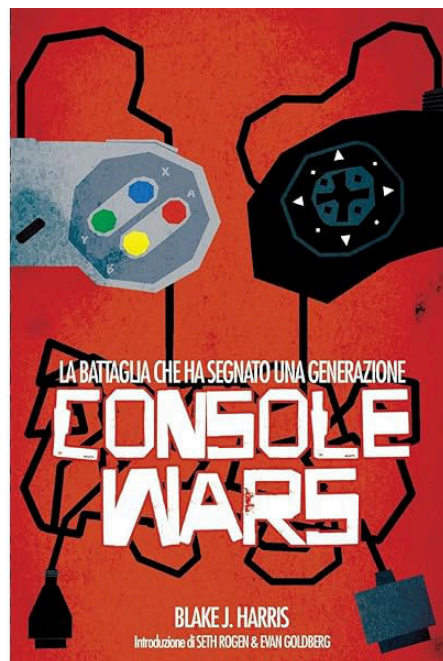
Piccola precisazione: qui non si fa pubblicità ad autori o editori, non riceviamo alcuna sponsorizzazione, quindi se notate un certo entusiasmo da parte mia non è per invogliarvi all'acquisto, si tratta di semplice gioia!

Andiamo allora!

## CONSOLE WARS (ITA-ENG)

La prima vera, epica "console war" fu, come ben sappiamo, quella che vide scontrarsi Sega e Nintendo con l'arrivo della generazione di console a 16 bit. Fino ad allora il mercato fu sostanzialmente monopolizzato da una sola azienda, con altre che inseguivano senza avere grandi speranze di stare al passo. Prima ci fu Atari e poi, dopo la crisi del 1983 che quasi spazzò via il settore videoludico, Nintendo che per anni fu sinonimo di videogiochi casalinghi con il suo meraviglioso FamiCom/NES.

Nel 1990 Hayao Nakayama, allora presidente di Sega, decise di sfidare apertamente Nintendo, che dominò l'epoca a 8 bit. Per farlo andò fisicamente a prelevare Tom Kalinske in vacanza con la famiglia per proporgli di diventare CEO della divisione americana di Sega, per portare a termine quella che sembrava allora una missione impossibile. Tom Kalinske vantava una brillante carriera



alla Mattel e fino ad allora mai aveva avuto a che fare con i videogiochi.

Kalinske accettò, il Megadrive era già in commercio in Giappone e di lì a poco sarebbe arrivato negli USA e in Europa. A lui si devono campagne di marketing a dir poco aggressive (celebre lo slogan "Genesis does what Nintendo'n't") che puntavano decisamente su un pubblico più cresciuto rispetto a quello che era il target di Nintendo. Questa intuizione si rivelò decisamente azzeccata. Prezzi a dir poco aggressivi, una mascotte come Sonic e il parco titoli del Megadrive fecero il resto e il tutto portò, almeno per un periodo, Sega a giocarsela ad armi pari contro Nintendo, arrivando addirittura a superarla in termine di vendite, anche se non durò molto.

Console Wars racconta tutto questo e lo fa alla grande, arricchendo il tutto con testimonianze e aneddoti, il tutto raccontato con un ritmo che tiene incollato il lettore.

Si potrebbe obiettare che l'autore, Blake J. Harris, sia un po' di parte simpatizzando palesemente per Sega e io mi trovo totalmente d'accordo con questa posizione. Lo scopo del libro non è quello di stabilire chi sia il migliore in questa battaglia, ma raccontare una storia che ricorda un po' il classico Davide contro Golia, l'idea quasi folle di Sega di sfidare il colosso Nintendo contro cui nessuno poteva nulla.

Finita l'era a 16 bit la vittoria fu indubbiamente di Nintendo







da qualunque punto di vista, il Super Nes si dimostrò infatti superiore su tutti i fronti nei confronti del Megadrive. Sega, dal 1993 in poi, iniziò a perdere terreno e il lancio di prodotti discutibili o non adeguatamente supportati da Sega stessa (Sega Mega CD e di lì a poco il Sega Saturn) portò all'abbandono di Kalinske e Sega iniziò il suo non troppo lento declino sul fronte gioco casalingo, almeno come produttore di hardware.

Console Wars è un libro bello ed entusiasmante, una lezione di marketing che fa della scorrettezza e aggressività la sua cifra stilistica, forse non più attuabile ai giorni nostri ma estremamente interessante, recuperatelo assolutamente!

### COMMODORE AMIGA: A VISUAL COMPENDIUM (ENG)

Se già non sapete chi siano, segnatevi assolutamente "Bitmap Books", editore che sforna libri di qualità davvero eccelsa.

Non è l'unico loro libro di cui parlerò, ma da vero adoratore di tutto ciò che è Amiga non potevo non cominciare da questo. Il libro ripercorre la storia di Amiga proponendo una carrellata dei titoli più significativi usciti per la mai troppo amata macchina Commodore.

Ad ogni gioco sono dedicate due pagine che vengono riempite da una grande immagine che lo raffigura, accompagnate da citazioni di sviluppatori o altri, riguardanti il titolo in questione. Vi sono inoltre approfondimenti di vario genere: le storiche software house che ci hanno regalato titoli epici, interviste, curiosità (perché nella storica immagine della mano che tiene il floppy viene usata la sinistra? Da dove nasce "Guru meditation"?), una corposa sezione dedicata alla demo scene, i titoli mai pubblicati e tanto altro.

Ogni pagina è un trionfo di pixel, sfogliare questo libro per chi, come me, con Amiga è letteralmente cresciuto, è come sfogliare un album di vecchie foto che mi fanno ricordare le emozioni che quei titoli hanno saputo regalarmi. La qualità della stampa, della rilegatura, la carta utilizzata,

tutto è davvero di altissimo livello, il libro è realizzato da appassionati per appassionati e la cura e l'amore riposti sono ben visibili ad ogni pagina.

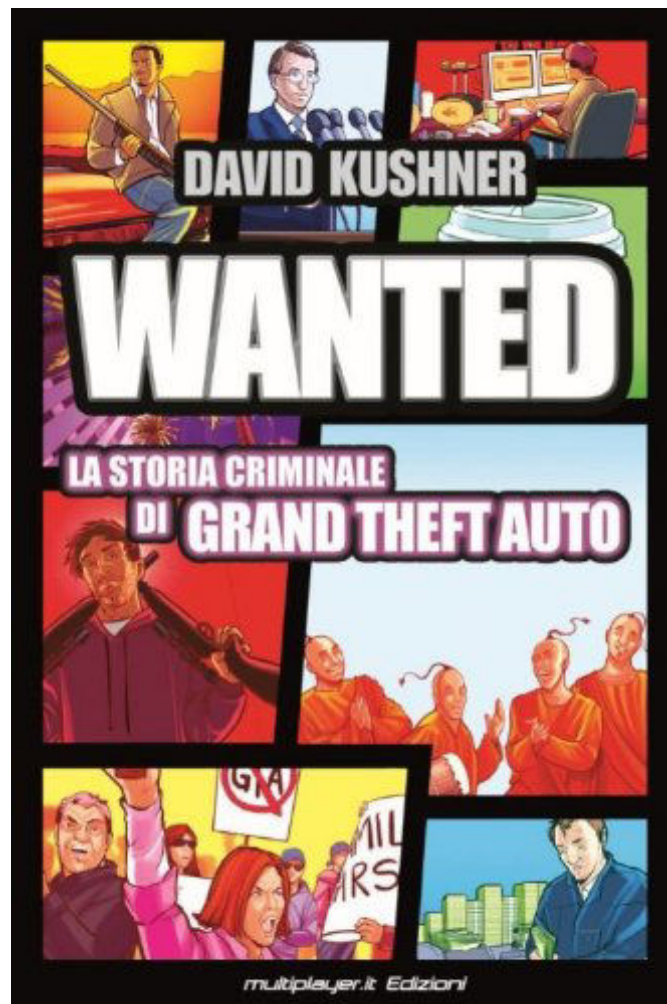
Potrei fare giusto due appunti (dettati dal puro gusto personale): il primo, per alcuni giochi forse non sono state scelte le immagini più significative. Secondo, tra i titoli manca Shadow Fighter, sacrilegio! Niente di grave comunque. Esistono altri libri della stessa serie dedicati ad altre piattaforme: Commodore 64, Sega Master System, ZX Spectrum, NES, SNES e Atari 2600/7800. Qualunque siano le vostre preferite (pian piano, tenendo un occhio sul bilancio familiare, li prenderò tutti!), acquistate ad occhi chiusi, avrete così tra le mani un prodotto di qualità eccelsa che vi ritroverete a sfogliare più e più volte, garantito.

### WANTED: La storia criminale di Grand Theft Auto (ITA-ENG)

Se come me avete vissuto appieno la nascita del fenomeno Grand Theft Auto nel lontano 1997 e da lì in poi non avete perso un capitolo della saga, questo libro fa decisamente per voi.

Wanted racconta la nascita di GTA e la conseguente crescita travolgente di Rockstar Games, partendo dal primo episodio fino al lancio di GTA IV.

Grazie alle testimonianze di chi ha vissuto quegli anni in





prima persona e un gran lavoro di ricerca dell'autore David Kushner (che ritroveremo dopo), è possibile rivivere la folle storia di una serie che ha segnato un prima e un dopo nella storia dei videogiochi, frantumando ogni record di vendite e diventando un vero fenomeno di costume.

Leggendo questo volume diventa chiaro come il nome "Rockstar" non sia affatto casuale, lo studio fondato da Sam Houser era pieno zeppo di genio e ancora più sregolatezza. Ritmi di lavoro forsennati, campagne pubblicitarie a dir poco controverse, battaglie contro politici e associazioni di genitori che tentarono in tutti i modi di ostacolare l'irresistibile successo di GTA, il tutto romanizzato quanto basta per non cadere nella fredda cronaca dei fatti, che rischierebbe di annoiare.

Insomma c'è tutto ciò che serve per tenere incollato il lettore, compreso il racconto del famoso scandalo "Hot coffee".

Wanted è una lettura davvero godibile ed entusiasmante, se avete amato e amate la serie GTA è assolutamente imprescindibile, ma anche in caso contrario rimane una lettura estremamente interessante e avvincente per chiunque voglia approfondire un pezzo fondamentale della storia dei videogiochi.

### VIDEO-GIOCHI: STAGE-ONE (ITA)

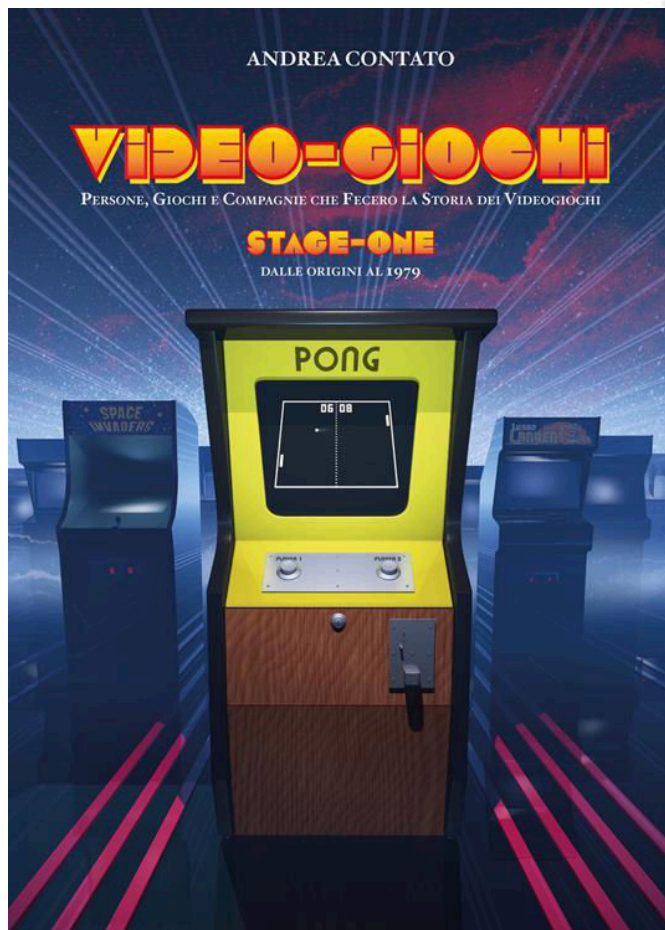
Se siete arrivati fin qui l'argomento videogiochi evidentemente vi interessa e "VIDEO-GIOCHI" è quindi un volume assolutamente irrinunciabile.

Il libro ripercorre la storia dei videogiochi dagli albori fino al 1979, ma questo è solo lo "STAGE-ONE", il secondo volume (1980-1984) è già disponibile mentre il terzo (1985-1989) uscirà alla fine di quest'anno. La serie sarà composta da cinque volumi totali, arrivando a coprire il periodo 1972-1999.

Il lavoro fatto da Andrea Contato, autore che vi invito a seguire, è davvero impressionante. La ricerca approfondita e meticolosa è evidente, così come lo sono la dedizione e la passione dell'autore. Il libro è ricco di citazioni e interviste dei personaggi che hanno contribuito alla nascita dei videogiochi, autentici geni visionari dell'informatica.

Si parte dal 1972 come detto, anno della nascita di PONG e tutto ciò che venne dopo, dal primo gioco online alle avventure testuali, da Space Invaders ad Asteroids e molto altro, con un lavoro di documentazione incredibile.

