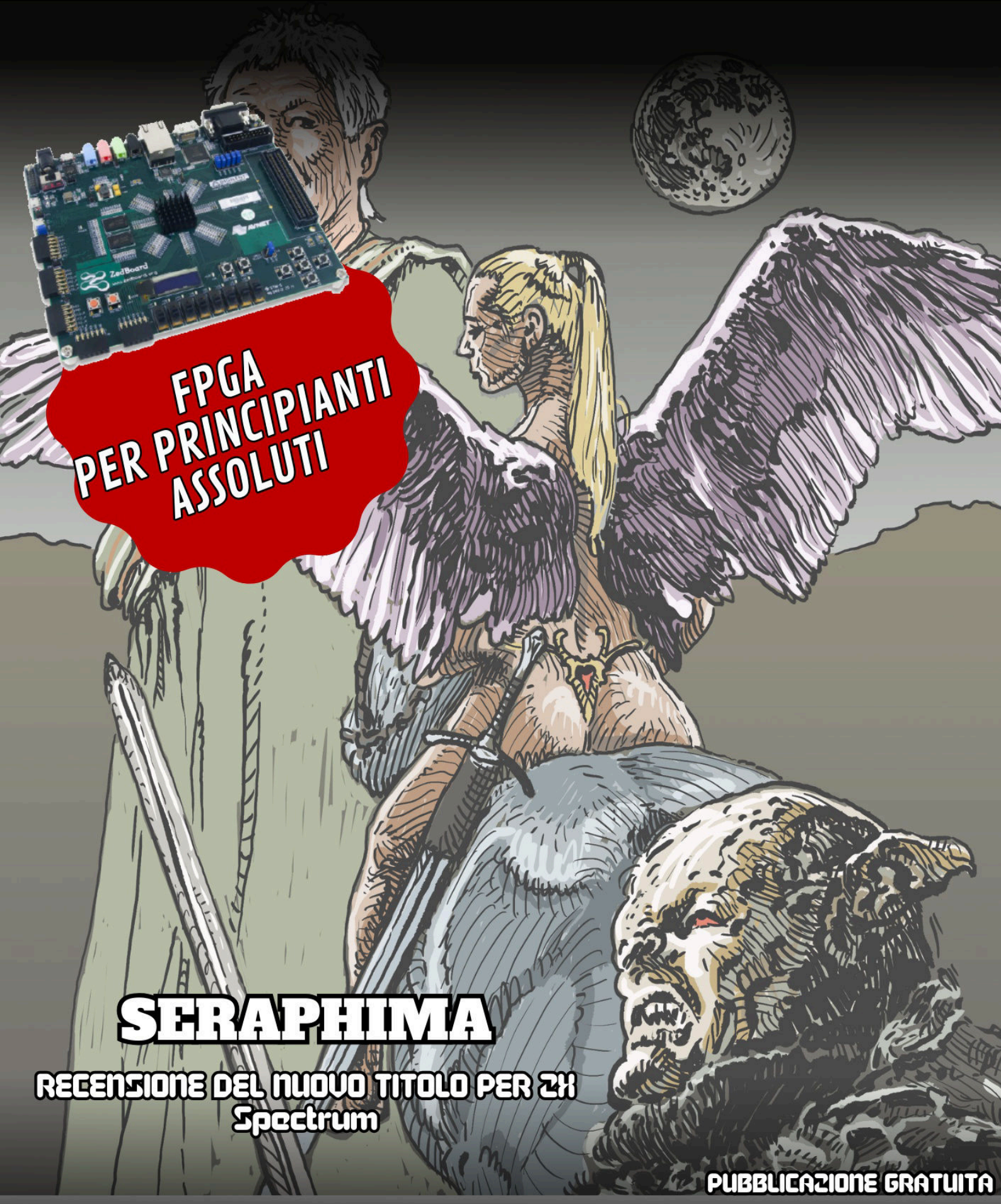


# RETROMAGAZINE



NUMERO 46 - FEBBRAIO 2024 - WWW.RETROMAGAZINE.NET



**FPGA  
PER PRINCIPIANTI  
ASSOLUTI**

## SERAPHIMA

**RECENSIONE DEL NUOVO TITOLO PER ZX  
Spectrum**

**PUBBLICAZIONE GRATUITA**

## Numeri

I numeri del movimento retrocomputing e retrogaming hanno ormai raggiunto livelli davvero notevoli. Il numero degli appassionati cresce in continuazione ormai da diversi anni e con la platea sempre in aumento sono molte le iniziative (commerciali e non) che nascono e si sviluppano per intercettare le richieste e i gusti del pubblico. Molte sono anche le iniziative editoriali volte a rinverdire i fasti delle riviste degli anni '80 e '90, allora unica vera fonte di informazioni e di aggiornamenti per tutti coloro che si avvicinavano al mondo degli home computer o delle console da gioco.

Nel 2024 gli strumenti per diffondere informazioni in modo efficace e rapido non mancano e tutto lascerebbe pensare che social media, blog, forum ed altre forme di comunicazione digitali presenti sulla Rete siano la naturale area di incontro fra chi cerca le ultime news sul mondo retro-hardware e software e chi s'impegna a ricercare, raccogliere, elaborare e pubblicare sistematicamente le notizie più interessanti e i dettagli storici sulle tante piattaforme a 8 e 16 bit del passato.

Le iniziative editoriali che oggi chiunque si avvicini al mondo del retrocomputing (e vi includo anche quelli interessati solo al retrogaming) trova sulla Rete sono numerose e multiformi: dal blog del singolo appassionato al forum monotematico, dal generico gruppo Facebook ai profili dedicati di Instagram o X, dagli affollati canali Telegram ai siti web specializzati, dagli account Youtube ai canali interattivi su Discord o Twitch. L'offerta è davvero ricchissima, forse superiore all'effettiva domanda.

Poi ci sono le imprese che strizzano l'occhio nostalgico alle riviste storiche del passato e che in qualche modo cercano di omaggarlo. RetroMagazine World si onora di far parte di questa categoria e percorre la strada tracciata da una lunga tradizione di pubblicazioni spontanee, non ufficiali o semi-ufficiali che si ispira alle e-zine amatoriali underground degli anni '90 e alle leggendarie riviste italiane ed europee uscite nei primi anni '80 ad anticipare e poi trainare il successo e la diffusione degli home e dei personal computer.

Negli ultimi 4 anni abbiamo anche assistito all'arrivo di alcune pubblicazioni cartacee (mono-piattaforma o "generaliste") che vengono distribuite tramite abbonamento o attraverso le edicole, ricalcando molto da vicino le dinamiche delle riviste informatiche di un tempo, con tutto ciò che la cosa comporta in termini di costi, solitamente ingenti quando c'è di mezzo la stampa e la distribuzione del media fisico. Queste edizioni sono a volte il frutto di una reale passione per questo nostro mondo, a volte si tratta di mere speculazioni che mirano a cavalcare l'onda del rinnovato interesse del pubblico verso macchine, console e videogiochi del passato.

Distinguere tra le varie iniziative editoriali è facile. Altrettanto facile decidere di supportare quelle più meritevoli, in base al taglio dei contenuti, alla qualità e all'impegno visibile. Noi di RMW crediamo in una formula "aperta". Aperta nei confronti di tutti i contributi che possono arrivare dai tanti esperti veri e sinceri che si nascondono qua e là nelle pieghe dei social media o dei forum dedicati alle varie piattaforme informatiche e videoludiche. Aperta alla collaborazione con altri gruppi di appassionati e con le redazioni di altre pubblicazioni. La storia di 46 numeri in italiano e 23 in inglese testimonia nei fatti la cooperazione attiva con ogni tipo di proposta o suggerimento che arrivi dall'esterno. Sappiamo per certo che questo nostro approccio funziona e che riesce a raccogliere attorno ad RMW una quantità sempre crescente di sostenitori e di seguaci. Lo dicono i numeri.

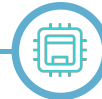
## SOMMARIO

◇ FPGA per principianti assoluti - parte 2	Pag. 3
◇ Il Fortran sul Commodore 64 - integer e float	Pag. 10
◇ Power C - il miglior compilatore C per C64/128?	Pag. 12
◇ Libdragon Development - parte 2	Pag. 17
◇ Guida al SGDK - parte prima	Pag. 20
◇ Casi di identità... - parte I	Pag. 23
◇ PunyInform, una nuova libreria...	Pag. 26
◇ Omaggio a Doriath (Merlino)	Pag. 29
◇ Dracula X (Saturn)	Pag. 34
◇ Briley Witch Chronicles 2 (C64)	Pag. 36
◇ Yetri Mountain (C64)	Pag. 38
◇ Corescape (C64)	Pag. 40
◇ Magical Pop'n (Snes)	Pag. 42
◇ The Adventure of Little Ralph (Playstation)	Pag. 44
◇ Hunter Girls (MegaDrive)	Pag. 46
◇ Seraphima (ZX)	Pag. 48
◇ Hermano (GameBoy)	Pag. 50
◇ Tetris 1200 (Amiga 1200)	Pag. 52
◇ Breaktru (Amiga)	Pag. 54
◇ Final Fight Enhanced (Amiga)	Pag. 56
◇ Dottie Flowers (Snes)	Pag. 58
◇ Turbo Outrun (Plus4)	Pag. 59
◇ Pac-Man Arcade (ZX)	Pag. 60
◇ The Mandarin II (Amstrad CPC)	Pag. 61
◇ Sergio Kidd (GB Color)	Pag. 62
◇ Duck Hunt (Amiga)	Pag. 63
◇ Keystone Kapers (C64)	Pag. 64

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

- Dave Nardella
- David La Monaca
- Carlo N. Del Mar Pirazzini
- Daniele Brahimi
- Francesco Fiorentini
- Francesco Donatini
- Roberto Del Mar Pirazzini
- Giampaolo Moraschi
- Giuseppe Rinella
- Fredrik Ramsberg
- Ingrid Poggiali
- Eugenio Rapella
- Takahiro Yoshioka
- Marta Rossmann
- Gianluca Girelli
- Marco Pistorio
- Immagine di copertina: **Giuseppe Mangini**
- Layout di copertina: **Carlo N. Del Mar Pirazzini**
- Immagine di chiusura: **Fabio de Renzis**



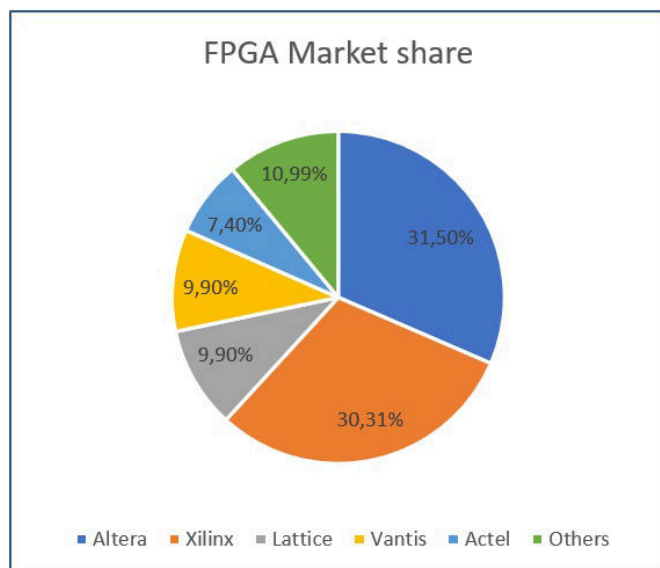


# FPGA per principianti assoluti: partiamo da zero - seconda parte

di Dave Nardella

## Brand e famiglie

Il mercato degli FPGA vede due grandi player, Xilinx, acquisita da AMD e Altera, acquisita da Intel, i quali si dividono equamente circa il 60% del volume d'affari complessivo. Nel rimanente 40% troviamo Lattice, Vantis, Microchip, Achronix, Gowin ecc.