Completa il tutto l'appendice finale, con le schede dei sistemi casalinghi nati nel periodo trattato in questo volume. Pong, Odyssey, Intellivision, Apple II, Commodore



PET e altri ancora, non manca davvero nulla.

Ho scritto di "STAGE-ONE" perché trattandosi del primo volume della serie sarebbe logico iniziare da qui, il mio giudizio totalmente positivo è esattamente lo stesso per il secondo volume e, ne sono certo, per quelli che usciranno in futuro.

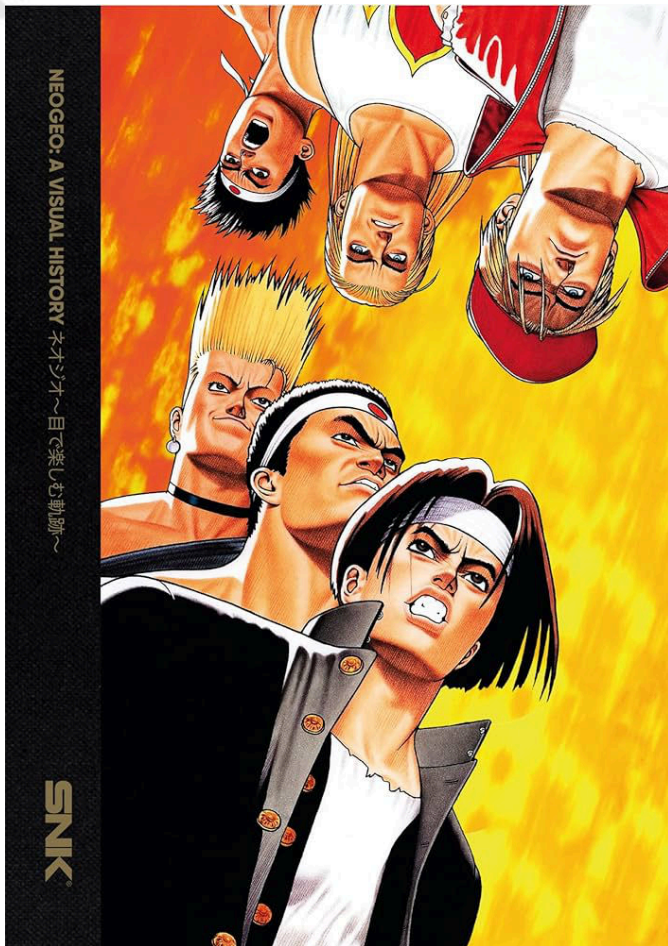
Come detto qualche riga più su, VIDEO-GIOCHI è un libro che non dovrebbe mancare sui vostri scaffali, trasuda amore e passione per la storia dei videogiochi da ogni pagina, il lavoro di Andrea Contato (con cui ho avuto il piacere di fare due chiacchiere in più di un'occasione, se frequentate fiere ed eventi legati al retrogaming e retro-computer è molto probabile che lo incontriate con il suo banchetto, andate a trovarlo, troverete una persona affabile e molto disponibile) è quasi commovente, vi consiglio caldamente di non farvelo sfuggire e lo stesso vale per i volumi successivi, io non vedo l'ora che esca il terzo!

### NEOGEO: A VISUAL HISTORY (ENG)

Chi come me è cresciuto a pane e videogiochi negli anni 80/90, può facilmente capire quanto e perché NEOGEO costituisca il massimo possibile in ambito videoludico. Dalle mie parti era sinonimo di giochi altamente spettacolari a cui potersi avvicinare però solo in sala giochi. Noi tutti sapevamo dell'esistenza della console NEOGEO casalinga, ma si trattava per noi di qualcosa di mitologico e







irraggiungibile, tanto da farci dubitare della sua reale esistenza, mentre i listini prezzi che comparivano nelle riviste di settore ci dicevano che invece sì, esisteva eccome e ci era possibile ammirarla giusto in foto.

Questo bel libro, pubblicato anche lui da Bitmap Books su licenza ufficiale SNK, ripercorre la storia di questo fantastico sistema e lo fa principalmente per immagini, come è facilmente intuibile dal titolo. Non solo quelle però, viene raccontata la storia del sistema prodotto da SNK corredata da interviste a sviluppatori e collezionisti e diverse curiosità, ma indubbiamente sono le immagini a rubare la scena e le dimensioni notevoli del volume (30X22) gli rendono onore.

Si parte con una serie di foto bellissime a tutta pagina che ritraggono tutto l'hardware prodotto dalla gloriosa azienda nipponica, è tutto un trionfo di cabinati, console, controller e altro ancora.

Si passa poi alle Box Art e successivamente ai bozzetti e studi di personaggi di alcuni dei giochi più rappresentativi, tra cui Art of Fighting, Fatal Fury, Samurai Showdown e Metal Slug.

È la parte successiva quella che probabilmente scatena gli entusiasmi di chi ha sempre ammirato i titoli NEOGEO, il mio sicuramente, quella dedicata alla Pixel Art che personalmente considero una forma d'arte e che adoro (chi non lo fa?). Un susseguirsi di enormi immagini che

permettono di apprezzare ogni singolo pixel di ogni singolo sprite, scenario e cut scene, si tratta di qualcosa che sfiora la pornografia.

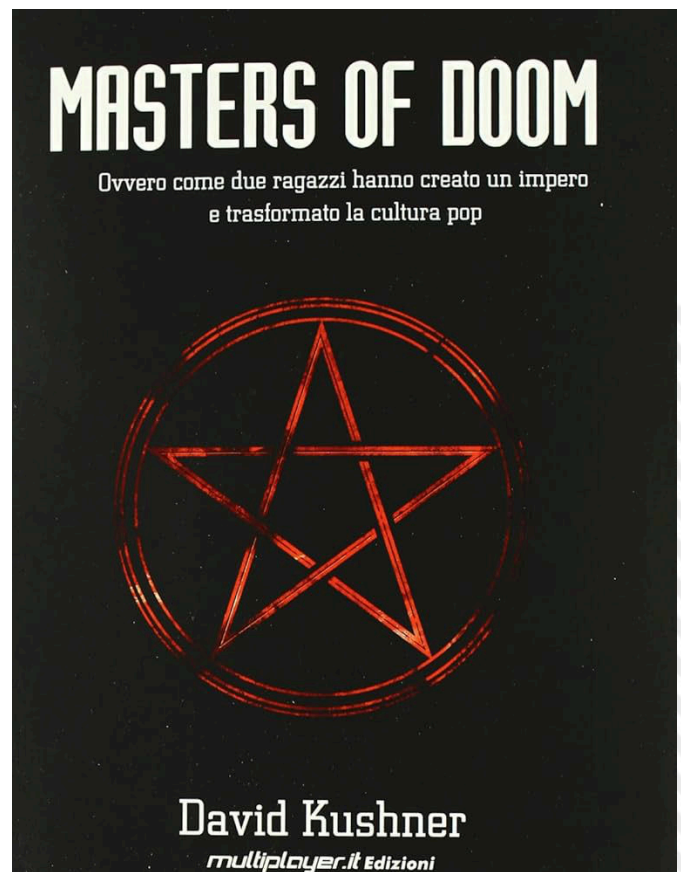
NEOGEO: A VISUAL HISTORY è un gran bell'oggetto, la copertina rigida e la carta di pregio gli conferiscono l'importanza che merita. Non è certo il libro da portarsi in metropolitana, le dimensioni e il peso di circa due chili non lo rende particolarmente comodo da trasportare ma fidatevi, se siete vissuti nel mito di NEOGEO come il sottoscritto, questo bel volume vi renderà davvero felici.

### MASTERS OF DOOM (ITA-ENG)

Ritroviamo David Kushner (questo volume è precedente a Wanted in realtà, quindi è lì che lo ritroviamo, ma cambia poco) e anche qui l'autore racconta la nascita e la crescita vertiginosa di una delle software house che ha fatto la storia dei videogiochi, ID Software, e del suo titolo dopo il quale nulla sarebbe stato più lo stesso: Doom.

Masters of Doom ripercorre l'incredibile storia dei due fondatori di ID, John Carmack e John Romero, autentici geni con personalità profondamente diverse tra loro e forse per questo, almeno fino a un certo punto di questa storia, complementari. Talmente diverse da essere definiti, nei capitoli iniziali, "La rockstar" (Romero) e "Lo scienziato" (Carmack).

Leggeremo le imprese di due autentici ribelli rivoluzionari (almeno in ambito videoludico), a partire dalla pubblicazione





di Commander Keen (dopo che la conversione di Super Mario Bros 3 che realizzarono per PC fu rifiutata da Nintendo) che sfruttò la rivoluzionaria tecnologia denominata "Adaptive tile refresh" creata da Carmack e che rese possibile lo scrolling orizzontale su PC, qualcosa di mai visto fino ad allora. E poi Wolfenstein 3D dove era già possibile intravedere quello che sarebbe arrivato dopo, fino ad arrivare a Doom che sconvolse il mondo videoludico e non solo, diventando un autentico fenomeno pop e successivamente Quake.

È possibile assaporare, grazie a questo volume, l'atmosfera di esplosiva creatività presente in ID Software, un gruppo di veri nerd geniali, e delle notti passate a creare qualcosa di unico e mai visto prima.

Il successo clamoroso ottenuto da Doom ed episodi successivi, la libertà totale concessa ai fan che iniziarono a creare editor di livelli e nuove versioni di Doom (tra cui una a tema Star Wars) tutt'altro che ostacolate dai due John. Leggendo Masters of Doom non solo ne saprete di più della nascita di una vera e propria pietra miliare videoludica, leggerete anche di qualcosa di unico e probabilmente irripetibile nella storia del nostro medium preferito.

Un libro da leggere tutto d'un fiato, divertente e appassionante, scritto ottimamente da David Kushner con un ritmo travolgente.

Unico problema per l'edizione in italiano: trovare questo libro nuovo è impossibile, dovrete rivolgervi al mercato dell'usato come feci io qualche anno fa, altrimenti c'è sempre la versione in inglese di più facile reperibilità.

Recuperatelo assolutamente se ancora non l'avete fatto, pochi libri che raccontano un pezzo di storia dei videogiochi come questo mi hanno entusiasmato così.

### THE ART OF POINT AND CLICK (ENG)

Riecco Bitmap Book con un altro gran bel libro, grande in tutti i sensi, trattandosi anche in questo caso di un volume di dimensioni e peso importanti, che non fa certo della portabilità il suo punto di forza.

A differenza di altri libri Bitmap Books, come la serie "Compendium" o volumi come "NEOGEO: A VISUAL HISTORY", ciò che viene approfondito non è la storia di una singola macchina, di un brand o una serie di videogiochi (come ad esempio avviene nel libro "Metal Slug: The ultimate History", edito anche lui da Bitmap Books), "The art of point and click" ripercorre l'evoluzione di un genere, dalla nascita fino ai giorni nostri, cioè quello delle avventure grafiche.

Il libro ripercorre in ordine cronologico tutta la storia del

genere a partire dalle prime avventure testuali, passando alle vere avventure "punta e clicca" e arrivando a giochi più recenti.

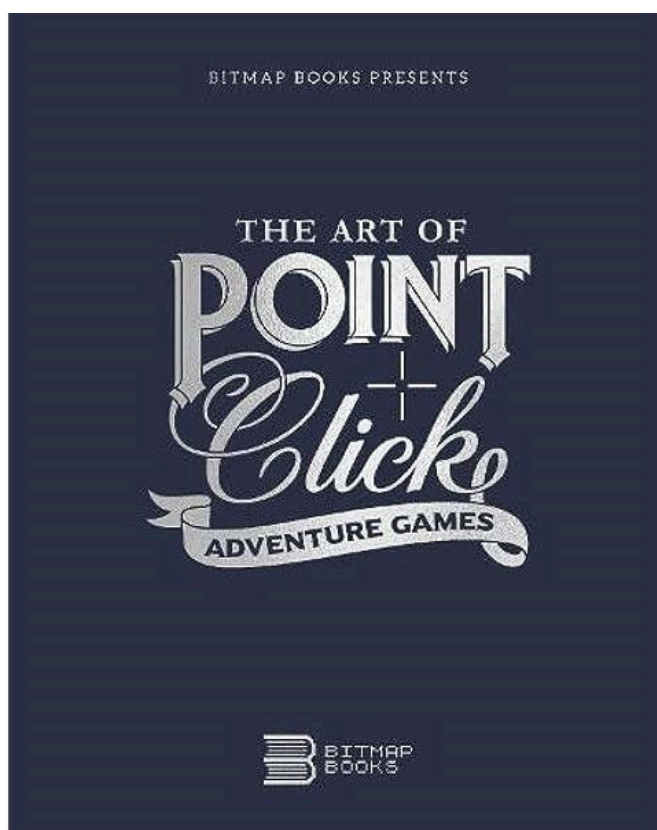
Così come gli altri libri di Bitmap Books, viene dato grandissimo spazio alle immagini, anche qui ritroviamo un tripudio di pixel art da brividi che beneficia del grande formato di questo massiccio volume.

Non solo immagini però, "The art of point and click" contiene anche più di quaranta interviste a chi sui capolavori qui illustrati ci ha lavorato, nomi del calibro di Al Lowe, Ron Gilbert e Tim Schafer giusto per citarne alcuni tra i più famosi. Le interviste sono davvero interessanti e ricche di curiosità, ad esempio non sapevo che a Ridley Scott non è fregato mai nulla del gioco dedicato a Blade Runner. "The art of point and click" è un libro assolutamente fondamentale per chi ha sempre amato questo genere di giochi, ma è estremamente interessante per chiunque, oltre che una gioia per gli occhi.

Come ci si può aspettare da un prodotto Bitmap Books, anche in questo caso la qualità del libro inteso come oggetto è pari a quella dei contenuti, carta e rilegatura sono di gran livello e la copertina rigida gli dona la giusta eleganza.

Piccola nota: l'edizione in mio possesso, la prima, arriva fino al 2017, leggo però che il volume è arrivato alla terza edizione che include 28 pagine aggiuntive, con nuove interviste e giochi non inclusi nelle edizioni precedenti.

Un libro da avere assolutamente, anche per chi come me







non è mai stato molto avvezzo al genere per manifesta incapacità (ad esclusione dei primi due Monkey Island, Simon the Sorcerer e poco altro) ma ammetto che sfogliare questa meraviglia mi fa venire una gran voglia di ripescare capolavori e gemme nascoste che colpevolmente mi sono sfuggite in passato.

### THE BITMAP BROTHERS UNIVERSE (ENG)

Un altro pezzo di storia videoludica, perché i Bitmap Brothers la storia l'hanno fatta davvero.

Chiunque abbia vissuto l'epoca dei 16 bit come il sottoscritto, ancora più se con una Amiga come il sottoscritto, non può non amarli e questo bel volume ne racconta le eroiche gesta.

In un'epoca in cui gli sviluppatori erano entità misteriose senza volto e in cui i publisher dettavano tempi e condizioni, i Bitmap Brothers fondati da Steve Kerry, Eric Matthews e Mike Montgomery furono un'autentica rivoluzione, mettendo al centro di tutto gli sviluppatori. A loro modo storiche le foto, pubblicate dalle riviste di settore, in cui i tre fondatori facevano mostra di sé, con pose e look che presto gli fecero guadagnare l'appellativo di vere "rockstar", qualcosa di mai visto prima per dei semplici programmatori e che mise in chiaro quanto la missione del trio inglese fu quella di far sì che i giochi li creava davvero ottenesse la giusta visibilità.

Non solo forma ma tanta sostanza, i giochi prodotti dai Bitmap furono sempre di qualità altissima, tecnicamente magnifici, con un'estetica metallica che li rendeva subito riconoscibili e una cura per l'aspetto sonoro eccezionale.

The Bitmap Brothers Universe ripercorre la storia della gloriosa software house, successivamente anche publisher con la fondazione di Renegade, raccontando la genesi di ognuno dei titoli realizzati.

C'è tanto da leggere ma non solo, completano l'opera bozzetti, studi di personaggi (alcuni mai utilizzati) e ambienti, una ricca galleria di immagini a colori degli storici titoli Bitmap Brothers (a cui è stato applicato un filtro che simula le TV CRT, meravigliose) e altro ancora, come i bozzetti di Speedball 3 che purtroppo non ha mai visto la luce.

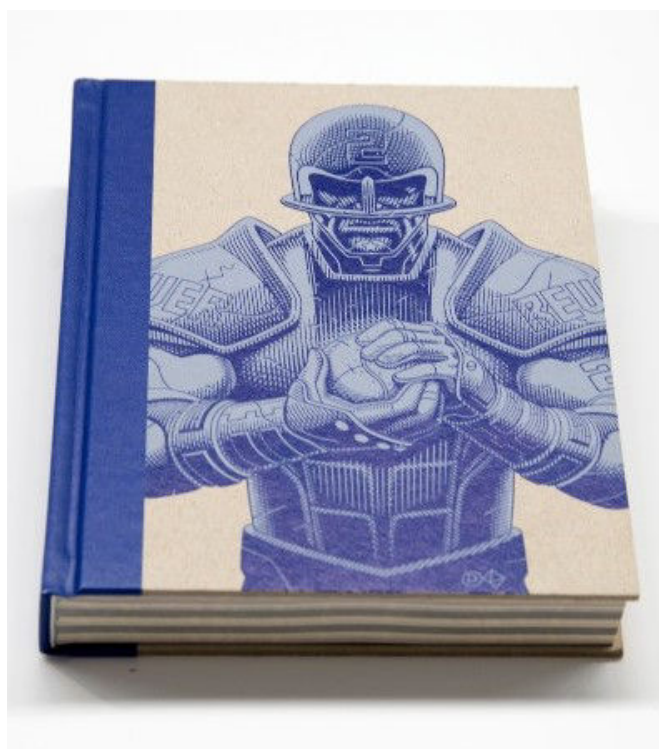
La qualità del volume è pari a quella del contenuto, la spessa copertina di cartone dona all'oggetto quell'aria massiccia che trasmettevano i capolavori Bitmap Brothers, la scelta di usare per i testi un violetto/indaco anziché il classico nero è azzeccatissima e perfettamente in linea con l'estetica Bitmap Brothers, le foto "cool" a cui si accennava prima e qui riprodotte in monocolor pixellato, stile foto realizzate con la mitica Gameboy Camera, sono un tocco di classe.

Un libro assolutamente consigliato, l'editore Read-Only Memory poi è da tenere sempre d'occhio, suggerisco di andare a guardare il loro catalogo che contiene opere di grande qualità.

Siamo giunti alla fine di questa piccola selezione di libri che mi sento di consigliarvi e ce ne sarebbero altri ancora, magari in futuro tornerò sull'argomento, chissà.

Se la storia dei videogiochi vi appassiona quanto giocarli, date un occhio a questi libri e se volete suggerire altri volumi che secondo voi meritano di essere letti, vi aspettiamo sui nostri canali social!

Saluti a tutti e Amiga Forever!



**NEW GAME**

# G-ZERO



Anno 2560. La razza umana è riuscita a sopravvivere ad un terribile invasione aliena, subendo ingenti perdite di vite. Per celebrare il grande giorno della vittoria, ogni 4 anni si tiene una gara, il G-ZERO Grand Prix! Solo i migliori piloti possono partecipare e solo uno sarà il vincitore.

G-ZERO è un semplice gioco di corse arcade per Game Boy ispirato alla leggenda F-ZERO per Super Nintendo. In ogni gara si dovranno completare due giri completi della pista e cercare di arrivare primi. Per vincere il trofeo si dovranno portare a termine le sei gare presenti.

Il nostro mezzo (su tre da scegliere) ha una potenza limitata, che diminuisce ogni volta che tocchiamo uno dei bordi della pista e una macchina avversaria. Se il potere termina, la

gara terminerà e sarà un game over. Si usa la croce direzionale per muoversi, il pulsante A accelera e il B rallenta la macchina. Il tasto start, infine, mette in pausa durante la partita e ovviamente avvia il gioco. È un buon homebrew, un omaggio al celebre fratello maggiore nato agli inizi degli anni 90.

Non ha lo stesso senso di velocità e le meccaniche di scivolamento che amavamo su Super Nintendo, ma c'è una buona sensazione di fluidità e un ottimo lavoro sulle mappe presenti. Il gioco è stato sviluppato in C utilizzando GBDK e solo la routine delle scanline ISR è scritta in ASM. Può essere giocato in emulazione e anche su hardware reale, cosa che consiglio perché rende l'esperienza molto più appagante.

È un titolo piccolo che si porta a termine dopo un po' di pratica ma che consente al giocatore di passare qualche ora sfrecciando sulle strade futuristiche del futuro.

Complimenti allo sviluppatore per questo tipo di prodotto, forse non il meglio del meglio ma sicuramente divertente e ben fatto.

di **Roberto Del Mar Pirazzini**



Anno: 2022

Editore/Sviluppatore: User0X7f

Genere: Driving

Piattaforma: Game Boy

Sito web: [https://](https://user0x7f.itch.io/g-zero)

[user0x7f.itch.io/g-zero](https://user0x7f.itch.io/g-zero)



## GIUDIZIO FINALE

### » Giocabilità 80%

Fluida e con percorsi ben strutturati. Comandi perfetti.

### » Longevità 70%

Non è difficile né lunghissimo, ma si gioca volentieri.





**NEW GAME**

# ATTACK OF PETSCII ROBOTS



Tecnicamente ricorda molto da vicino la versione Sega Megadrive ma con maggiore fluidità. Ho trovato ricca la grafica e di bell'aspetto.

Il titolo sfrutta l'audio FM OPL4 e devo dire che è un gran bel sentire. In emulazione questa cosa un po' si perde. Su BlueMSX questa possibilità è esclusa e quindi consiglio, per chi non possiede l'hardware originale, di utilizzare OpenMSX per la resa sonora perfetta.

I controlli di gioco sono personalizzabili, ma personalmente preferisco le impostazioni da tastiera predefinite rispetto al pad.

Come per le precedenti versioni c'è da dire che non è un gioco semplice d'approccio e richiede logica e molta pratica, ma è indubbiamente stimolante e piacevole.

Insomma un altro gioiellino su MSX. Godetevolo!

di **Giampaolo Moraschi**

**Sito web:** <https://www.msxdev.org/2023/09/19/msxdev23-18-attack-of-the-petscii-robots/>

Anno: 2023

Editore/Sviluppatore: The 8-bit-guy/Robosoft

Genere: Strategy/Adventure

Piattaforma: MSX V9990 con OPL4



Nato su Commodore PET e poi portato su una marea di sistemi differenti, il titolo di 8 Bit Guy arriva, grazie alle manine dorate di Robosoft anche sui nostri amati MSX V9990.

Lo scopo del gioco è sempre quello ovvero cercare di infiltrarti all'interno di alcuni insediamenti e distruggere tutti i robot ribelli. Il trucco sta nel trovare gli strumenti utili e usare l'ingegno invece della semplice forza bruta.

Azione dinamica e molto ben congegnata e un ottimo gameplay che non fa perdere interesse e tiene alta la tensione di gioco.

Questo porting esce durante l'MSXDev 2023 ed è stato reso possibile grazie alla fantastica libreria MSXgl di Aoineko e della libreria Replay di Grauw per la musica.



## GIUDIZIO FINALE



### » Giocabilità 85%

Un titolo ben sviluppato e rodato grazie alle numerose versioni su tante piattaforme che presenta un buon gameplay e comandi ben strutturati.

### » Longevità 80%

Non è per tutti e non è per chi cerca azione frenetica. Ma è solido e piacevole.





NEW GAME

# SAM'S JOURNEY

**Anno:** 2023**Editore/Sviluppatore:** Poly.play,  
Knights of Bites**Genere:** Platform**Piattaforma:** Nintendo NES**Sito web:** [https://  
www.knightsofbytes.games/  
samsjourney/nas](https://www.knightsofbytes.games/samsjourney/nas)

Definire Sam's Journey un buon gioco sarebbe un eufemismo. La presentazione generale e il gameplay sono a dir poco meravigliosi. Sam's Journey su Commodore 64 si è rivelata una boccata d'aria fresca ed un motore di spinta per la "new wave" di titoli di ultima generazione per il nostro amato "breadbin".

Scorrimento direzionale fluido e veloce che travolge i sensi visivi e un comparto audio magistrale e ambientazioni di gioco vivaci e coloratissime.

Era così su Commodore 64 e questa versione su NES è ancora più sfarzosa, grande e colorata.

Il gioco rimane fedele a sé stesso. Sam viene rapito nel cuore della notte da un grande artiglio che esce da un armadio. Si ritrova in un folle mondo coloratissimo che dovrà esplorare attraverso 30 livelli divisi in 3 megamappe per ritrovare la strada di casa. Livelli di gioco disseminati di una vasta gamma di nemici che vogliono impedire al piccolo e buffo Sam di continuare il suo viaggio attraverso i mondi. Il nostro ciiccotto eroe dovrà utilizzare interruttori, trampolini, passaggi segreti, gufi salterini e raccogliere le chiavi per progredire attraverso i vari livelli e capire come utilizzare tutti questi oggetti nel posto giusto o nel modo giusto. Grazie a costumi speciali (nascosti dentro forzieri o in altre posizioni) cambierà l'aspetto di Sam ed aumenteranno le sue abilità speciali. Il Pirata Sam è in grado di combattere con la sciabola, il Ninja può aggrapparsi ai muri, Disco Sam può roteare in giro e così via (ci sono anche Sam Astronauta e Vampire Sam, incredibili). I costumi sono anche utili in quanto forniscono uno strato di protezione



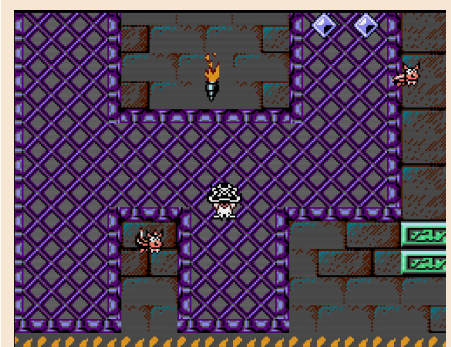
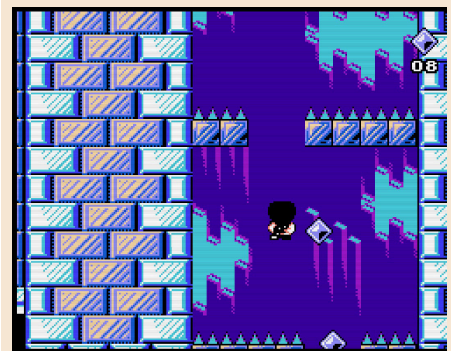
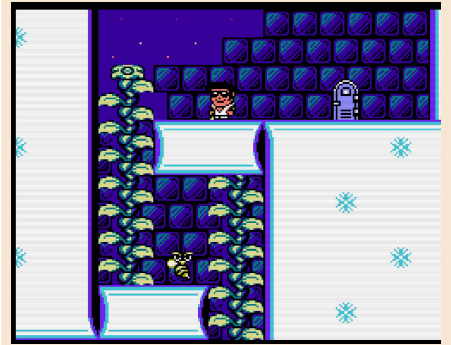
dai nemici, un po' come l'armatura di Sir Arthur in Ghost 'n' Goblins, in modo da evitare una morte da un colpo letale. Tuttavia è possibile indossare un solo costume alla volta, quindi è necessario scegliere se cambiare il costume ogni volta che ne incontri uno. In tutto questo il gioco contiene anche diamanti da raccogliere, monete e trofei collezionabili che aumentano il punteggio finale.