Se deciderete di dedicarvi agli FPGA, le prime applicazioni non banali vi costringeranno a scegliere la vostra “religione”, la quale sarà, quasi sicuramente, Xilinx o Altera.

Entrambi hanno un catalogo molto ricco, da FPGA entry-level fino a chip military-grade.

Ognuno propone quattro o cinque famiglie caratterizzate da una “tendenza” di utilizzo, all’interno di queste famiglie ci sono vari modelli di FPGA che si differenziano per le dotazioni: numero di CLB, quantità di RAM, numero di blocchi DSP ecc.

I loro tools di sviluppo sono completi e costantemente aggiornati, entrambi forniscono una versione lite free per programmare FPGA medio-piccole. Non è una grossa limitazione, considerate ad esempio che con Quartus II Lite di Intel è possibile gestire la famiglia Cyclone V, che vi assicuro è molto potente.

Anche in questo caso troverete accaniti sostenitori di uno o l’altro brand. Personalmente non vedo grandi disparità di catalogo o prezzi che possano far preferire, tout-court, l’uno o l’altro.

Si ripropone la vecchia guerra Intel/AMD, ma in questo caso combattono ad armi pari.

## Approccio consigliato

Le FPGA sembrano interessanti, di cosa ho bisogno per cominciare? Se poi mi accorgo che non soddisfano i miei interessi, avrò sprecato molto denaro?

L’utilizzo degli FPGA è rivolto al mercato professionale, in genere i sistemi di sviluppo tendono a costare più di quanto è abituato a spendere un hobbista/maker, ma fortunatamente non è sempre così, esistono in commercio dei sistemi didattici entry-level per tutte le tasche, bisogna saper scegliere.

Però, prima ancora di dotarsi di un sistema di sviluppo, è importante capire l’approccio giusto a questi sistemi. Quando acquistiamo Arduino, colleghiamo il nostro display I2C, scarichiamo un paio di librerie e dopo qualche minuto e poche righe di codice visualizziamo il nostro “Hello World”. Con un FPGA è diverso, non abbiamo interfacce I2C, SPI o video, dobbiamo crearle noi istanziando degli IP; all’inizio del percorso, il risultato più eccitante sarà quello di far lampeggiare un led.

Dico questo, non per scoraggiarvi, ma per evidenziare che è possibile realizzare tutto, ma bisogna procedere a piccoli passi.

I miei consigli per partire con il piede giusto sono:

1. Acquistate una board FPGA molto economica.
2. Evitate i demo e partite con un buon libro Verilog/VHDL, semplice ma completo.

Ricordate che tutto quello che realizzerete utilizzando un libro, che vi assicuro che non è poco, potrete portarlo su FPGA di qualsiasi brand.

Una volta acquisite le conoscenze base sarete anche in grado di valutare il passo successivo, ovvero di quale board più performante avete bisogno; all’inizio è facile acquistare options che non servono subito o non serviranno affatto.

Il problema dei demo ufficiali è che sono sempre datati; una volta caricati nelle nuove versioni dei sistemi di sviluppo, vengono emessi tanti warning, alcuni dei quali invalidanti, ed è difficile gestirli senza le conoscenze necessarie.

Infine, se purtroppo vi accorgete che gli FPGA proprio





non vi interessano, avrete investito meno del costo di una cena.

Prima di partire con i consigli per gli acquisti, vediamo velocemente le differenze fra le varie famiglie di board.

Sostanzialmente i sistemi FPGA che trovate in commercio si dividono in tre categorie, la divisione non è netta, a volte può essere difficile catalogare qualche board; queste sono:

1. Sistemi di sviluppo
2. Sistemi didattici
3. Breakout Boards

I primi servono a testare le funzionalità di una famiglia, contengono RAM dinamica, periferiche integrate quali Ethernet, circuiteria analogico/digitale, ed un numero elevato di connettori I/O. Questi sistemi sono dedicati ad un utilizzo professionale/evoluto, partono da qualche centinaio di euro fino ad arrivare anche 70.000 euro. Il concetto di utilizzo è il seguente: devo realizzare una board ed ho selezionato la famiglia più adatta. Utilizzo un sistema di sviluppo che contiene il chip più potente di quella famiglia per sviluppare il mio programma. Quando funziona tutto, vedo quante risorse occupa il mio programma; quindi per la mia produzione seleziono, nella famiglia, il modello più adatto: capace, ma non troppo.

I sistemi didattici hanno a bordo pulsanti, led, interruttori, display a 7 segmenti, buzzers e quant'altro può servire per testare i primi programmi avendo a disposizione tutto già collegato. Qui i costi sono più bassi, però rimaniamo nella fascia 100-300€ a seconda del modello di FPGA che montano.

Infine, ci sono le Breakout Boards che sono sistemi didattici a basso costo che non hanno pulsanti displays e led, o ne hanno pochissimi; sostanzialmente sono costituiti da un FPGA e poco più, al quale sono stati collegati i pin direttamente ai connettori di interfaccia.

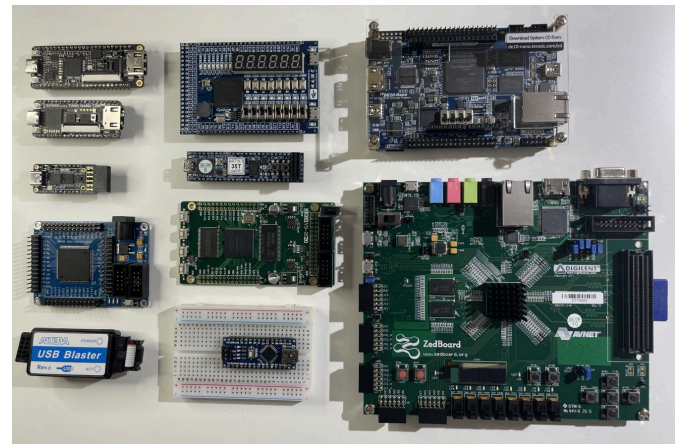
Il vantaggio è che in genere possono essere inseriti in una BreadBoard. Il principio di utilizzo è simile a quello di Arduino Nano, moduli ESP32 ecc. Con qualche euro e un po' di manualità possiamo collegare pulsanti e led a piacimento. Per un utilizzo didattico entry-level queste boards sono perfette, ricordate che l'FPGA già contiene tutto quello che ci serve.

### Consigli per l'acquisto

Vi elenco brevemente i modelli che ho in casa, in ordine di prezzo a partire dal più basso (che ovviamente possono variare in base al fornitore), e qualche altra board che

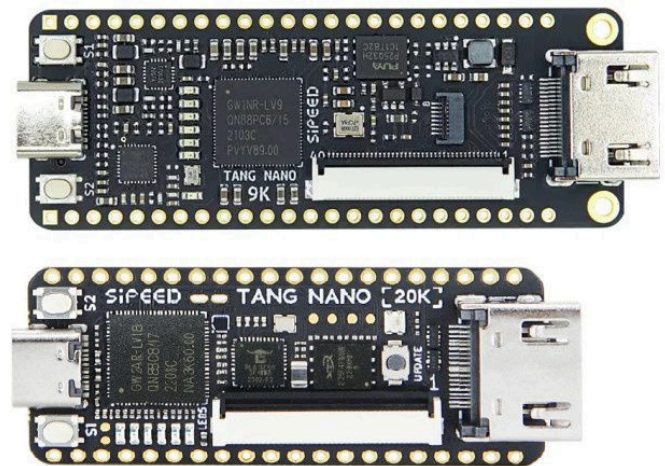
non possiedo ma che conosco abbastanza bene, evidenziando anche in quale step di apprendimento conviene utilizzarli. Mi soffermerò di più solo sul secondo modello, che è quello che vi consiglio per iniziare.

In figura ho inserito anche un Arduino Nano su una mini-Breadboard per dare l'idea delle dimensioni.



### Sipeed Tang Nano 9k e Tang Nano 20k

Sono due board prodotte da Sipeed, la seconda è l'evoluzione della prima, la differenza è di pochi euro ma offre molto di più.



Questo è il modello che vi consiglio per una serie di motivi:

1. Monta un chip abbastanza interessante, il Gowin GW2AR-18 (20,737 CLB, RAM, DSP, PLL e Flash) che, oltre ai primi esperimenti, permette di ospitare un softcore RISC-V e di far girare una micro-distribuzione di Linux. Ci sono alcuni progetti retrogame realizzati per questa piccola board.
2. Ha una dotazione hardware interessante: molti pin di I/O, un connettore dedicato per schermi LCD, un connettore HDMI, il socket per una micro-SD, un'interfaccia USB utilizzata sia per la programmazione che per la comunicazione RS232, un connettore speaker, sei led e due pulsanti user.
3. Il sistema di sviluppo Gowin è molto leggero e semplice da usare, non contiene molte opzioni avanzate, le quali al contrario, in una prima fase potrebbero essere fuorvianti. Questa board può essere programmata con la versione lite.





4. È possibile far girare, direttamente o con insignificanti modifiche, tutti gli esempi presenti per la famiglia Tang, che sono tantissimi.

È possibile acquistarla dal negozio ufficiale Sipeed su Aliexpress, a circa 34€ (Tang Nano costa circa 19€) ma è disponibile in molti altri store online.

Qui trovate il wiki ufficiale (di tutta la serie Tang) ed i link a tutorials ed esperimenti.

### ICESugar-nano

È una delle board più piccole, monta un iCE40LP1K-CM36 di Lattice e viene prodotta da MuseLab (attivate il traduttore del browser). Costa poco, circa 22€.