Spostare Sam nel suo mondo nuovo su NES è un piacere. Il personaggio, come nella sua incarnazione su Commodore 64, è piuttosto versatile grazie alla sua capacità di correre, saltare, arrampicarsi, nuotare e raccogliere oggetti di grandi dimensioni da lanciare sui nemici.

Nemici spesso ostici e tutti ben definiti e con movimenti e comportamenti diversi, il che significa che eliminarli richiederà spesso approcci diversi. Il gameplay è solido. I livelli sono ben







costruiti e la possibilità di cambiare i costumi è una caratteristica vincente e intelligente che introduce elementi strategici per affrontare una battaglia in un certo modo rispetto ad un altro. Il titolo permette di salvare attraverso comodi check-point come avveniva nella sua controparte per Commodore 64, opzione davvero gradita visto le 30 ore di gioco che ci troveremo di fronte per affrontare l'avventura. Questi check-point ci permettono di ripartire da quel punto in caso di morte e questa cosa è molto apprezzata.

Arriviamo ai paragoni... questa versione NES è superiore a quella Commodore 64?

Domanda che possiamo subito risolvere in questo modo... Questa versione ha una grafica con una maggiore risoluzione e molti più colori. Il pad con due tasti ci permette di ottimizzare al meglio il gioco rispetto al solo tasto

su C64. La visuale di gioco è decisamente più larga e, come vi dicevo, la possibilità di usufruire di un maggior numero di colori aiuta molto.

A livello audio preferisco la versione C64. Non fraintendetemi, il sonoro e l'accompagnamento su NES sono ottimi, ma su C64 il SID ha un "flow" maggiore e rende meglio.

Come per la versione precedente anche qui avrei preferito scegliere la possibilità di selezionare il costume invece di applicarlo automaticamente, ma è un piccolo piccolissimo pelo nel perfetto uovo che è Sam's Journey.

I Knights of Bytes hanno mostrato di saperci fare anche questa volta, creando uno dei platform più belli su Nintendo NES dell'ultima generazione e forse uno dei titoli più belli del genere per questa console di sempre, come senza ombra di dubbio avevano fatto realizzando questo gioco su Commodore.

Il titolo viene distribuito da Poly.play in formato cartuccia e, successivamente, in formato digital download.

Ha un bel packaging e un manuale che sprizza "old school" da tutti i pori.

Un altro grande titolo per questo 2023.

di Carlo Nithaiah Del Mar Pirazzini

## GIUDIZIO FINALE

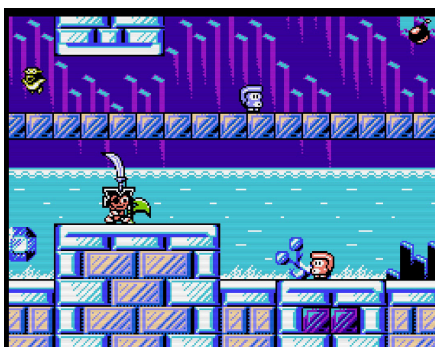


### » Giocabilità 98%

Gameplay bello e dinamico. Livelli ben strutturati e tante chicche in gioco. Il pad a due tasti aggiunge una maneggevolezza migliore rispetto al joystick su C64.

### » Longevità 95%

Ci sono 30 livelli e vanno affrontati con coraggio e accuratezza. Non impossibile, ma decisamente ben calibrato.





**NEW GAME**

# ELECTRONIC DREAMS THE SPACE ODISSEY

Anno: 2023

Editore/Sviluppatore:

Mikgames

Genere: Action game

Piattaforma: Sega Master  
System

Sito web: [https://  
www.smspower.org/forums/  
19827-ElectronicDreams](https://www.smspower.org/forums/19827-ElectronicDreams)

È il 2123 e vivi sul pianeta Terra. Un gruppo di terroristi attaccano il cuore di Capital City, uccidendo nella strage anche tuo padre.

Sei una guerriera e sei pronta a vendicarlo.

Il tuo viaggio sarà lungo tutto il sistema solare, dalla tua casa alla luna Sirio. Farai di tutto per portare a termine la tua vendetta.

Un titolo nuovo e fresco per Sega Master System che omaggia tantissimi altri giochi.

Troverete un po' di Contra e di Metroid con un sacco di minigiochi interni e persino un livello shoot em up multidirezionale davvero ben fatto. La protagonista è una ragazza, inizialmente armata della sua pistola a ripetizione.

Dovrà farsi largo lungo i livelli tra nemici, trappole e salti di piattaforme estremi e potrà anche raccogliere armi e caricatori aggiuntivi per progredire più facilmente.

Ogni livello ha un suo boss ben realizzato che dovrà essere abbattuto con un'attenta osservazione dei pattern di combattimento.

Un tasto del pad serve per far fuoco e un tasto per saltare. Il tasto PAUSE funziona come menù per la selezione delle armi raccolte durante l'avventura. È un titolo che può essere affrontato in modalità normale oppure in modalità "facile". Quest'ultima non ha il livello di energia del personaggio e le munizioni.



Il gioco è piuttosto impegnativo e mette in trappola il giocatore disattento.

Mi è piaciuta la parte grafica con la classica palette del Sega Master System. C'è tanta varietà nei livelli, diverse animazioni interessanti e si muove il tutto senza rallentamenti.

Una bella colonna sonora (che mi ha ricordato Mega Man) ci accompagna lungo dei livelli strutturati con cura.







## GIUDIZIO FINALE



## » Giocabilità 90%

Diverse modalità di gioco lo rendono interessante.

## » Longevità 80%

La difficoltà è forse troppo alta.

Comandi non reattivi.

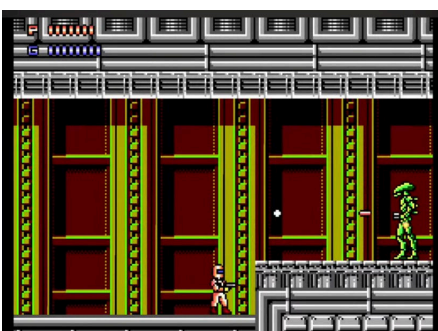
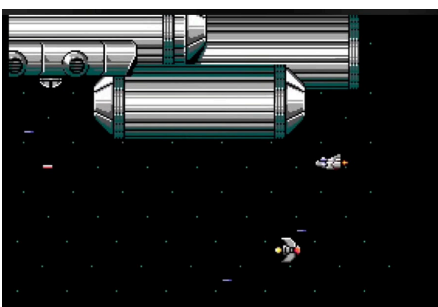
La scelta di inserire più modalità di gioco (corsa in moto, livello metroid, livello shoot'em-up e i minigiochi) rendono il titolo interessante.

Punti dolenti: i comandi non sono reattivi come dovrebbero e forse una sola vita è troppo poco.

Lo sviluppatore sta però sistemando il tutto e promette che ci saranno future versioni corrette.

Tirando le somme è un buon titolo per il nostro 8bit targato Sega.

di **Giampaolo Moraschi**





NEW GAME

# RESHOOT PROXIMA 3

Anno: 2023

Editore/Sviluppatore:

Spieleschreiber

Genere: Shoot em up

Piattaforma: Amiga AGA

Sito web: [https://](https://spieleschreiber.itch.io/rp3)[spieleschreiber.itch.io/rp3](https://spieleschreiber.itch.io/rp3)

Dopo anni di annunci e una lunga gestazione (dal 2019) ecco che arriva Reshoot Proxima 3.

Rilasciato dopo l'Amiga Germany 38 dagli sviluppatori Spieleschreiber, un collettivo già noto per la forte presenza nella scena demo.

Un roboante shoot em up a scorrimento verticale dal look impressionante e... un po' di problemi che non mi aspettavo.

Il gioco appare nella maniera dei più classici titoli del genere: alieni da blastare lungo 5 livelli piuttosto articolati e particolareggiati (notevole il secondo livello nel cielo) con boss fight alla fine di ogni livello.

La nostra nave si può potenziare con due tipi di fuoco, un pod doppio di supporto e lo speed up (per un totale di tre potenziamenti effettivi) che fungono anche da "barra d'energia". Ogni volta che veniamo colpiti perdiamo un potenziamento fino a perdere la vita.

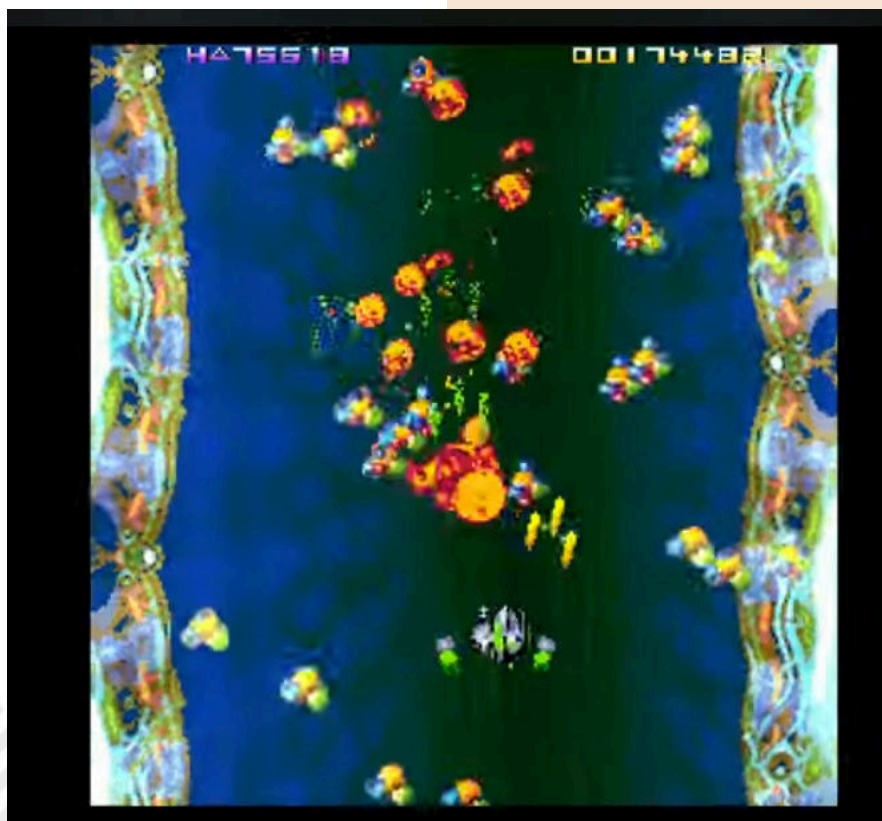
Dopo uno scarico menu con le opzioni, il gioco ci mette di fronte a 5 livelli differenti dalla lunghezza variabile e con un buon design.

Ogni livello è caratterizzato bene e presenta non solo le classiche ondate dei nemici in caduta, ma mette alla prova anche i riflessi dei giocatori nel districarsi tra numerose trappole. Importantissimo (e un po' frustrante) ricordarsi il numero e l'esatta posizione delle wave avversarie. Colpire successivamente porta a delle compo



che accumulano punti per lo score finale.

E' decisamente un titolo che strizza







## GIUDIZIO FINALE

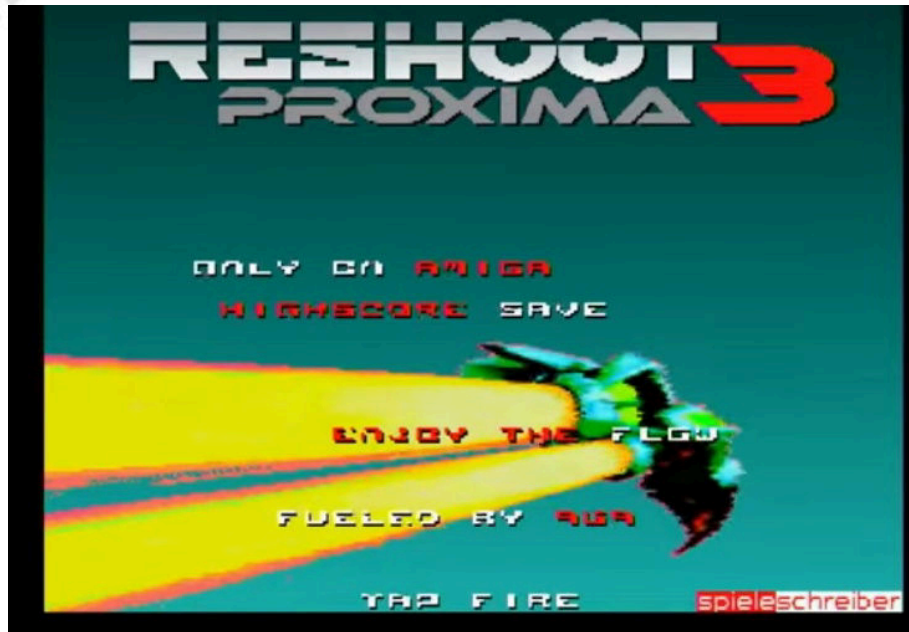


## » Giocabilità 80%

Livelli ben strutturati ma stile di gioco non proprio moderno. Il secondo tasto come "Pausa" non si può vedere.