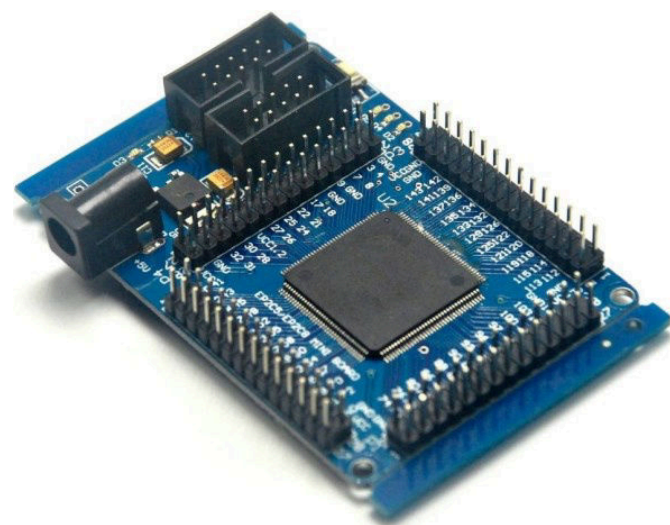
La particolarità di questa board è che è completamente open-source, e lo è anche il sistema di sviluppo, IceStorm, che permette di programmare tutta la famiglia iCE40. Non è potentissima ma permette di ospitare dei micro-softcore.

È possibile programmarla anche con la versione Lite di iCEcube2 di Lattice.

Qui trovate il wiki: <https://github.com/wuxx/icesugar-nano>

### Cyclone II EP2C5 mini Dev Board

La nascita di questa board è avvolta nel mistero. Non ha un produttore ufficiale, è possibile trovarla a pochi euro in quasi ogni store online.



Monta un chip abbastanza vecchiotto Cyclone II EP2C5T144C8 di Altera, per programmarlo è necessario scaricare una vecchia versione Quartus II, non oltre la

V13 (è comunque presente sul sito Intel).

Il vantaggio è che è una board estremamente diffusa ed esiste una documentazione sterminata in rete. Non è possibile programmarla direttamente, è necessario utilizzare un adattatore USB-Blaster, in genere lo si trova in bundle con la board o separatamente per pochissimi euro.

Questo è un buon punto di partenza:

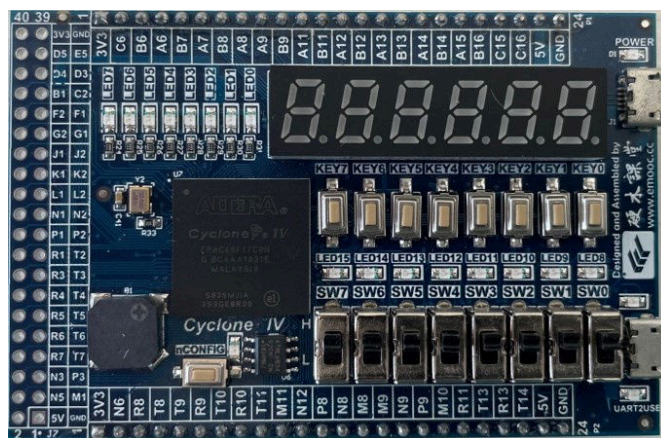
[https://land-boards.com/blwiki/index.php?title=Cyclone\\_II\\_EP2C5\\_Mini\\_Dev\\_Board](https://land-boards.com/blwiki/index.php?title=Cyclone_II_EP2C5_Mini_Dev_Board).

Anche su Github è possibile trovare molti esempi:

<https://github.com/search?q=EP2C5&type=repositories>

### Mini FPGA Cyclone IV Board

Questa board è molto interessante, costa poco (circa 37€) monta un Altera Cyclone IV EP4CE6, ha otto pulsanti, otto led, otto interruttori 6 display a sette segmenti, un buzzer e due micro-USB (non necessita del programmatore esterno). Può essere programmata con l'ultima versione di Quartus II Lite.



Ve la consiglio come acquisto preferenziale solo se avete un po' di spirito di avventura.

Lo scopo di questa scheda è essenzialmente didattico, sul wiki del produttore ci sono tantissimi esempi di difficoltà crescente ben calibrata.

Il problema è che il sito è solo in lingua cinese, ma vi assicuro che il traduttore del browser fa un ottimo lavoro. L'unico problema è rappresentato dalle scritte nelle immagini, ma non sono molte.

Purtroppo, se volete gli esempi, che sono liberamente scaricabili, da utilizzare direttamente nella board, dovete interfacciarvi con baidu, il quale non è accessibile (come registrazione) da tutti i paesi per un banale problema di bug delle pagine html.

Considerate però che, quando si lavora con gli FPGA, l'unico documento assolutamente indispensabile è lo schema elettrico per ottenere la numerazione dei pin collegati a pulsanti e display, in questo caso può essere





richiesto al fornitore e viene inviato per mail immediatamente, il resto non è essenziale.

Io l'ho trovata molto divertente, è compatta e funziona benissimo.

Qui trovate gli esperimenti:

<https://www.yuque.com/yingsmuketang/01/ri3fkv>

Questo è il produttore:

<http://www.emooc.cc/>

La board la trovate su Aliexpress.

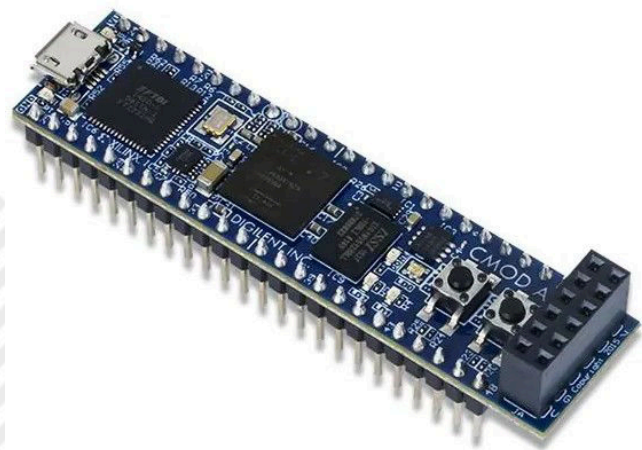
Le boards a partire da questo punto ve le consiglio come acquisto solo in un successivo step di preparazione perché il costo aumenta; non sono cifre esagerate, ma il mio obiettivo è farvi spendere solo la cifra necessaria per partire, imparare le basi e qualcosa in più.

A voi comunque la scelta finale, sono sicuramente degli ottimi prodotti.

### Digilent Cmod A7-35T

È una breakout board ma monta uno Xilinx Artix-7 XC7A35T, un mid-range abbastanza performante che ha 33,280 CLB, 1,800Kb di blocchi RAM, 90 DSP e 5 Transceiver a 6,6 Gb/s. Questa board è provvista di 512MB di SRAM e permette di ospitare MicroBlaze con performances di 303 DMIPs o altri softcore senza problemi.

Il vantaggio interessante, dato dalle sue dimensioni molto ridotte, è che è possibile utilizzare questa board direttamente nelle proprie realizzazioni finali montandola su una scheda ospite che potrebbe contenere la circuiteria ausiliaria di interfaccia.

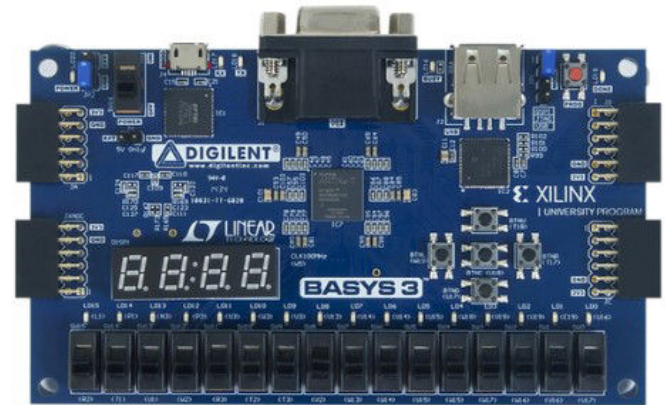


La famiglia Artix 7 è più energivora della cugina Spartan 7; quindi, sinceramente mi aspettavo che scaldasse di più, in realtà la temperatura, con il 70% della capacità impegnata (oltre non è consigliabile spingersi), si mantiene a livelli accettabili. Per applicazioni critiche consiglio, comunque, di utilizzare un mini-dissipatore.

Il suo costo si aggira intorno ai 111€.

### Digilent Basys 3

Ha lo stesso chip di Cmod A7 35-T ma è una board orientata alla didattica, infatti presenta 16 interruttori, 5 pulsanti, 4 display a 7 segmenti ed un'uscita VGA a 12 bit come potete vedere dalla foto.

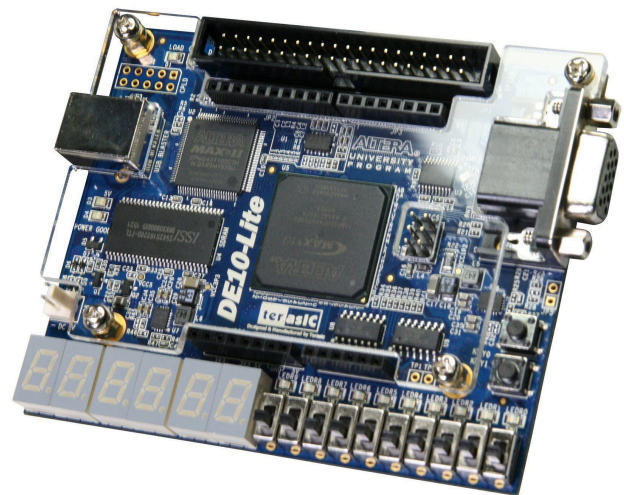


È un prodotto storico e molto diffuso, esistono interi siti dedicati a lei e tanti esempi. La trovate a circa 190€.

Questa è la sua pagina ufficiale: <https://digilent.com/reference/programmable-logic/basys-3/start>

### Terasic DE-10 Lite

Qui siamo nel mondo Intel-Altera. Questa board monta un MAX 10 10M50DAF484C7G che contiene 50,000 CLB, 1,638 Mbit di RAM, 144 DSP a 18 bit e 4 PLL.



L'FPGA è di fascia medio-bassa, praticamente una versione ridotta della famiglia Cyclone, ma la board ha una ricca dotazione di interruttori, display ed interfacce ad un prezzo onesto di circa 162€.

Ha i connettori di espansione compatibili Arduino, non è l'unica board ad averli, però attenzione, tutti gli FPGA lavorano max a 3,3V quindi gli shield pensati per i 5V possono danneggiare la scheda. Nel caso, orientatevi su shield compatibili con Arduino 2 o Arduino GIGA R1 WIFI, o in alternativa, potete utilizzare tutte le interfacce pensate





per la serie ESP32 o per gli ARM (STM32, Renesas, ecc..). Supporta il softcore Altera Nios II, due istanze sicuramente. La trovate qui: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1021>

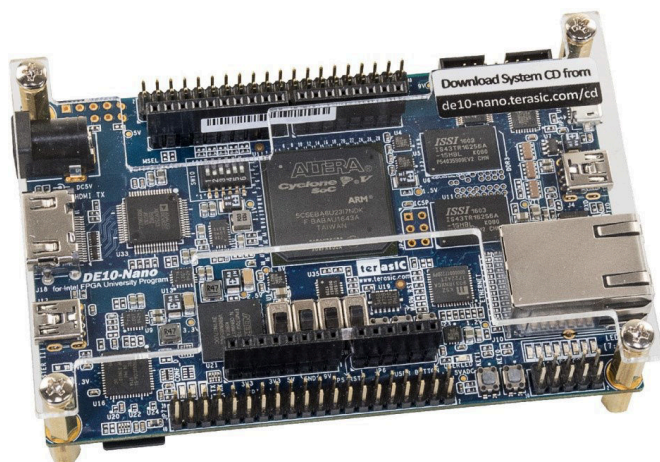
### Terasic DE-10 Nano

Questa è una bellissima scheda equipaggiata con un Cyclone V SE 5CSEBA6U23I7, è un SoC che contiene un ARM Cortex-A9 Dual core a 800MHz come HPS.