## » Longevità 80%

È sicuramente un bel giocare, anche se bisogna ricordarsi ogni singola wave per poter proseguire.



l'occhio agli shooter arcade di un tempo. Condivide con molti di loro la difficoltà e la durezza.

Tecnicamente è un gioiellino. Grafica coloratissima in pixel art a 24bit super veloce e super animata. Alcuni livelli sono una gioia per gli occhi (il secondo livello e quello sottomarino sono incredibilmente belli).

Il tutto fila liscio anche nel semplice A1200 basico dove abbiamo testato il gioco senza rallentamenti.

Sonoro pompato e ben fatto. Si vede l'impostazione da Demoscene.

Arriviamo alle note dolenti.

Poteva essere il capolavoro Amiga della nuova generazione ma è soltanto un bel gioco.

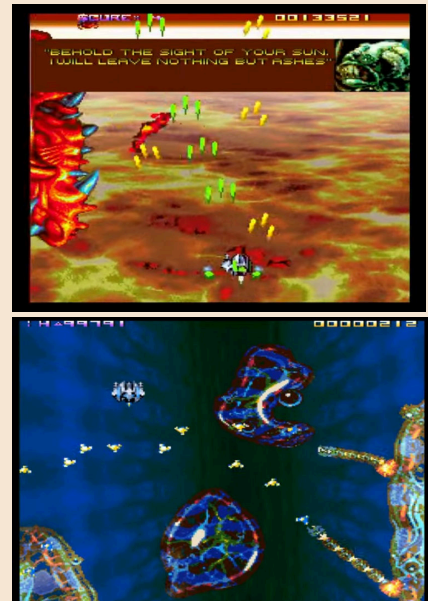
La struttura degli assalti nemici come dicevamo è un po' arcaica per i tempi di oggi così come l'impossibilità di avere un solo tipo di fuoco e di non poter avere la smart bomb (che in certi casi è essenziale!).

Seppur supportato, il secondo tasto viene limitato ad un semplice tasto per mettere in pausa l'azione. Troppo poco e personalmente ritengo la scelta irritante.

Mi auguro una futura patch che possa espandere il gioco e migliorare le problematiche che vi ho elencato.

Reshoot Proxima 3 resta comunque un titolo ben fatto e giocabile a cui manca quel "pizzico" in più per poter essere perfetto.

di Roberto Pirazzini



**NEW GAME**

# ROCKY & CO

**Anno:** 2023**Editore/Sviluppatore:** Icon 64  
– Psytronik Software**Genere:** Puzzle/Platform**Piattaforma:** Commodore 64**Sito Web:** [https://](https://psytronik.it/rocky-and-co)[psytronik.it/rocky-and-co](https://psytronik.it/rocky-and-co)

Il grande esploratore Rocky Memphis ha scoperto l'ingresso del tempio perduto che, secondo lui, contiene il leggendario tesoro conosciuto come il CONDOR D'ORO.

Sulle tracce di questo manufatto antico si trova anche il suo acerrimo nemico Russo Baluch, intenzionato a metterci le mani sopra.

Quindi, accompagnato da due compagni d'avventura, parte per una incredibile caccia al tesoro nascosto. Rocky & Co è un puzzle platform che mi ha ricordato The Lost Viking e il titolo Datasoft The Goonies. Ciascuno dei tre membri della squadra ha un insieme di abilità diverse da cui si può trarre vantaggi. L'obiettivo di ogni livello è utilizzare una combinazione di abilità per portare in salvo tutti e tre gli eroi alle vie d'uscita evitando ogni pericolo sulla strada (e vi assicuro che di pericoli ce ne sono tantissimi!).

Il giocatore controlla un personaggio alla volta. Per cambiare personaggio basta cliccare la Barra Spaziatrice o in alternativa cliccare per un po' il tasto Fire e premendo la leva in basso. Nel risolvere tutti i vari livelli, ci saranno interruttori su cui stare in piedi, maniglie da abbassare e, a volte, bisognerà combinare le abilità di due personaggi per far funzionare qualcosa.

Il gioco è strutturato su 12 livelli standard, ma se si riesce ad illuminare le lettere della scritta ICON nel



pannello di gioco inferiore si sbloccherà una fase bonus e un finale "speciale". Questo gioco è un ritorno del protagonista su Commodore 64.

Prendendo in prestito il design di The Goonies di Datasoft è un perfetto esempio delle capacità del team Icon







## GIUDIZIO FINALE

### » Giocabilità 80%

Bel strutturati gli enigmi e la modalità di gioco. Avrei incluso una versione per l'uso dei due tasti in gioco. Bella anche l'opzione dei livelli bonus e il finale alternativo.

### » Longevità 80%

Dodici livelli possono sembrare pochi ma vi assicuro che ci impiegherete il tempo giusto per portarlo a termine. Interessante anche l'opzione Pratica per i nuovi giocatori. Meccaniche un po' old school.

64 di produrre un prodotto moderno di alta qualità e superando il titolo ispiratore. Trovo interessante l'uso dei personaggi in gioco e il design di alcuni puzzle davvero ben pensati e generalmente accessibili.

Grafica dettagliata e ricca. Sembra davvero saltar fuori dallo schermo. La colonna sonora è di Saul Cross, autore anche del precedente titolo di Rocky Memphis, perfetta per il gioco e fa moltissimo per migliorare l'esperienza complessiva.

Rocky & Co., contiene solo 12 livelli standard, ma questo non lo rende un gioco veloce e semplice poiché i puzzle sono strutturati ad una difficoltà crescente che richiederà molta abilità e ingegno. Hanno quell'elemento di prova e riprova alla Rick Dangerous che può trasformare il gioco in una vera sfida (nessun continue e solo 5 vite, ocio!).

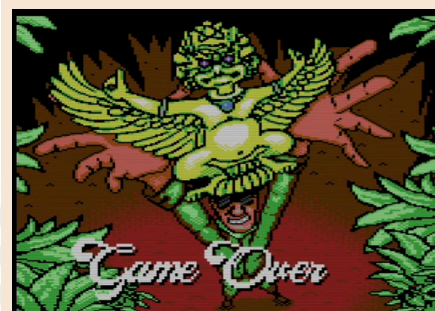
Icon 64 per venire incontro ai casual gamers ha inserito una modalità pratica che permette di "lavorare" sullo stile di gioco da affrontare per iniziare l'avventura. Scelta saggia che aumenta di qualche punto la longevità del prodotto.



Ha alcune meccaniche della vecchia scuola ed è sicuramente non semplicissimo, ma è un buon titolo.

Se vi piacciono questo tipo di giochi e amate il connubio platform/puzzle allora Rocky & Co è quello che fa per voi.

di Carlo Nithaiah Del Mar Pirazzini





NEW GAME

# SNK VS CAPCOM

Anno: 2023

Sviluppatore: Gianluca Alberico

Genere: Beat em Up

Piattaforma: Commodore 64

Sito web: [https://](https://www.youtube.com/watch?v=fPOwBpNTXv8&ab_channel=RetroGL)[www.youtube.com/watch?](https://www.youtube.com/watch?v=fPOwBpNTXv8&ab_channel=RetroGL)[v=fPOwBpNTXv8&ab\\_channel=](https://www.youtube.com/watch?v=fPOwBpNTXv8&ab_channel=RetroGL)

RetroGL

Sono tantissimi anni che il Commodore 64 ci fa compagnia. Anni di titoli indimenticabili e infiniti pomeriggi passati al suo cospetto. Infiniti come i giochi che continuano ad uscire sul leggendario "biscottone".

Questo 2023, quarantunesimo anno di vita del C64, ha visto l'uscita di perle come L3STER, A PIG QUEST! e questo SNK vs CAPCOM.

Autore del titolo il caro **Gianluca Alberico** (accompagnato da **Jon Egglton**). Inizialmente nato come motore di sviluppo per piacchiaduro su C64 e poi evolutosi in un gioco completo.

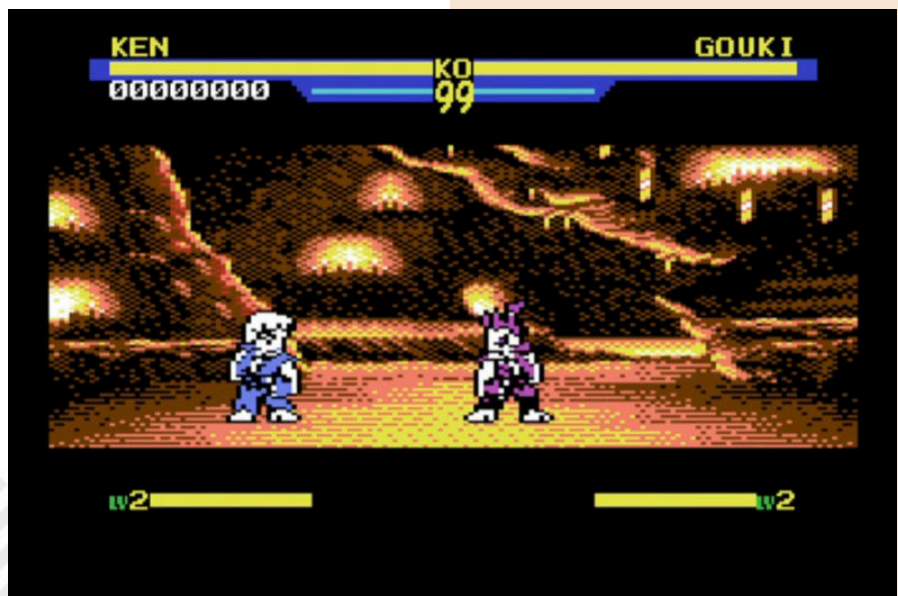
Il titolo si ispira alla saga SNK vs CAPCOM e in particolare al titolo "MATCH OF THE MILLENNIUM" già ammirato sul poco fortunato Neo Geo Pocket Color. Uno sforzo di programmazione, grafica e sonoro mostruoso inserito in una cartuccia da 1MB (file CRT). Sarà difficile vedere una versione in altri formati. È un gioco "ciccione" e necessita di tante risorse.

Cosa vediamo all'avvio di questo titolo? Beh, sono presenti una bella serie di opzioni. C'è la canonica Story Mode che vi vedrà impegnati a liberare il mondo dallo Shadaloo del cattivissimo VEGA (Bison per chi lo conosce dalla versione USA) e dal suo nuovo alleato G. Howard. C'è la versione VERSUS adatta per chi vuole "prendersi a mazzate" in compagnia ed infine c'è l'interessante versione TOURNAMENT con singoli scontri diretti ed eliminatorie, quarti di finale, semifinali e finale del torneo. Modalità questa anche per due giocatori. Disponiamo anche di altre opzioni che ci permettono di selezionare la

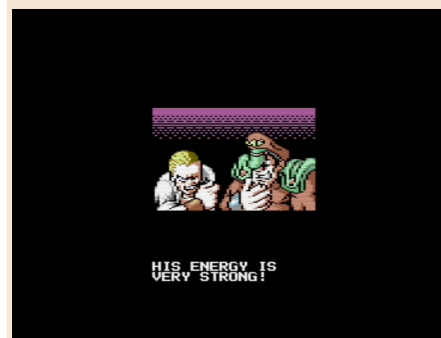
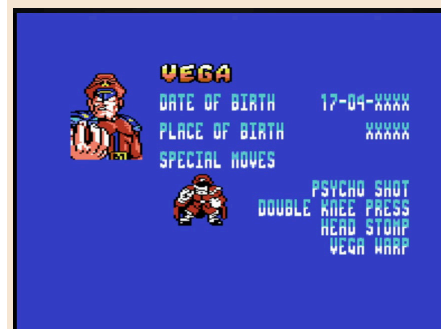


difficoltà (easy, normal e hard) e la possibilità di utilizzare il joypad con il supporto del secondo pulsante. Opzione molto gradita quest'ultima perché permette di avere i calci e i pugni separati.

Nel gioco ci sono tutti i colpi possibili: combo moves, parate, reverse, super mosse e mosse speciali e sono spiegate benissimo in un bel manuale in







PDF di ben 30 pagine. C'è tutto e c'è scritto come farlo. Super!

Sono stati aggiunti 3 nuovi combattenti rispetto all'Alpha release precedente, 2 nuovi round bonus e l'intelligenza artificiale è stata migliorata per aumentare la sfida.

Siamo di fronte al miglior picchiaduro per Commodore 64 di sempre! Senza se e senza ma. Ha tutto! Grafica di livello con tante animazioni e caratterizzazioni dei personaggi e degli stage, un bellissimo comparto audio e una giocabilità perfetta che porta ad una longevità pressoché infinita.

Non manca nulla... e se non bastasse è gratis e si presta a modifiche e miglioramenti futuri. Un capolavoro!

di Carlo Nithaiah Del Mar Pirazzini

## GIUDIZIO FINALE



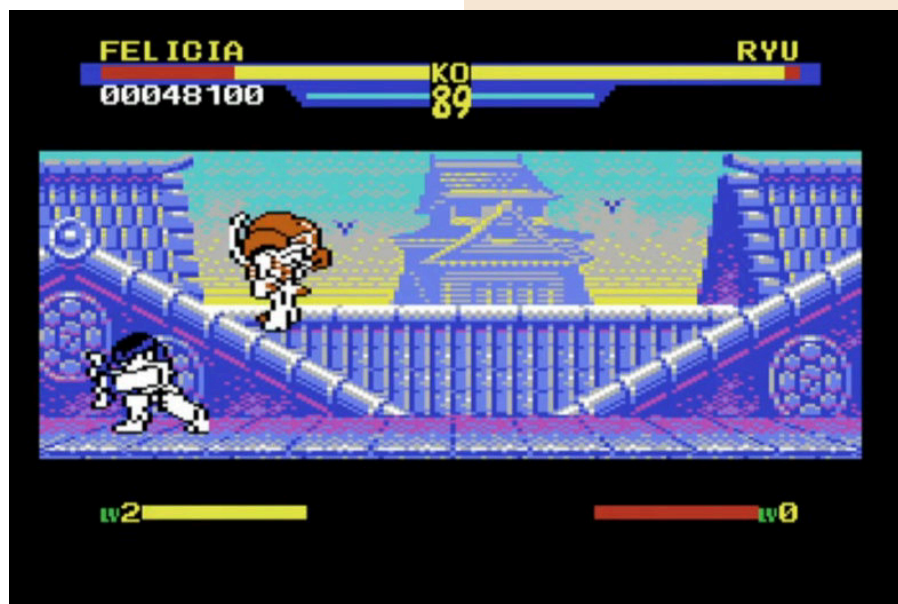
### » Giocabilità 99%

Tutto ben bilanciato. Comandi perfetti e l'opzione del secondo tasto.

I tre livelli di difficoltà sono perfetti e la possibilità di poter giocare in tre modi diversi lo rendono appetibilissimo.

### » Longevità 99%

Eterno come tutti quei buoni picchiaduro di un tempo.







# LEGEND OF WUKONG

Anno: 1996/2023

Editore/Sviluppatore: Gamtec,  
Super Fighter Team

Genere: RPG

Piattaforma: Sega Megadrive

Sito web: [https://](https://www.legendofwukong.com/)

[www.legendofwukong.com/](https://www.legendofwukong.com/)

Un bel titolo uscito in Cina nel 1996, tradotto in inglese nel 2008 e finalmente reinserito nel mercato anche in formato cartuccia, questo Legend of Wukong (Wukong Waizhan in cinese) è un gioco di ruolo sviluppato dalla taiwanese Gamtec per Sega Megadrive (ma uscito anche su MacOS e PC).

La sua storia ricorda vagamente quella del romanzo Viaggio in Occidente, un racconto cinese pubblicato nel XVI secolo attribuito a Wu Cheng'en e considerato uno dei più grandi classici cinesi e l'opera letteraria più popolare in oriente. Così popolare da generare numerosi cartoni animati, ad esempio ispirando Akira Toriyama per la saga di Dragon Ball.

La traduzione in inglese è stata realizzata dal team Super Fighter Team, rendendo il titolo più giocabile e fruibile fuori dal lontano oriente.

Il gioco è simile a molti giochi di ruolo a turno degli anni 90. Le battaglie vengono viste di lato e noi dobbiamo semplicemente pianificare i nostri attacchi, le nostre parate, l'uso di cure e oggetti per ciascun personaggio e guardare lo svolgersi degli eventi. C'è molta esplorazione e tanto da fare in termini di esperienza in gioco: tutto quello che ci si deve aspettare da un classico gioco di ruolo.

Per me, ciò che distingue questo titolo è lo sviluppo della storia. Tantissime scene di intermezzo ci spiegano il viaggio del protagonista e sono unite ad una serie di dialoghi pieni di esempi spiritosi e divertenti che spesso vi faranno sbellicare dalle risate. Un impatto differente dalle solite trame tristi o malinconiche dei giochi di

ruolo standard dell'era 16 bit. Ho davvero gradito questa freschezza e spensieratezza narrativa durante il test. Il gioco è ben bilanciato. Le battaglie, anche se piuttosto frequenti, sono fluide e corrette. Non ci sono "Giri a vuoto" o percorsi dove perdersi; se tutto è fatto come comanda la storia il gioco scorrerà in modo fluido e interessante, coinvolgendo il giocatore fino allo scontro finale.

La magia in gioco è varia e molto potente, anche se manca di "caratteristica". Forse dovuto alle animazioni semplici che gli hanno dato gli sviluppatori. Svolge il suo lavoro in gioco e questo può e deve bastare.

Notevole il lavoro svolto nei dungeon. Sebbene non enormi nell'esplorazione sono ben strutturati e mettono alla prova un party di gioco che non ha esperienza. Suggerisco di entrare nei dungeon dopo aver livellato un pochino il gruppo di gioco.







## GIUDIZIO FINALE



### » Giocabilità 90%

Un buon design di livelli unito ad una storia solida e a dialoghi brillanti e divertenti.

### » Longevità 90%

Dungeon ben strutturati, tanta esplorazione. Un mix perfetto per diverse ore di gioco.

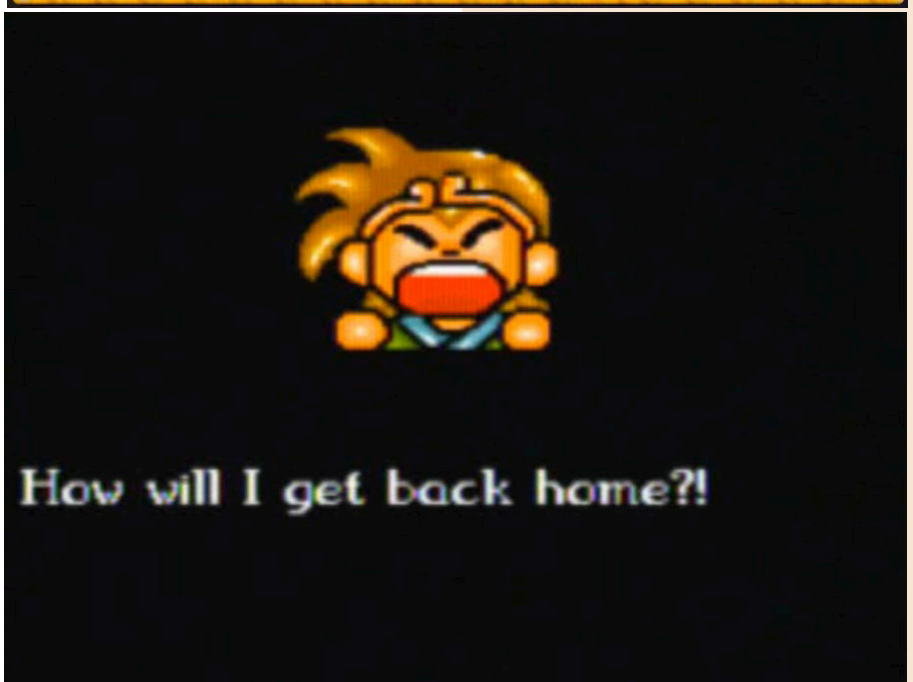
La narrazione, il gameplay, il design: tutto questo è realizzato in modo particolarmente raffinato, con un sottile fascino estetico che cresce di più andando avanti nei livelli. L'aspetto grafico è classico ma funzionale e non è necessario avere un sacco di parallasse, scrolling o effetti di luci mirabolanti, quello che c'è è perfetto così. Le città e i dungeon sono colorati e dettagliati e Gamtec ha aggiunto piccoli tocchi davvero ben fatti.

Una buona colonna sonora condiscie il tutto e si adatta all'avventura. Ogni area ha il suo tema di apertura e cambia alla pari di ciò che ci appare andando avanti nel gioco, presentandosi sempre diversa.

La conclusione è che Legend of Wukong è un solido gioco di ruolo dallo stile raffinato da aggiungere alla scuderia di questo genere per il nostro Megadrive. L'interazione tra gli eroi, la varietà dei nemici e degli ambienti e i dialoghi divertenti sono un ottimo modo per trascorrere del tempo.

Un titolo da riscoprire che finalmente trova anche una bella versione su cartuccia con tanto di manuale ben fatto e colorato.

di **Carlo Nithaiah Del Mar Pirazzini**





NEW GAME

# PAMPAS & SELENE

## INTO THE MAZE OF DEMONS

Anno: 2023

Editore/Sviluppatore:

UnEPICFRAN

Genere: Platform

Piattaforma: MSX 2

Dopo aver sconfitto il terribile Galious, gli eroici Popolon e Afrodite tornarono alla loro vita quotidiana. Vivendo una vita felice come coppia nel loro accogliente castello e allevando il loro primogenito Pampas, istruendole alle regole del cavalierato.

Trent'anni dopo un nuovo nemico si scaglia all'orizzonte e questa volta toccherà al giovane Pampas liberare il mondo dal male.

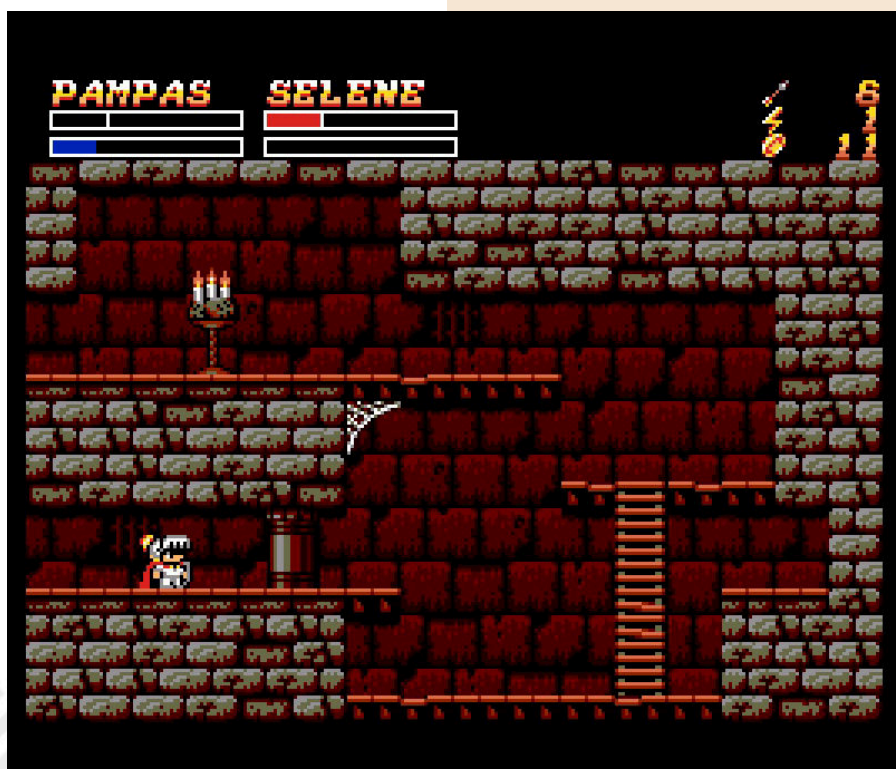
Questo titolo per MSX2 è il seguito "spirituale" del famoso Nightmare II- Maze of Galious che Konami realizzò nel lontano 1987 (a sua volta seguito del bellissimo Nightmare).

Il gameplay ricorda quello del suo celebre predecessore. Il giocatore può scegliere se utilizzare il giovane eroe o la sua sorellina Selene, in modo da poter utilizzare le particolari caratteristiche di entrambi per poter proseguire nelle oltre 250 stanze piene zeppe di mostri, trappole e livelli segreti.

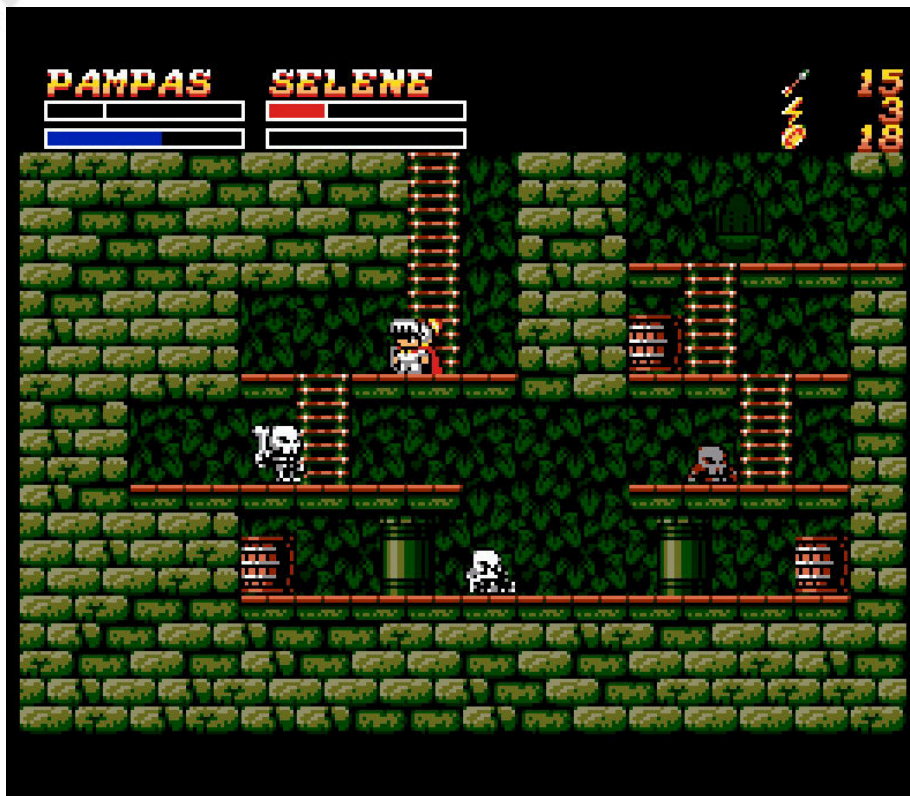
Il gioco è molto vasto e offre una notevole varietà di elementi di esplorazione.