A questo link: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=1046&PartNo=1#contents>

potete approfondire le specifiche e la dotazione di interfacce che sono davvero ricche. L'unica nota, come in tutte le board contenenti SoC, le periferiche sono divise fra HPS e FPGA, ad esempio l'interfaccia Ethernet è collegata ad HPS ed invisibile da FPGA.

potete approfondire le specifiche e la dotazione di interfacce che sono davvero ricche. L'unica nota, come in tutte le board contenenti SoC, le periferiche sono divise fra HPS e FPGA, ad esempio l'interfaccia Ethernet è collegata ad HPS ed invisibile da FPGA.



Ha solo quattro interruttori, per cui non può essere considerata una board didattica, in compenso ci sono molti esempi in circolazione.

Attenzione però, non è una scheda semplicissima, il fatto di avere un FPGA SoC complica abbastanza il primo approccio; è necessario studiare molto bene il manuale. Utilizzarla senza l'HPS (i core ARM) è possibile ma non ha molto senso se non si pensa per lo meno di farlo in futuro. La programmazione di HPS di per sé non è complicata, ma mettere in comunicazione FPGA con HPS può provocare qualche mal di testa se non si procede con ordine.

È una board molto diffusa, anche grazie al progetto di retrogaming MisTer, per il quale però necessita di alcune schede aggiuntive.

Le considerazioni sui connettori Arduino sono le stesse fate per DE-10 Lite.

L'unico difetto è il pannello in plexiglass, inutile ma soprattutto nocivo perché limita la circolazione dell'aria.

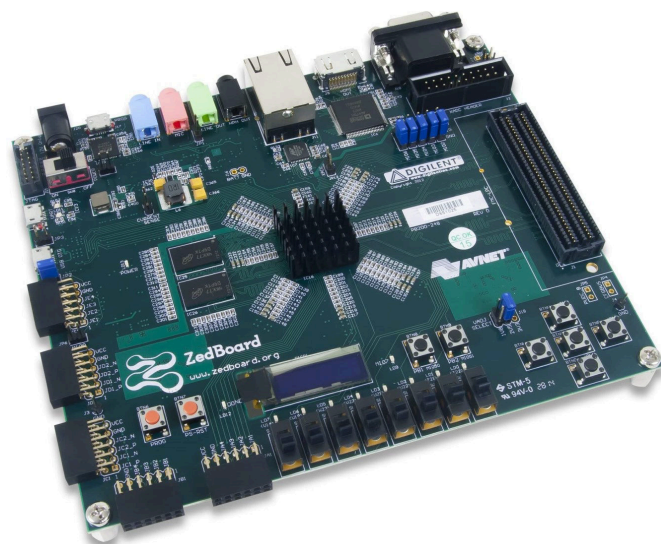
L'FPGA scalda molto anche "a vuoto". Secondo me sul chip va installato un dissipatore di rame ellittico low-profile e in questo caso, il pannello di plexiglass, opportunamente forato, può servire da supporto per una ventolina. Trovate DE10-Nano a circa 260€.

### Digilent ZedBoard

Anche questa è una board storica, il punto di riferimento per imparare lavorare con la famiglia Xilinx ZynQ. Monta un SoC XC7Z020 ed ha una dotazione impressionante di periferiche.

Viene pubblicizzata come low-cost development board anche se il suo costo è di circa 680€, considerando però l'FPGA installato e la dotazione, è comunque un prezzo adeguato a uno strumento semiprofessionale.

Qui trovate la pagina ufficiale: <https://digilent.com/shop/zedboard-zynq-7000-arm-fpga-soc-development-board/>.



Non ve la consiglio come primo acquisto, è abbastanza complicata da utilizzare se non si ha esperienza con il mondo Xilinx e l'utilizzo di IP avanzati.

L'HPS è un ARM Dual-core A9 a 667Mhz, Non ha senso utilizzare questa boards per piccole sperimentazioni, di default è installato Petalinux, una distro molto componibile derivata Yocto, purtroppo non è di utilizzo semplicissimo; per modificarlo/installarlo è necessario utilizzare l'SDK di Xilinx ed il BSP (Board Support Packages).

Il primo approccio con questa board è stato un po' ostico, fortunatamente in rete c'è molto materiale.

Se vi capita di trovarla di seconda mano è importante che vi venga trasferita la licenza Xilinx collegata alla board per poter utilizzare il sistema di sviluppo con questa famiglia di processori, con la versione lite non potete farlo.

### Risorse consigliate

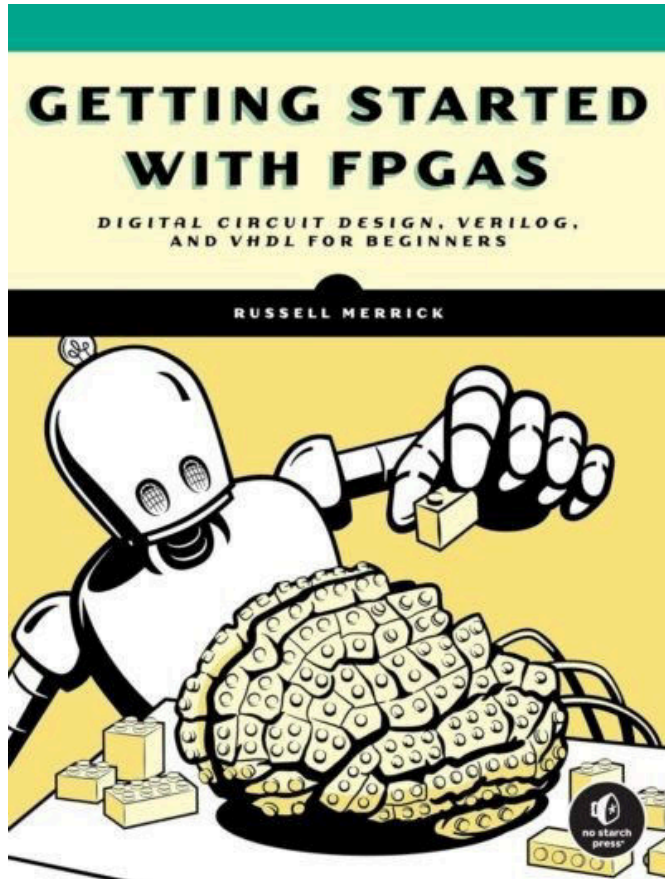


**Libri**

Io preferisco sempre libri “agnostici” cioè non troppo legati ad un Brand, alle specificità dei suoi strumenti di sviluppo ed ai suoi IP speciali. Vi consiglio qualche testo che ho letto, alcuni completamente, altri concentrandomi solo su alcuni argomenti.

Il libro, in assoluto, più semplice e concreto che ho trovato è senz'altro:

**Getting Started with FPGAs di Russell Merrick.**



Gli esperimenti vengono condotti su una piccola board low-cost contenente un FPGA Lattice, ma sono estremamente portabili ovunque, tratta sia Verilog che VHDL.

Si legge davvero bene e i concetti più complessi sono sempre approfonditi con degli esempi. Se avete un budget limitato (purtroppo i libri costano) e potete acquistarne solo uno, questo è quello che vi consiglio.

Trovate tutti i riferimenti nel sito di NAND LAND, che vi invito a visitare perché contiene tanti tutorials ed informazioni utili.

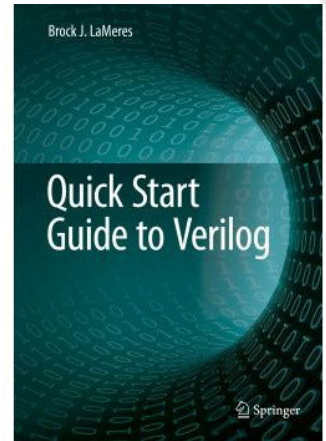
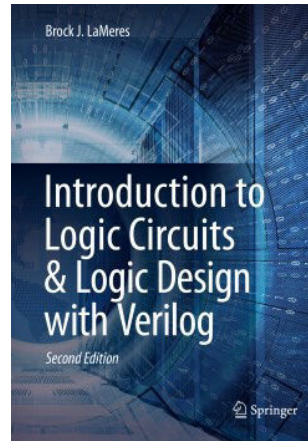
Se volete anche approfondire i concetti di elettronica digitale e logica booleana, potete acquistare:

**Introduction to Logic Circuits & Logic Design with Verilog di Brock J. LaMer.**

È molto completa e facile da seguire, questo autore scrive molto bene.

Dello stesso autore potete acquistare

**Quick Start Guide to Verilog, più focalizzato su Verilog.**



Se infine volete davvero imparare l'elettronica digitale, i microprocessori e la loro implementazione in VHDL vi consiglio due libri che potete trovare sia in italiano che in inglese.

**Introduction to Digital Systems Design**

**Introduction to Microprocessor -based Systems Design**

**Di Guido Donzellini e altri.**

(Si veda figura a pagina 9).

Gli esperimenti vengono condotti con deeds un editor/simulatore freeware scritto dallo stesso autore, molto semplice da usare ma completo che permette di esportare in VHDL i circuiti realizzati.

Questo è il sito in cui potete trovare i link al software, agli esempi ed ai libri in doppia lingua:

<https://www.digitalelectronicsdeeds.com/index.html>

Vi avverto però che sono testi universitari, io li ho trovati molto completi e non complicatissimi, però richiedono un percorso di un certo impegno. La possibilità di simulare visivamente i circuiti studiati è senz'altro un plus notevole.

**Tutorials**

Se preferite iniziare in modo “soft” con qualche tutorial, prima di acquistare uno o più libri, sulla rete ci sono un'infinità di risorse; probabilmente troppe, e questo, dato che sono gratis, potrebbe crearvi un'indigestione.

Vi consiglio solo due link, iniziate da questi e seguiteli completamente, alla fine sarete in grado voi stessi di selezionare le risorse più adeguate alle vostre esigenze.