Titolo notevole e ben realizzato che utilizza la modalità grafica screen 5 per la grafica.

Tecnicamente è senza dubbio un







## GIUDIZIO FINALE

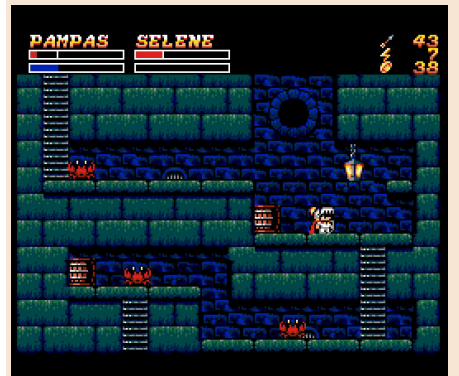


### » Giocabilità 95%

Gameplay rodato e divertente.

### » Longevità 98%

Impegnativo e VASTISSIMO! Ci vorrà tempo per portarlo a termine tutto e questo è un bene.



prodotto di livello sotto ogni punto di vista. Le animazioni, i colori, la velocità in gioco... tutto funziona alla perfezione.

È un titolo concepito come ai vecchi tempi e richiede manualità nelle meccaniche di gioco.

Mappa vastissima e gameplay ben sviluppato rendono questo titolo uno dei migliori titoli per MSX 2 di sempre.

di Giampaolo Moraschi





NEW GAME

# STAR WARS: EMPIRE STRIKES BACK

La forza scorre potente su Commodore Plus/4! Lo dimostrano la marea di titoli usciti recentemente su questa piattaforma, tutti piccoli capolavori perfettamente convertiti.

Mi piace il Plus/4! È una di quelle piattaforme realizzate in modo "strambo" ma con un grande potenziale.

Quando uscì non venne nemmeno capita e fu sfruttata malissimo, ma grazie al cielo un manipolo di fan sta sbalordendo noi addetti ai lavori e tutti voi appassionati con titoli meravigliosi.

In questo caso si tratta di una conversione da un titolo moderno per Commodore 64 ovvero la versione Fan di Star Wars Empire Strikes Back uscita nel 2021.

Se la versione vista sul biscottone faceva cadere le mascelle questa è ancora più impressionante se pensiamo ad alcuni limiti hardware della macchina.



Il piccolo Plus/4 se la cava benissimo con questo titolo e lo fa mettendo in mostra una buona velocità di gioco e un grandissimo uso del chip sonoro TED.

Tecnicamente bello come quello per C64 ma con lo stesso problema.

Empire Strikes Back è sostanzialmente un clone di Defender, gioco meraviglioso ma figlio del suo tempo. Nel 2023 uno stile di gioco simile potrebbe annoiare e risultare ripetitivo. Ma chi se ne importa, alla fine! È un bel gioco ed è ben realizzato in questa sua versione e merita senza ombra di dubbio di essere provato.

di **Giampaolo Moraschi**



Anno: 2023

Editore/Sviluppatore: TCFS

Genere: Shoot em up

Piattaforma: Plus/4

Sito web: <https://plus4world.powweb.com/software/>

The\_Empire\_Strikes\_Back



## GIUDIZIO FINALE

### » Giocabilità 95%

Uno dei giochi più divertenti della golden age. Semplice, immediato e divertente.

### » Longevità 90%

Curva della difficoltà crescente e una gran voglia di vedere come va a finire.







NEW GAME

# SUPER MONZA GP 2

Anno: 2023

Editore/Sviluppatore: AJ Layden

Genere: driving game

Piattaforma: VIC 20

Sito web: [https://aj-](https://aj-layden.itch.io/super-monza-gp-2)[layden.itch.io/super-monza-gp-2](https://aj-layden.itch.io/super-monza-gp-2)

## GIUDIZIO FINALE

### » Giocabilità 90%

Semplice e ben strutturato.

### » Longevità 90%

La modalità Campionato è un punto in più alla durata del gioco.

Un gioco di corsa con visuale in soggettiva della vecchia scuola, realizzato con passione da AJ Layden per tutti i VIC 20 con espansione a 32K. Il gioco espande il precedente titolo aggiungendo tre nuovi circuiti (Monaco, SPA e Interlagos).

Si muove veloce, con un bell'aspetto grafico e una cura dei dettagli impagabile.

Rispetto al primo capitolo ora le auto ricevono danni da possibili collisioni e possono essere riparate entrando nei box durante la gara. Più danni

equivalgono a una velocità minore. È stata inserita la modalità campionato in cui si deve completare ogni gara entrando a podio per poter continuare. Funziona su sistemi PAL e NTSC ma è progettato per i primi. Abbiamo notato che in modo NTSC gira leggermente più veloce.

Sonoro minimalista ma efficace.

Il titolo si lascia giocare bene e invoglia ad andare avanti nei circuiti successivi. Consigliatissimo a tutti gli appassionati di corse.

di **Giampaolo Moraschi**





NEW GAME

# CORNELIUS IN THE FORBIDDEN ZONE

In un mondo di giochi con grafica 4K, ultra realismo, CGI e di tutto di più, noi amanti dei computer retrò siamo probabilmente più entusiasti di vedere immagini disegnate a mano sullo schermo. La vera maestria dei pixel da parte di persone vere. È un'arte, sia che si utilizzi un vero pennello, una matita o una configurazione tastiera/mouse. Vedere su MSX un titolo dalla grafica curata come quella di questo Cornelius è gioia per gli occhi e fa battere il cuore.



Ispirato alla serie di film "il pianeta delle scimmie", il gioco Cornelius in the Forbidden Zone, è l'ultima produzione dei Physical Dream, casa di sviluppo di nuovi titoli per MSX.

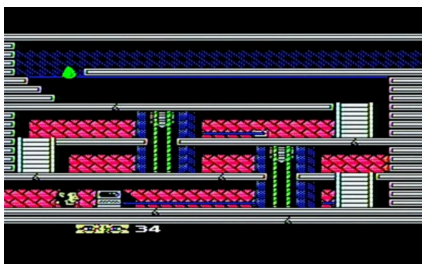
Nel gioco guideremo lo scimpanzé Cornelius attraverso i territori proibiti dove vivono i terribili... esseri umani! Il gioco segue i classici di questo genere. Si saltano i nemici, si evitano le trappole e si cerca di arrivare inermi (con tutto la barra d'energia piena) alla fine del livello.

L'azione di gioco si svolge in modo fluido e li livelli crescono di difficoltà



con una giusta misura.

Forse non sarà il più originale dei titoli presenti su questo sistema di gioco, ma è decisamente divertente e dal punto di vista tecnico ben sviluppato. La grafica semplice è apprezzabile tanto quanto il sonoro, incalzante e ben fatto.



Il titolo è disponibile in inglese, spagnolo e italiano ed è disponibile anche in versione fisica contando i ragazzi di Physical Dream via email.

Il costo del titolo sarà di circa 30 euro, un prezzo decisamente onesto per un titolo divertente.

di **Giampaolo Moraschi**

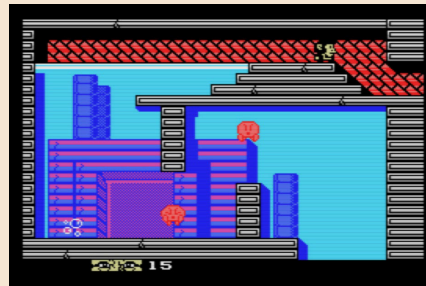
Anno: 2023

Editore/sviluppatore: Physical Dreams

Genere: Platform

Piattaforma: MSX

Sito Web: <https://www.facebook.com/profile.php?id=100064593073130>



## GIUDIZIO FINALE



### » Giocabilità 85%

Semplice ma ben fatto, si lascia giocare senza troppe cose da pensare.

### » Longevità 80%

Non è lunghissimo ma è divertente.







NEW GAME

# TIME WIZARD



Prima recensione su RetroMagazine World e primo titolo per una macchina che mi ha sempre affascinato.

Da piccola mio padre aveva un negozio di informatica pieno di tutte queste sigle strane per me allora: "Atari", "Commodore", "Spectrum". Io però giocavo poco ai videogames: ero più interessata alle Barbie.

Solo crescendo ho cominciato ad appassionarmi al retrogaming, racimolando tra mercatini e Internet le cose che più mi incuriosivano.

Gli Atari a 8-bit sono macchine incredibili. Hanno sempre mostrato giochi coloratissimi e bellissimi e ho recuperato un computer un annetto fa da un amico. Peccato sia ancora da riparare.

Intanto provo a giocare in emulazione (sono conscia che non sia la stessa cosa, ma ahimè, per ora mi devo accontentare!).

Time Wizard è un gioco di piattaforme con una trama che definirei banale. I robot cattivi vogliono distruggere il mondo cambiandone la storia. Noi interpretiamo il Time Wizard, il mago del tempo, unico baluardo rimasto a

difesa dell'umanità.

Verremo mandati nel futuro per prevenire il disastro.

Il compito sarà quello di raccogliere le clessidre del tempo in modo che i robot non possano tornare indietro. Il tempo lo si può fermare anche durante il gioco se commettiamo un errore.

Durante la partita si può premere il pulsante di fuoco sul pad per avviare la manipolazione temporale. Tenendo premuto, spostandoci a sinistra si può riavvolgere, mentre a destra ripristinare. Un'opzione curiosa che mi ha entusiasmato tantissimo.

Nella schermata introduttiva è possibile selezionare uno dei 15 livelli. All'inizio tutti i livelli tranne il primo sono bloccati.

È necessario superare con successo il livello per sbloccare il successivo. Il gioco ha partecipato alla ABBUC Software Contest 2023, manifestazione per sviluppatori Atari, piazzandosi al primo posto. Per me una posizione meritatissima.

È divertente, ha un concept di gioco ben strutturato e congegnato ed è eccellente il lavoro che ha fatto lo sviluppatore utilizzando il MADPascal, un compilatore Turbo Pascal a 32-bit per Atari XE/XL.

La grafica è coloratissima e veloce e la colonna sonora (che fa ampio uso di musica classica) è bellissima.

Sono contenta di aver iniziato su RMW con il piede giusto grazie a questo titolo.

Il gioco gira senza problemi in emulazione con una configurazione con almeno 128KB di RAM.

di **Marta Rossmann**

Anno: 2023

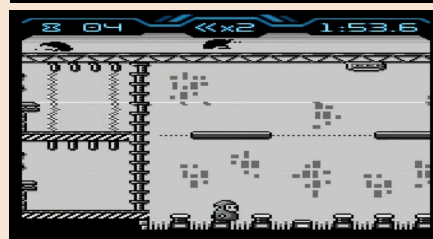
Editore/Sviluppatore: Amarak

Genere: Platform

Piattaforma: Atari XL/XE

Download: [https://](https://forums.atariage.com/topic/354234-time-wizard-a-game-for-abbuc-software-contest-2023/page/3/)

[forums.atariage.com/topic/354234-time-wizard-a-game-for-abbuc-software-contest-2023/page/3/](https://forums.atariage.com/topic/354234-time-wizard-a-game-for-abbuc-software-contest-2023/page/3/)



## GIUDIZIO FINALE



### » Giocabilità 92%

Bellissime le meccaniche di manipolazione del tempo e l'impostazione tecnica.

### » Longevità 90%

Le meccaniche di gioco e l'impostazione dei livelli vi terranno impegnati.





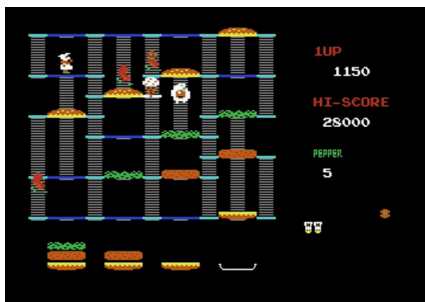
NEW GAME

## BURGERTIME

Peter Pepper su Commodore 64 non è mai stato così in forma!

Certo, il suo aspetto è sempre quello del 1982 ma in questa sua nuova apparizione su C64 appare splendido e splendente più che mai.

Il meraviglioso e pluri-convertito titolo di Data East per DECO Cassette System giunge grazie ad Arlasoft in una versione praticamente perfetta.



Tutte le animazioni sono al posto giusto e lo spirito del coin op è riprodotto con molta cura.

Il titolo, amatissimo da generazioni di videogiocatori, non ha bisogno di tante presentazioni.