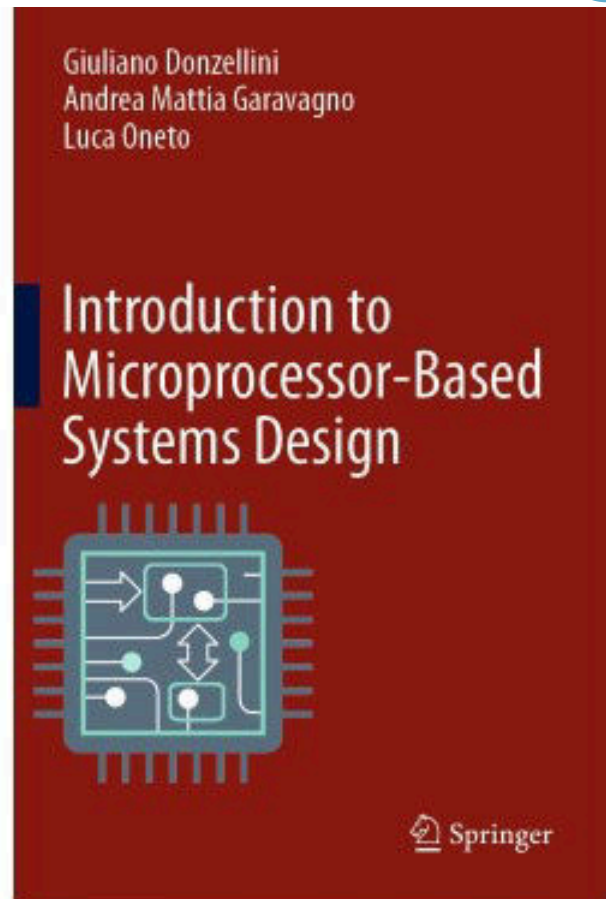
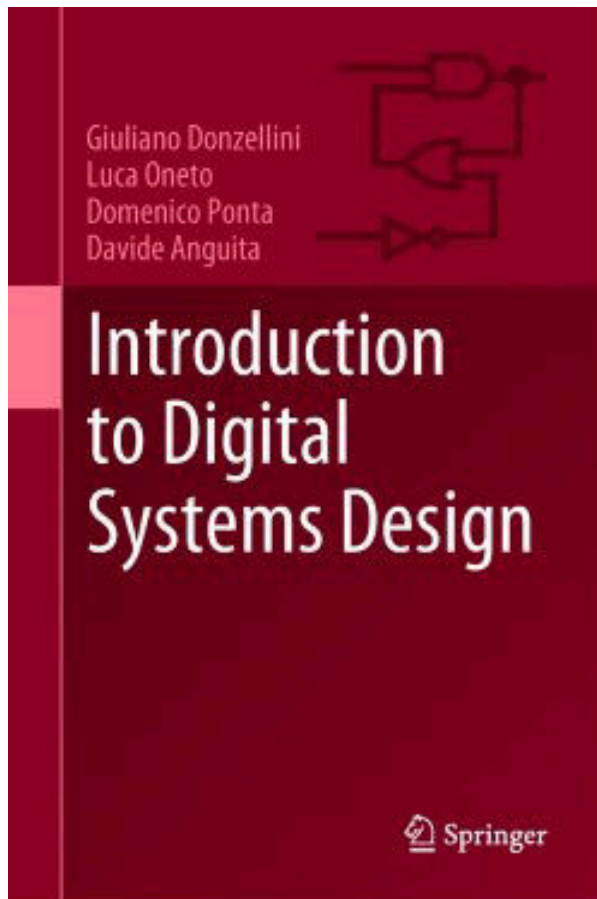
FPGA Tutorial: <https://fpgatutorial.com/>

NAND LAND: <https://nandland.com/>

Infine, nel sito di OpenCores potete trovare un'infinità di IP di esempio e softcore: <https://opencores.org/>







### Conclusioni

Siamo arrivati alla fine di questa chiacchierata, spero di aver stimolato il vostro interesse verso questi sistemi che secondo me sono parecchio longevi, nel senso che difficilmente vi stancherete perché, per quanto si scavi,

c'è sempre un altro livello di complessità da affrontare. Buon divertimento.





# Il Fortran sul Commodore 64 - integer e float

di Francesco Fiorentini

Nel numero scorso abbiamo visto come utilizzare l'Abacus Fortran-64 per scrivere, compilare, linkare ed eseguire un programma in linguaggio Fortran sul nostro amato Commodore 64. Siccome l'appetito vien mangiando ho voluto continuare i miei esperimenti con questo interessante linguaggio, soprattutto per capire se il compilatore del C64 potesse essere utilizzabile per un uso professionale o al limite semi-professionale.

Dato che il codice Fortran viene compilato, ho pensato immediatamente che fosse molto piú performante del bistrattato Basic V2... Ma da quello che vedremo, le sorprese non mancheranno.

## Che programma utilizzare

La mia esperienza con il Fortran è veramente poca cosa, quindi dovevo trovare il modo di scrivere un programma semplice, da replicarsi facilmente in Basic e Fortran, ma che allo stesso tempo mettesse alla frusta le capacità di calcolo dei due linguaggi. Idea: perché non scrivere un programma che sommasse tutti i numeri da 1 a 10000? Semplice, efficace e, così credevo, privo di errori.

La versione in Basic V2 è veramente banale, la riporto solo per dovere di cronaca:

```
10 a=0: f=10000
20 for i = 1 to f
30 a=a+i
40 next i
50 print "la somma dei primi"; f ;"numeri: ", a
```

La versione Fortran è invece leggermente più complicato, ma non troppo. Ci sono solo degli accorgimenti da adottare visto che dobbiamo fare attenzione ad utilizzare numeri float ed integer al posto giusto:

```
10 : program test
15 :* this program sum
16 :* the first 10000 numbers
20 : real f,a
25 : f=0.0
30 : open 3,3
```

```
40 : do 10 i = 1, 10000
45 : a=float(i)
50 : f=f+a
60 : 10 continue
65 : write (3,100) f
70 : close 3
75 : stop
80 : 100 format(f10.0)
90 : end
```

Nota: per editare, compilare e linkare il codice soprastante vi rimando all'articolo del numero precedente.

A questo punto non mi restava altro che eseguire i due programmi e... Non potevo credere ai miei occhi! Il codice Basic ritornava il risultato quando ancora il Fortran sembrava essere bloccato... Tra me e me pensavo: avrò fatto qualche errore nel codice, probabilmente le conversioni non sono corrette... Non potevo assolutamente pensare che il Fortran fosse così lento. Potrete sicuramente immaginare il mio stupore quando, dopo un lungo tempo di attesa il C64 su cui girava il Fortran ritornava il suo risultato: 49969900. Cavolo se era lento...

Ma, un momento. Il risultato ritornato dal Basic era 50005000, mentre quello del Fortran era 49969900. Chi diceva il vero?

Poiché la somma dei numeri interi da 1 a I è  $I*(I+1)/2$ , sicuramente il valore corretto era quello del Basic.

Ma come mai il Fortran aveva ritornato un valore così errato? Da un rapido consulto con **Eugenio Rapella**, abbiamo ipotizzato che sia la lentezza che l'errore di calcolo possano essere imputati alla conversione da INTEGER a FLOAT. Come fare quindi per trovare l'errore (a questo punto il fattore velocità era secondario)?

## Sommare i numeri interi ed i float

Ho fatto una prova sommando float ed integer fino a 200 ma, come possiamo vedere dal risultato (Fig. 1), non ci sono discrepanze:

```
10 : program test
```





```

LINKER
ARE YOU SURE? (Y/N CR=Y)
ENTER OBJECT FILE NAME: TESTOBJ
LIST MODULES? (Y/N CR=Y)
RUNABLE CODE RESIDES
FROM $0850 TO $1CE1
LINK COMPLETE
READY.
SAVE "TESTFRT",8
SAVING TESTFRT
READY.
RUN
 20100.
 20100
READY.

```

Fig. 1 - Somma utilizzando solo integer

```

15 :* this program sum
16 :* the first 200 numbers
20 : real sf,a
22 : integer si
25 : sf=0.0
26 : si=0
30 : open 3,3
40 : do 10 i = 1, 200
42 : si=si+i
45 : a=float(i)
50 : sf=sf+a
60 : 10 continue
65 : write (3,100) sf
66 : write (3,200) si
70 : close 3
75 : stop
80 : 100 format(f10.0)
85 : 200 format(i6)
90 : end

```

Come fare quindi per capire dove stava l'errore? Il consulto con Eugenio non si era limitato alle sole supposizioni, ma anche a suggerire un sistema pratico per trovare la risposta a questo quesito: *poiché la somma dei numeri interi da 1 a I è  $I*(I+1)/2$ , potresti inserire un "controllo" man mano che il programma la esegue in modo da vedere quando e perché si ottengono "discrepanze".*

```

10 : program test
15 :* this program sum
16 :* the first 10000 numbers
17 :* with error control
20 : real f,a,c
25 : f=0.0

```

```

30 : open 3,3
40 : do 10 i = 1, 10000
45 : a=float(i)
50 : f=f+a
51 : c=a*(a+1.0)/2.0
52 : if (f .ne. c) then
53 : write (3,*) 'errore:', f, c
55 : write (3,100) a
59 : endif
60 : 10 continue
65 : write (3,100) f
70 : close 3
75 : stop
80 : 100 format(f10.0)
90 : end

```

In questo programma andiamo a testare la somma dei numeri in virgola mobile con il calcolo  $a*(a+1.0)/2.0$  sempre ottenuto con i numeri in virgola mobile. Come si può vedere (Fig. 2) la differenza è immediata ed il numero ritornato dalla sommatoria è esattamente quello mostrato alla fine del ciclo... La nostra supposizione riguardo alla conversione da integer a float era esatta.

### Conclusioni

Che dire, non mi sarei mai aspettato queste difficoltà con il compilatore Fortran del Commodore 64. Per il momento, vista la cronica mancanza di tempo mi fermo qui, ma in futuro voglio trovare il modo di eseguire questo calcolo con il Fortran in maniera corretta.

Ho già alcune idee...

E voi? Volete suggerirmi qualcosa?

```

9990.
ERRORE : 4.98800E+07 4.99146E+07
9991.
ERRORE : 4.98900E+07 4.99246E+07
9992.
ERRORE : 4.99000E+07 4.99346E+07
9993.
ERRORE : 4.99100E+07 4.99446E+07
9994.
ERRORE : 4.99200E+07 4.99546E+07
9995.
ERRORE : 4.99300E+07 4.99646E+07
9996.
ERRORE : 4.99400E+07 4.99746E+07
9997.
ERRORE : 4.99500E+07 4.99846E+07
9998.
ERRORE : 4.99599E+07 4.99946E+07
9999.
ERRORE : 4.99699E+07 5.00044E+07
10000.
49969900.
READY.

```

Fig. 2 - Output del programma di controllo





# Power C – il miglior compilatore C per C64/128?

di David La Monaca

## Note personali

Devo essere sincero: nel periodo d'oro della diffusione del Commodore 64, la seconda metà degli anni 80, non avevo mai sentito parlare di Power C, né (mea culpa) di altri compilatori per il linguaggio C per quello che era il mio primo computer. Una combinazione di motivi sono alla base di questa mia lacuna: consideravo, a torto, il C64 troppo poco potente per un compilatore del linguaggio C che invece associavo, anche qui erroneamente, a macchine più evolute come Amiga o PC 386. Quegli anni furono per me gli ultimi del liceo ed i primi dell'università e tutto ciò che conoscevo per esperienza diretta era la programmazione in BASIC e assembly per C64. Quando passai ad Amiga, il C mi sembrava molto complesso da programmare per creare qualcosa di utile sotto Workbench, per cui la mia prima vera esperienza di codice in C fu su un PC 386 con il famigerato Turbo C di Borland. Nei primi due anni di Ingegneria Elettronica i soli corsi che prevedevano l'uso di un linguaggio di programmazione trattavano

soltanto il Fortran (Analisi Numerica) e Pascal (Calcolatori Elettronici). Del C neppure l'ombra. Per esercitarmi in Pascal avevo rimediato l'ottimo Oxford Pascal per C64 ed usavo a volte il C1 Pascal sul Sinclair QL di un mio caro amico che frequentava la facoltà di matematica. Quando cominciai ad esplorare sul serio il C, il PC DOS con scheda VGA ed il Turbo C 1.0, con il suo comodo ambiente integrato, semplificarono moltissimo il processo di scrittura, editing, compilazione e linking dei piccoli programmi con cui mi cimentavo.

Oggi, dopo aver scoperto il Power C per Commodore 64, pubblicato in realtà nel lontano 1986 da Spinnaker Software Corporation, devo ammettere che se l'avessi conosciuto all'epoca, difficilmente l'avrei mollato a favore del Turbo C o comunque l'avrei fatto con maggiore fatica e senso di nostalgia. Sì, perché il Power C, a mio avviso, rappresenta davvero una piccola stella brillante nella galassia del software esistente per Commodore 64. Un esempio formidabile di tool di programmazione completo e dalle mille possibilità. Un Commodore 64/128 base dotato di drive ed il dischetto di base del package di Power C sono tutto ciò che serve per iniziare a programmare in C e, aggiungo, iniziare alla grande! Il sistema di sviluppo proposto da Power C è semplice, ben progettato e ottimamente reso disponibile all'utente. È un sistema DOS-based, ossia tutti i componenti della catena di sviluppo sono programmi residenti su disco che di volta in volta vengono chiamati ad agire sul nostro codice sorgente, sul codice oggetto e sulle librerie di funzioni necessarie al nostro progetto, al fine di ottenere un programma finito che può girare anche indipendentemente dalla shell di Power C. Ed è proprio la shell l'elemento innovativo e centrale di questo compilatore che fornisce un prompt molto simile a quello che si può trovare su un sistema Unix.

## Il pacchetto software

Il manuale che accompagna il software di Spinnaker mostra in bella vista la sua caratteristica principale, cioè di essere un compilatore completo del linguaggio C per

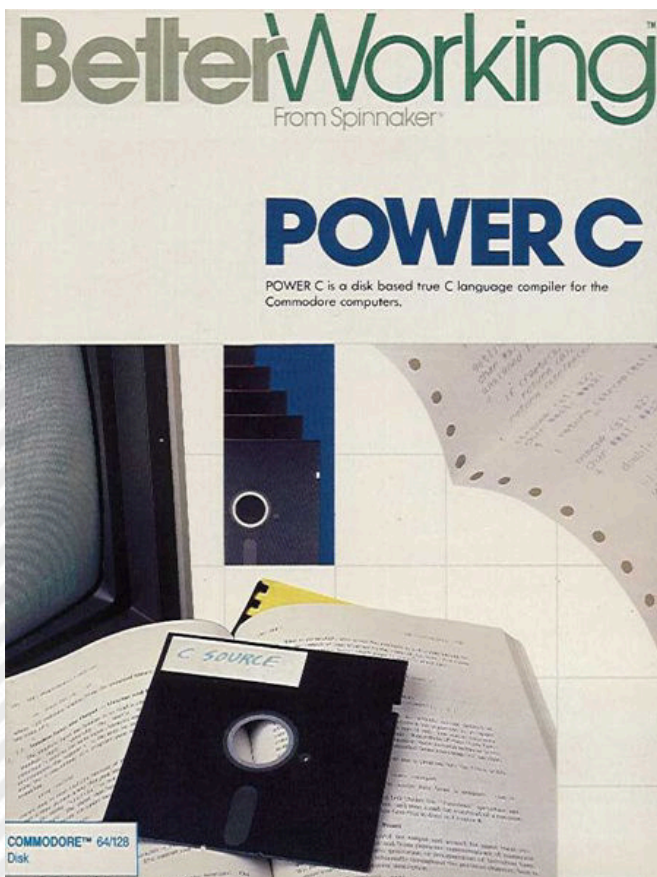


Fig. 1 - Copertina originale del package Power C





```
01, files scratched,01,00
$ ls *.c
0  Wpowerc      " la 3d
0  "sort.c"      seq
0  "find.c"      seq
0  "wfreq.c"     seq
3  "format.c"    seq
8  "print.c"     seq
8  "format2.c"   seq
8  "test.c"      seq
8  "lib.c"       seq
8  "pimci.c"     seq
8  "pigr.c"      seq
8  "pig.c"       seq
8  "pic.c"       seq
8  "picq.c"      seq
8  "piprova.c"   seq
8  "pia.c"       seq
8  "pib.c"       seq
8  "pinew.c"     seq
8  "pimc.c"     seq
220 blocks free.
$ ed pimc.c
```

**Fig. 2 - Schermata Shell C**

C64/128 interamente basato su disco. Oltre al Commodore 64 è supportato anche il Commodore 128, di cui sfrutta la maggiore quantità di memoria e le caratteristiche hardware. Sulle piattaforme più popolari a 8-bit di casa Commodore Power C promette di generare programmi dieci volte più veloci del BASIC, producendo codice oggetto nativo in linguaggio macchina, compatto e riutilizzabile. I componenti del compilatore (shell, editor a tutto schermo, compilatore su linea di comando, linker flessibile, libreria completa di 95 funzioni per i più svariati campi di applicazione, librerie aggiuntive in C di utilità generale per ordinamento, ricerca e formattazione di testi) si combinano perfettamente per offrire al programmatore un ambiente efficace e snello oltre ad un'esperienza di sviluppo proficua.

Lo standard C, per capirci l'ANSI C di Kernighan e Ritchie, è pienamente supportato e per chi vuole sfruttare le potenzialità del processore, della grafica e dell'audio dei computer Commodore, le librerie di supporto, anche quelle fornite da appassionati utenti nel tempo, offrono tutta una serie di funzioni specifiche per interagire facilmente con i chip custom del C64. Il pacchetto di base, come già ricordato, consiste di un solo dischetto 5.25" doppia faccia. Il primo lato chiamato System Disk include la C Shell, l'Editor, il Compilatore, il Translator, il Linker, le librerie standard Stdio e Math e 5 esempi di listati di codice sorgente C. L'altro lato del floppy disk è il Library Disk e contiene le librerie di funzioni Stdlib.l e Syslib.l, utili durante la fase di compilazione e linking dei progetti. Il manuale include anche un comodo tutorial per velocizzare i tempi d'apprendimento nell'uso dei vari strumenti offerti dal compilatore.

### Passare all'azione (nel 2023)

Se fossimo ancora negli anni 80, per procedere all'uso del compilatore avremmo effettuato una copia di backup del dischetto originale e avremmo cominciato con il classico comando LOAD"\*",8,1 seguito dal tasto RETURN per avviare la Shell di Power C e cominciare il nostro primo progetto. In caso disponessimo di un drive 1571, cosa piuttosto rara per i possessori di C64, meno per gli utenti di C128 (e ovviamente di C128D che l'avevano integrato), il dischetto sarebbe letto da entrambe le facce, mentre per tutti gli altri drive (1541, 1541-II, 1570, ecc.) la fase di compilazione avrebbe comportato alcuni cambi di disco (o, meglio, cambio di lato) per portare a termine la procedura ed ottenere il programma eseguibile su Shell o caricabile dal BASIC. Poiché era raccomandato l'utilizzo di un disco vuoto formattato come disco di lavoro in cui memorizzare i sorgenti dei file in C, i codici oggetto ed i file eseguibili, il numero di cambi di dischetto sarebbe salito vertiginosamente. Naturalmente i fortunati possessori di un drive 1581 avrebbero avuto vita più facile: sarebbe bastato riversare il contenuto dei due lati del dischetto in dotazione su un unico floppy disk 3.5" da 800 KB. E avrebbero avuto ancora tanto spazio a disposizione per molti progetti. Questo è proprio ciò che abbiamo fatto noi per provare Power C sia su un emulatore per PC sia su hardware reale, grazie alla grande quantità di periferiche moderne disponibili sul mercato del retrocomputing per C64. In particolare abbiamo utilizzato VICE 3.7 su PC Windows 10 configurandolo con un singolo drive 1581 e con una cartuccia Action Replay per velocizzare i caricamenti da disco. Il floppy disk virtuale da 800 KB è

```
main() {
    float xp,yp,sd;
    int pp,np,ci;

    printf("\nCalcolo di PI greco con");
    printf("\nil metodo Montecarlo.\n");
    getchar();

    np=900;
    ci=0;
    sd=0.0;

    for(pp=0; pp<np; pp+=1)
    {
        xp=rndgen();
        yp=rndgen();

        printf("\npp:%u  xp:%f  yp:%f",pp,
            if (xp<=sqrt(1-yp*yp)) {
                ci=ci+1;
                printf("\n  **");
            }
    }
}
main                               line 39  column 1
```

**Fig. 3 - Editor in azione**





stato creato con DirMaster (un tool utilissimo per la gestione di dischetti immagine D64, D71, D81, ecc.). Sul dischetto D81 abbiamo poi trasferito il contenuto dei due D64 originali di Power C. Per i test su hardware reale lo stesso file .D81 è stato copiato su chiavetta USB, poi collegata ad una Ultimate II+ connessa ad un Commodore 64 PAL originale. Anche l'interfaccia Ultimate II+ è stata configurata con una versione di cartuccia Action Replay (file CRT) per fornire una spinta decisa alla velocità di trasferimento sulla porta seriale del C64.