Ripasso rapido: lo scopo del gioco è sempre quello di completare i livelli composti di scale a pioli e ripiani su cui giacciono enormi strati di hamburger. Gli strati sono sistemati verticalmente, in modo da cadere



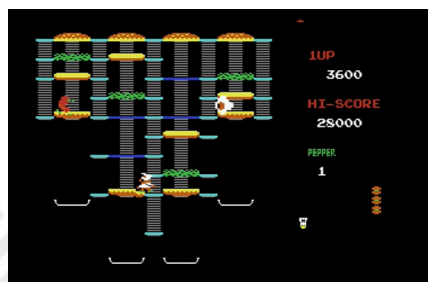
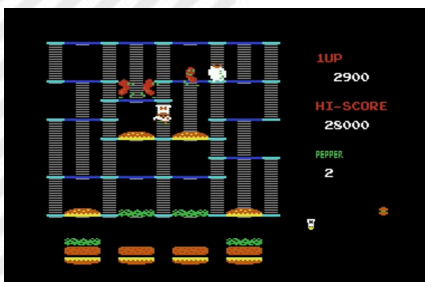
direttamente l'uno sull'altro.

Il giocatore guida il paffuto cuoco Pepper, che camminando sopra gli strati li farà cadere in basso e comporrà così il panino completo. Il tutto eludendo i nemici bislacchi che si trovano nei livelli: Mr. Hot Dog, Mr Pickle e Mr Egg.

Il titolo è scaricabile da qui: <https://arlagames.itch.io/burger-time> e vale la pena di essere collezionato nella nostra ludoteca.

Immortale!

di Carlo Nithaiah Del Mar Pirazzini



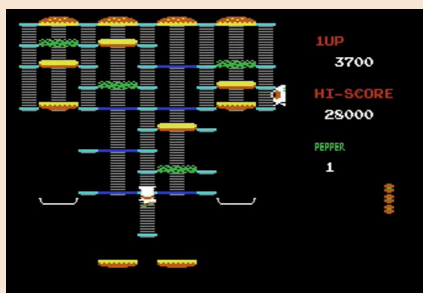
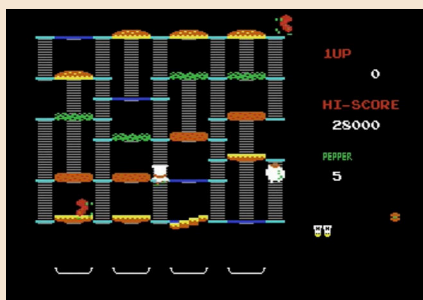
Anno: 2023

Editore/sviluppatore: Data East/  
Arlasoft

Genere: Action/Arcade

Piattaforma: Commodore 64

Sito web: <https://arlagames.itch.io/burger-time>



## GIUDIZIO FINALE



## » Giocabilità 90%

Un classico del genere perfetto in questa sua versione.

## » Longevità 90%

Anche qui, se vi fate prendere la mano difficilmente smetterete.







# DRACULA'S CASTLE

**Anno:** 2017

**Editore:** MP Software

**Sviluppatore:** Marco Pistorio

**Genere:** Platform

**Piattaforma:** Commodore 64

Anche questo Halloween si è concluso all'insegna di videogiochi, colonne sonore e immagini vecchie e nuove grazie ai sessantaquattisti che, malgrado tutto, nel 2023 rivivono quegli anni d'oro e non hanno mai detto di no alle loro (e nostre) passioni. Ero convinto di prendere in mano un gioco trito e ritrito quest'anno, magari anche già recensito su decine di riviste, e invece no! Qualche giorno prima di Halloween il destino ha voluto che vedessi un gameplay di un gioco di soli sei anni fa e passato inosservato a molti.

Si tratta di Dracula's Castle, gioco a piattaforme a schermata fissa in cui il protagonista, il cacciatore di fantasmi di turno, si trova in una casa infestata da pipistrelli, vampiri ed il conte Dracula in persona. Uscito nel 2017 grazie al lavoro diligente dei nostri amici retro appassionati (in questo caso il nostro collega redattore Marco Pistorio, ndFF) che ancora oggi programmano videogiochi per le macchine a 8-bit, in stile leggermente moderno ma sempre meritevoli.

Tutto ciò che dobbiamo fare, oltre ad evitare i nemici assetati del nostro sangue, è recuperare una chiave che permette di aprire la porta e giungere alla successiva stanza ed una volta superate tre stanze giungeremo infine all'uscita del castello. Il tutto non è paragonabile ad una gita scolastica in un castello ottocentesco; qui i vampiri sono veri e non sarà facile arrivare in fondo evitandoli come fossero persone che non vi stanno simpatiche.

Per fortuna ci sono le croci che ci rendono invincibile per un breve periodo di tempo. Ho detto bene,

breve! Breve potrebbe essere anche il gioco con sole tre stanze; molti potrebbero osservare che difficilmente un enorme castello conterrebbe solo tre stanze... La longevità del gioco è sicuramente incentrata sulla giocabilità semplice e sulla suspense dettata dalla colonna sonora: toccata e fuga di Johann Sebastian Bach.

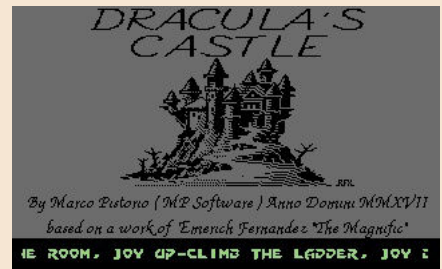
Il comparto grafico ricorda molto quello degli anni '80 e presenta sprite un po' piccoli, ma comunque fedeli ai personaggi. La componente strategica per il raggiungimento della chiave, l'uso delle croci di invulnerabilità e l'aggiramento dei nemici sarà fondamentale per arrivare in fondo al gioco. C'è una sola pecca, ossia i nemici che sbucano all'improvviso dalla parte destra o sinistra delle pareti; comunque nulla di grave e non occorrerà di certo un'infinita pazienza per terminarlo.

Questo bel titolo non è solo da giocare durante le notte più mostruose dell'anno, bensì ogni volta che volete fare una partitina veloce anche in piena estate o durante un temporale notturno, insomma un titolo che di certo non prenderà mai polvere.

Non penso che ci sia bisogno di forire consiglio per terminare il gioco, l'unica cosa che posso dirvi è godetevi! Siamo appena entrati nella stagione fredda, quella con copertina, tè caldo, serie tv e per ultimi, ma non per importanza, i videogiochi retro!

Niente di meglio di una fredda serata con pioviggella, toccata e fuga di Bach e Dracula's Castle!

di **Daniele Brahimi**



## GIUDIZIO FINALE

» **Giocabilità 80%**  
Controllo semplice, lineare.

» **Longevità 50%**  
Solo tre livelli, ma suggestivo e mai ripetitivo!



## Nuove idee all'orizzonte!

Che ricchezza questo numero 45!

Hardware, software, programmazione, listati, interviste, recensioni di giochi, libri e chi più ne ha più ne metta. Ad oggi possiamo affermare, senza tema di smentita, che RetroMagazine World è una delle riviste più complete nel panorama del retrocomputing.

Quando tutto iniziò, 6 anni fa, non avremmo mai creduto di poter arrivare a pubblicare 45 numeri in italiano e ben 21 in lingua inglese.

Tutto questo, vale sempre la pena ricordarlo, sempre in forma gratuita, perché noi di RMW siamo spinti solo dalla nostra passione e non dalla necessità di fare soldi; e questo ci mette in una posizione di privilegio!

Non abbiamo scadenze da rispettare, non dobbiamo rendere conto a nessuno e di conseguenza non subiamo nessuna pressione e nessuna spada di Damocle pende sulla nostra testa.

Che vuol dire tutto questo?

Vuol dire che possiamo documentarci, sperimentare e raccontare gli argomenti che noi stessi scegliamo, senza nessuna imposizione né vincoli temporali che potrebbero mettere fretta al nostro lavoro.

Se un articolo è pronto, lo pubblichiamo. Se non lo è lo pubblicheremo nel numero successivo o quando l'autore riterrà che sia il momento giusto per farlo.

Perché vi racconto tutto questo? Perché voglio farvi comprendere le dinamiche della nostra redazione e dimostrarvi come i redattori siano padroni del loro tempo e gestiscano gli articoli in totale autonomia. Questa sorta di 'anarchia' funziona sin dall'inizio della nostra avventura e credo che sia un po' il nostro marchio di fabbrica.

Senza voler apparire presuntuosi, un po' come negli ambienti accademici, dove i giovani ricercatori sono lasciati liberi di sperimentare e seguire le proprie intuizioni alimentando la creatività di tutto il gruppo, anche qui da noi ognuno è libero di suggerire e proporre nuove idee.

Un paio di queste nuove idee stanno prendendo piede proprio nelle pagine di questo numero. Vediamo se siete in grado di identificarle.

E voi? Avete idee o spunti interessanti da suggerire?

Fatevi avanti, noi siamo sempre pronti ad ascoltarvi e, se possibile, mettere in pratica i vostri suggerimenti!

**Francesco Fiorentini**

## Disclaimer

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

RetroMagazine World viene concessa al pubblico con licenza: Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale (CC BY-NC-SA 4.0 INT) <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it>

In pratica sei libero di: condividere, riprodurre, distribuire, comunicare o esporre in pubblico rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato, modificare, rielaborare, trasformare il contenuto e basarti su di esso per altre 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 farlo in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o l'utilizzo del materiale da parte tua.

### NonCommerciale

Non puoi utilizzare il materiale per scopi commerciali.

### StessaLicenza

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

Il licenziante non può revocare questi diritti fintanto che tu rispetti i termini della licenza.

### 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 World**  
Anno 8 - Numero 45 - NOVEMBRE 2023

**Direttore Responsabile**

*Francesco Fiorentini*

**Vice Direttore**

*Marco Pistorio*

**Coordinatore Redazione/Editing**

*David La Monaca*

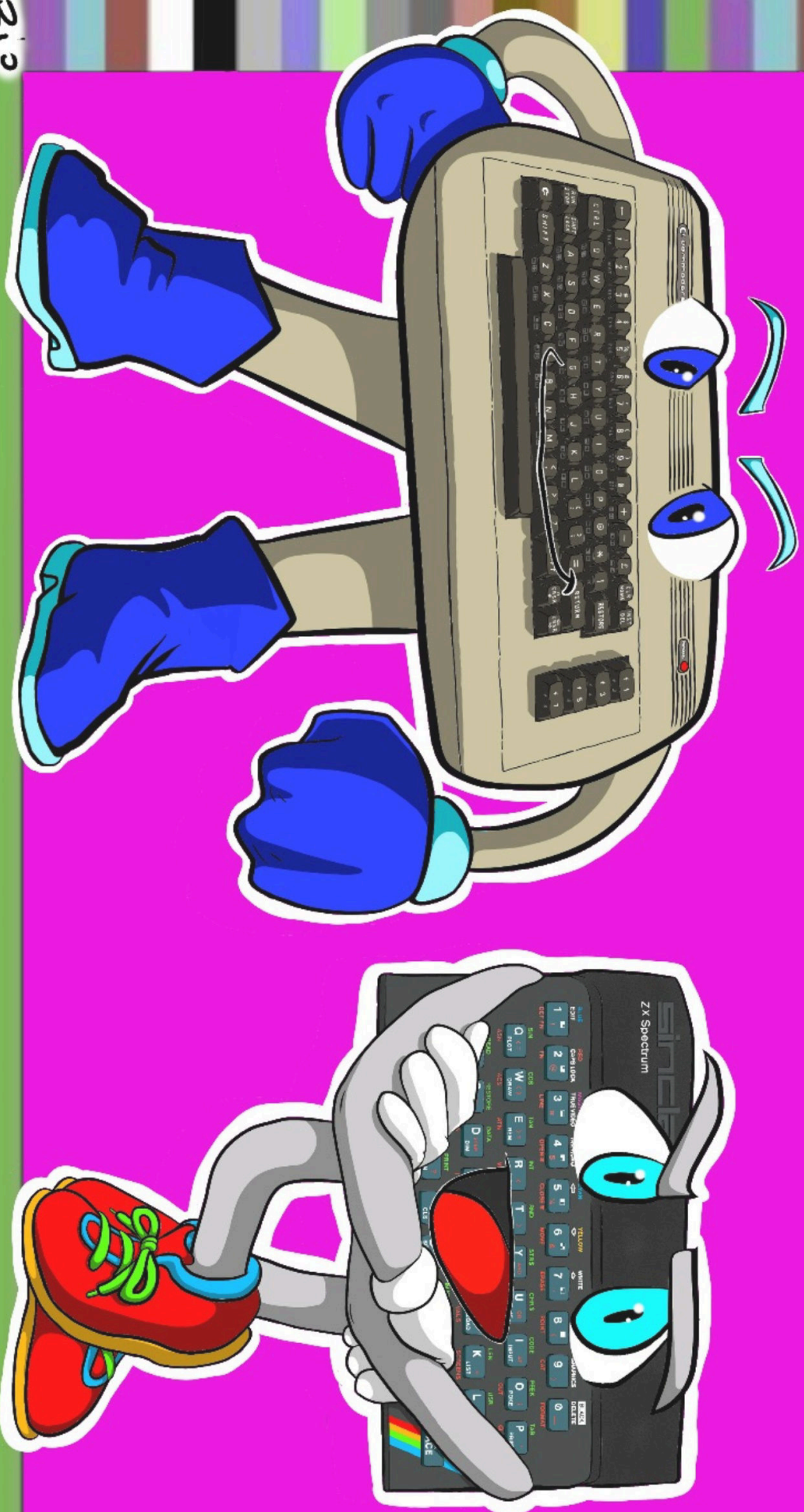
**Responsabili Area Web**

*Giorgio Balestrieri*





# CI VEDIAMO NEL PROSSIMO NUMERO



FABIO  
DE  
RENZIS

IL DINAMICO DUO! RETROMAGAZINE