### Il solito programma di esempio? No, stavolta no!

Potevamo usare il solito "hello\_world.c" o il file "test.c" incluso per mettere alla prova la catena di sviluppo di un programma in C utilizzando Power C, ma sinceramente, spero siate d'accordo con me, non se ne può più! E allora abbiamo scelto un breve programma per il calcolo di pi greco secondo il metodo Montecarlo, che, se avete familiarità con le pagine di RMW dedicate alla programmazione, non vi suonerà per niente nuovo. In passato è stato usato come esempio di algoritmo e programmazione in BASIC, Forth e altri linguaggi di varie piattaforme a 8-bit. Di seguito mostriamo le operazioni compiute sull'emulatore o sulla macchina reale per illustrare l'uso del compilatore. Non ci sono grandi differenze tra i due contesti, ma naturalmente vedere dal vivo il modesto C64 macinare dati e sputare fuori il programma eseguibile perfettamente funzionante in relativo poco tempo è una soddisfazione notevole per chi, come il sottoscritto, ha mosso i primi passi nel settore proprio su questo computer.

```

$ cc pimc.c
insert source disk
#include <stdio.h>
#define EOF -1
#define NULL 0
#define stdin 1
#define stdout 2

typedef int FILE;
char *gets(), *fgets();
#include <math.h>

float exp(), log(), log10(), pow();
float sqrt(), floor(), ceil(), fabs();
float hypot(), cabs(), sin(), cos();
float tan(), asin(), acos(), atan();
float atan2(), sinh(), cosh(), tanh();

float rndgen(_seed) +
char *s1 = (char *) 0xd012;
char

```

Fig. 4 - Compilazione in corso

Cominciamo allora col caricare la Shell, vero e proprio centro nevralgico del pacchetto software e per farlo digitiamo il classico comando:

```
LOAD"*",8 <RETURN>
```

Al termine del breve caricamento, dopo il RUN, avremo a disposizione il simbolo \$ della Shell, pronto a ricevere i comandi di gestione del file system e lanciare programmi di Power C. Tra i vari comandi disponibili segnaliamo i più importanti rimandando alla documentazione ufficiale per la lista completa:

\$ **l** (oppure **ls**) [**pattern**]; elenca i file del disco corrente, es. **ls \*.c**

\$ **rm filename**; cancella il file specificato

\$ **mv file1 file2**; sposta o rinomina il file1 su file2 sul disco di lavoro

\$ **pr [filename]**; mostra il contenuto del file sullo schermo

\$ **pr >> [filename]**; stampa il contenuto di un file sul device 4 (stampante)

\$ **disk [command string]**; manda un comando al drive: es. **disk n0:[headername],[id#]** formatta il disco di lavoro

\$ **ed [filename]**; esegue l'editor a tutto schermo del file indicato

\$ **ced [filename]**; come sopra ma l'editor effettua il controllo della sintassi

\$ **cc [-p] [filename.c]**; compila il codice sorgente contenuto nel file filename.c. L'opzione -p specifica al compilatore che vi sono due drive in uso

\$ **link [-s [address]]**; lancia il linker per finalizzare la produzione di un eseguibile. L'opzione -s specifica che il programma risultante sarà eseguibile senza l'uso della Shell, quindi direttamente dal BASIC del C64. Se specificato, [address] indica la locazione di memoria di partenza del codice eseguibile, default = \$0801 inizio del BASIC

Il nostro primo step è dunque di aprire l'editor per creare il nostro programma in C per il calcolo di pi greco con il metodo Montecarlo. Nel riquadro 1 potete consultare il codice sorgente adeguatamente commentato. Quindi digitiamo nella Shell:

```
$ ed pimc.c ; oppure $ ced pimc.c
```

Verrà lanciato l'editor dove sarà possibile scrivere riga dopo riga l'intero programma. L'editor è molto flessibile





```

        printf(" *");
        |
        printf("\n ci:%u",ci);
        printf("\n pp:%u",pp);
        |
        printf("\n--> PI approx.: %f\n",4.0*
ci/pp);
        getchar();
        return(0);
        |
insert compiler disk
insert object disk
$ link
vvv pimc.o
vvv math.l
vvv ↑
output file name: pimc.sh

```

Fig. 5 - Processo di linking

```

pp: 879 xp: 815500 yp: 702400
pp: 880 xp: 768400 yp: 132400
pp: 881 xp: 768400 yp: 162400
pp: 882 xp: 768400 yp: 192400
pp: 883 xp: 768400 yp: 222400
pp: 884 xp: 768400 yp: 252400
pp: 885 xp: 768400 yp: 282400
pp: 886 xp: 768400 yp: 312400
pp: 887 xp: 768400 yp: 342400
pp: 888 xp: 768400 yp: 372400
pp: 889 xp: 768400 yp: 402400
pp: 890 xp: 768400 yp: 432400
pp: 891 xp: 768400 yp: 462400
pp: 892 xp: 768400 yp: 492400
pp: 893 xp: 768400 yp: 522400
pp: 894 xp: 768400 yp: 552400
pp: 895 xp: 768400 yp: 582400
pp: 896 xp: 768400 yp: 612400
pp: 897 xp: 768400 yp: 642400
pp: 898 xp: 768400 yp: 672400
pp: 899 xp: 768400 yp: 702400
C i : 707
P P : 900
--> PI approx. : 3.142222
$

```

Fig. 6 - Programma pimc in esecuzione

e offre diversi comandi di modifica, ricerca e formattazione del codice. Premendo il tasto Run/Stop sulla tastiera l'editor entrerà in command mode per poter impartire diversi comandi di gestione, inclusi quelli per leggere (GET filename), salvare (PUT filename), stampare (PRINT filename), listare la directory del disco (DIR), ricercare testo (/searchstring<F3>), ecc. All'interno dell'editor alcuni simboli speciali molto utilizzati in C si possono ottenere sulla tastiera del C64 mediante la combinazione di tasti. Ad esempio le parentesi graffe {} si ottengono con <SHIFT>+ e <SHIFT>-, back-slash “\” con il tasto “£”, underscore “\_” con la combinazione Commodore+@, pipe “|” con Commodore+\*. I comandi BYE o QUIT terminano il programma di editing e tornano alla Shell. Dopo aver digitato e salvato il file con il nome pimc.c, siamo ora pronti a compilarlo con il comando:

```
$ cc pimc.c
```

Inizierà il processo di compilazione, durante il quale verranno segnalati eventuali errori di sintassi del linguaggio e altre anomalie del codice. Al termine, se tutto sarà andato per il meglio, otterremo il codice oggetto nella forma di un file chiamato con lo stesso nome del file sorgente e con estensione “.o”.

A questo punto non resta che effettuare il linking del file oggetto con le eventuali librerie non standard utilizzate nel codice sorgente. La fase di linking è forse la procedura meno lineare di quelle previste nella toolchain di sviluppo di Power C, ma nulla di impraticabile. Lanciamo quindi il programma linker specificando il parametro -s se vogliamo che il nostro programma possa essere eseguito al di fuori della Shell di Power C, altrimenti il linker produrrà automaticamente un file con estensione “.sh” che potrà

essere eseguito solo nella Shell.

```
$ link [-s]
```

Il programma produrrà un prompt che ci consentirà di specificare nell'ordine: il codice oggetto, le librerie non standard da includere e infine il carattere/comando di avvio del processo effettivo. Per il nostro esempio, dal momento che nel programma abbiamo fatto uso di funzioni matematiche, oltre al codice oggetto pimc.o includeremo anche la libreria math.l.

```
>pimc.o
```

```
> ↑ math.l
```

```
>
```

Il tasto/carattere (“freccia in alto”, non il tasto “cursore su”) serve ad avviare il processo di linking al termine del quale otterremo il programma eseguibile. Nel nostro caso avremo il file pimc.sh (eseguibile via shell) oppure pimc.prg (caricabile ed eseguibile dall'ambiente BASIC del C64). Il nome del file eseguibile può essere indicato in questa fase quando il linker ne farà richiesta. In caso di errore sarà necessario fornire al linker le necessarie librerie di funzioni non standard. Naturalmente, nel test che abbiamo condotto, il drive 1581 emulato ci consente di evitare tutti i cambi di dischetto che invece dovremmo affrontare nel caso di un sistema con due drive 1541 o con drive singolo. Il pacchetto Power C fornisce molti altri programmi di utilità per semplificare e ottimizzare tutte le fasi dello sviluppo di programmi in C. Inoltre offre librerie aggiuntive fornite da terze parti per gestire efficacemente le capacità grafiche, sonore e di I/O (user port e porte joystick) peculiari della piattaforma C64.

## Conclusioni





In passato su RMW sono apparsi diversi articoli riguardanti interpreti e compilatori di diversi linguaggi più o meno popolari e diffusi per le piattaforme a 8-bit. Ad esempio il compilatore Abacus Super C è stato ampiamente recensito da Francesco Fiorentini e devo dire che al giorno d'oggi c'è sempre da rimanere stupiti del livello che possono ancora raggiungere dopo 30-40 anni i nostri piccoli amati retro computer quando fanno girare il giusto pacchetto di sviluppo. Abacus Super C si era dimostrato un compilatore semplice da usare e molto versatile ed efficiente (rif. RMW #32-IT e #10-EN). Anch'io lo avevo provato su emulatore e su macchina reale e posso sinceramente affermare che mi aveva impressionato per capacità e flessibilità. In modo altrettanto sincero posso dire, senza timore di essere smentito, che Power C era ed è uno dei compilatori più rapidi e potenti per i computer 8-bit di casa Commodore. Imparare ad usarlo da esperti e in modo produttivo non è un processo immediato, ma la curva di apprendimento, benché appaia più lunga di quella relativa ad altri compilatori, riserva molte soddisfazioni lungo il percorso. Lo strumento operativo della Shell è davvero sorprendente: con le dovute proporzioni è come disporre di una Bash per Linux, dinamica e piena di comandi e funzioni utili per la gestione del filesystem e dell'ambiente di sviluppo. Davvero inaspettato per un sistema di soli 64KB di RAM e CPU a 1MHz. Il compilatore ed il linker sono veloci e utili nella fase di controllo del codice C, l'editor a tutto schermo nel suo piccolo non fa rimpiangere gli IDE di piattaforme molto più potenti del C64 e tutta la catena di produzione del software (edit-compile-link) si muove fluida e senza particolari intoppi. Dopo essersi allenati un po' con i comandi disponibili nei vari strumenti del pacchetto, l'efficienza, la produttività e, perché no?, il divertimento sono garantiti.

## Riferimenti

**C Programming Language Book** - [https://en.wikipedia.org/wiki/The\\_C\\_Programming\\_Language](https://en.wikipedia.org/wiki/The_C_Programming_Language)

**DirMaster** - <https://style64.org/dirmaster>

**Power C** - <https://www.lyonlabs.org/commodore/onrequest/powerc/index.html>

**Power C Wiki** - [https://www.c64-wiki.com/wiki/Power\\_C](https://www.c64-wiki.com/wiki/Power_C)

**RMW past issues** - <https://www.retromagazine.net>

```
#include <stdio.h>
#include <math.h>

float rndgen(_seed) {
    // This function's code (c) Marco Spedaletti
    char *s1 = (char *) 0xd012; //pointer to raster
    register
    char *s2 = (char *) 0xa2; //pointer to location
    161-162
    char *s3 = (char *) 0xa1; //jiffy clock updated
    by KERNAL IRQ every 1/60 second
    int seed = 0;

    if ( _seed ) {
        seed = ((*s1<<8)^(*s2<<8)).((*s3)^(*s2)); //
    computes seed
    } else {
        seed = ((*s3<<8)^(*s1<<8)).((*s2)^(*s1));
    }

    //generates random number
    srandom(seed);
    unsigned int result = (unsigned int)random();
    return( (result%10000)/10000.0 );
}

main() {
    float xp,yp;
    int pp,np,ci;

    printf("\nCalculus of PI with");
    printf("\nMontecarlo's method.\n");
    getchar();

    np=300; // no. of iterations
    ci=0; // counter

    for(pp=0; pp<np; pp+=1) {
        xp=rndgen();
        yp=rndgen();

        printf("\npp:%u xp:%f yp:%f",pp,xp,yp);

        if (xp<=sqrt(1-yp*yp)) {
            ci=ci+1;
            printf(" *"); //prints * if point is in
        }
    }

    printf("\n ci:%u",ci);
    printf("\n pp:%u",pp);

    printf("\n--> PI approx.: %f\n",4.0*ci/pp);
    getchar();
    return(0);
}
```

Listato del programma pimc.c per il calcolo di Pi greco con il metodo Montecarlo







# Libdragon Development: il kit per creare giochi su Nintendo 64 - parte 2

di Takahiro Yoshioka – trad: Carlo Nithaiah Del Mar Pirazzini

## Console Libdragon

La console Libdragon è quella utilizzata per visualizzare testo semplice in modo semplice e sicuro sul display dell'N64. Viene utilizzato principalmente per il debug, ma può essere utilizzato anche come motore per un semplice gioco basato su testo.

## Avvio della console

Avviare la console è abbastanza semplice, tutto ciò di cui hai bisogno è questa seguente riga di codice. Avvierà la console e ti consentirà di iniziare a stampare il testo su di essa.

```
1 | console_init();
```

La console ha 28 righe con 64 caratteri ciascuna ed è riempita verticalmente di 8 pixel e orizzontalmente di 64 pixel. È possibile fare riferimento a questi valori con queste macro:

```
1 | #define CONSOLE_WIDTH 64
2 | #define CONSOLE_HEIGHT 28
3 | #define HORIZONTAL_PADDING 64
4 | #define VERTICAL_PADDING 8
```

## Facoltativo: impostare la modalità

Sono disponibili due modalità per stampare sulla console: `RENDER_MANUAL`– Mantiene le istruzioni stampate in un buffer e attende che vengano richiamate manualmente da `console_render()`.

`RENDER_AUTOMATIC`– Rende ogni comando di stampa della console così come viene stampato.

Queste opzioni sono impostate in questo modo (scegli solo una riga):

```
1 | console_set_render_mode(RENDER_MANUAL);
2 | console_set_render_mode(RENDER_AUTOMATIC);
```

## Stampa del testo

Una volta attivata la console, è ora di iniziare a stampare del testo su di essa. Ciò può essere ottenuto utilizzando funzioni che stampano su stdout, principalmente le seguenti: `printf()`– Stampa una stringa formattata `puts()`– Stampa una stringa non formattata con un carattere di fine riga ( `\n`) alla fine `putchar()`– Stampa un singolo carattere Queste funzioni sono presenti nella libreria standard `<stdio.h>`, quindi dovrebbero essere abbastanza familiari.

```
1 | printf("Hello world %i!\n", 1234);
2 | puts("Hello to you too!");
3 | for (char i=0; i<26; i++) {
4 |     putchar(65+i);
5 | }
6 | putchar('\n');
```

Questo frammento restituirà qualcosa del genere:

```
Hello world 1234!
Hello to you too!
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

## Appunti

È importante aggiungere un ritorno a capo alla fine altrimenti il testo precedente non verrà stampato. È possibile stampare più istruzioni print su una riga senza





un ritorno a capo (come putchar()) sopra) purché alla fine abbia un \n punto successivo.

I ritorni a capo ( \r) non funzionano sulla console Libdragon, né esiste alcun modo per modificare il testo già stampato senza eliminare e riscrivere il tutto.

### Tabella ASCII

Ecco la tabella ASCII dei caratteri stampabili. Il set ASCII principale (32-127) è più o meno quello che ti aspetteresti, ma alcuni caratteri non stampabili (1-31) emettono alcuni simboli. I caratteri ASCII estesi (128-255) sono per lo più versioni senza senso o leggermente danneggiate di altri caratteri, quindi dovrebbero essere evitati, a meno che tu non stia cercando di creare una sorta di arte ASCII.



### Cancellazione dello schermo

Se le stampe della console raggiungono la parte inferiore dello schermo, la console scorrerà automaticamente verso il basso. Una volta raggiunta la parte inferiore dello schermo ci sono solo due modi per modificare ciò che si trova sopra: cancellare e sovrascrivere.

Sovrascrivere significa che devi semplicemente continuare a stampare il testo per spostare nuovamente la console fino a quando non la porti dove desideri. Può essere un po' noioso tenere traccia di tutte le linee, quindi non è consigliabile a meno che non si desideri conservare una parte delle stampe precedenti.

La cancellazione comporta la pulizia del buffer della console e il posizionamento del cursore nella parte superiore sinistra dell'array come se non fosse stato scritto nulla. Questo può essere fatto utilizzando la seguente funzione. È particolarmente utile quando si desidera scrivere un output su più righe sulla console, ad esempio emulando una GUI in modalità solo testo.

```
1 | console_clear();
```

### Personalizzazione della console

Puoi utilizzare il testo bianco standard su sfondo nero, ma puoi anche personalizzare il colore della console come ritieni opportuno. Per fare ciò è necessario utilizzare la graphics\_set\_color() funzione, che accetta come input due int a 32 bit senza segno; il primo per il colore del testo e il secondo per lo sfondo. Ecco il prototipo della funzione:

```
1 | void graphics_set_color (uint32_t forecolor, uint32_t backcolor);
```

Quindi puoi impostare il colore utilizzando un numero esadecimale RGBA o utilizzando la funzione graphics\_make\_color() in questo modo (entrambi i metodi fanno la stessa cosa):

```
1 // Prototype for syntax purposes only, not needed in actual code
2 uint32_t graphics_make_color (int r, int g, int b, int a);
3
4 graphics_set_color(
5     graphics_make_color(255,0,0,255),
6     graphics_make_color(0,255,0,255)
7 );
8 graphics_set_color(
9     0xF801F801,
10    0x07C107C1
11 );
```

Tieni presente che il numero esadecimale è a 32 bit ma il valore al suo interno è costituito da due colori ripetuti a 16 bit. Inoltre, il colore viene impostato per l'intera console contemporaneamente, non è possibile codificare in modo specifico determinati caratteri o righe alla volta.

### Ripulire le cose

Una volta che hai finito con la console, puoi richiamare la seguente funzione per chiudere la console e liberare la memoria che stava utilizzando:

```
1 | console_close();
```

Tieni presente che il testo rimarrà sullo schermo a meno che non lo usi console\_clear(); prima di chiudere la console.

### Debug

Se utilizzi un carrello flash con debug USB, puoi utilizzare quanto segue per abilitare il debug utilizzando questo







# Guida al Sega Genesis Development Kit (SGDK) - parte prima

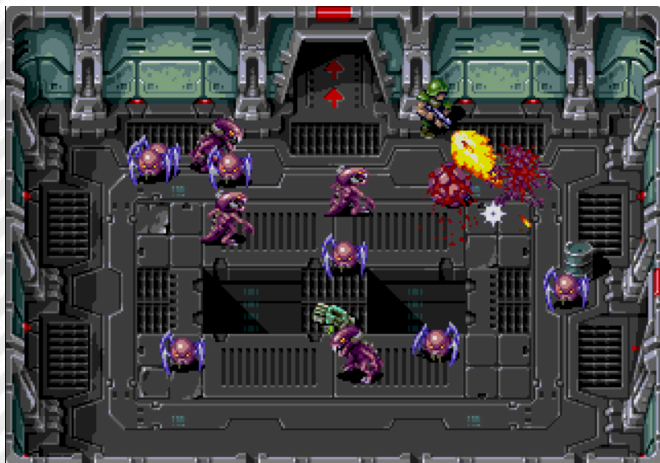
di Francesco Donatini

Creato e correntemente mantenuto da Stephane Dallongeville, il Sega Genesis Development Kit (SGDK) è un maturo framework open source (con licenza MIT) disponibile per Windows, Linux e MacOS X, che consente di sviluppare giochi per il Sega Megadrive (alcuni titoli: The Cursed Knight, Metal Dragon, Tanzer, Demons of Asteborg, o il più noto Xeno Crisis) utilizzando il linguaggio C. Provenendo da esperienze di programmazione con linguaggi di alto livello come Java e C#, ed essendo da tempo alla ricerca di un modo per ottenere performances decenti su macchine a 8-16 bit senza passare dall'assembly, l'SGDK ha letteralmente costituito il mio punto di ingresso per lo sviluppo di giochi su macchine retro, pertanto in questo e nei successivi articoli vorrei condividere con i lettori della nostra amata rivista le mie esperienze con questo strumento, confidando che possano essere di interesse per altri neofiti del retro coding come me.

## PREPARIAMO IL SETUP

Per iniziare ad usare l'SGDK si può sfruttare l'immagine Docker oppure, una volta che ci siamo assicurati di avere installato il runtime JRE di Java (dalla versione 8 in su), per chi come me usa Windows si potrà:

- scaricare la versione più recente del framework da Github (attualmente la 1.90 di luglio 2023);
- creare nel filesystem una cartella destinata ad ospitare il nostro progetto con le sequenti sottocartelle:
- src → in cui metteremo il codice sorgente C



**Fig. 1 - Xeno Crisis (2019 - Bitmap Bureau) convertito anche per Switch, PS4, PC, Dreamcast e Neo Geo**



**Fig. 2 - Il mio Megadrive (con Everdrive) all'opera al Festival 4GAME (Cerea - 16/17 Settembre 2023)**

- inc → in cui metteremo i file di include C
- res → in cui metteremo le risorse del gioco (grafica, audio, ecc.)
- out → in questa cartella sarà automaticamente generato il risultato della compilazione
- scrivere il codice sorgente;
- compilare utilizzando i binari di GCC 6.3 inclusi nell'archivio SGDK scaricato (make.exe nella cartella bin) eseguendo il seguente comando nella cartella di root del nostro progetto: {CARTELLA SGDK}\bin\make -f {CARTELLA SGDK}\makefile.gen

In caso di compilazione avvenuta con successo otterremo nella cartella "out" un file "rom.bin" che se dato in pasto ad un emulatore (o ad un vero megadrive tramite una flash cart come l'Everdrive) permetterà alla nostra creazione di essere eseguita e giocata!

Personalmente utilizzo come IDE di sviluppo Visual Studio Code con le estensioni Genesis Code di zerasul e C/C++ for Visual Studio Code di Microsoft, e mi sono preparato un semplice file BAT la cui esecuzione, associata ad una shortcut dell'IDE, fa partire la compilazione del progetto e l'emulatore con caricato il gioco appena compilato.





```
#include <genesis.h>

int main(u16 hard)
{
    VDP_drawText("Hello RetroMagazine!", 10, 13);

    while(TRUE)
    {
        SYS_doVBlankProcess();
    }

    return 0;
}
```

Fig. 3 - Il nostro codice di Hello World

#### OTTENIAMO LA NOSTRA PRIMA ROM

Per ottenere la nostra prima Rom con il classico Hello World sarà sufficiente creare un file con estensione .c con contenente il codice in Fig. 3.

Anche chi fosse a totale digiuno di C non fatterà ad individuare una struttura di semplice lettura composta da:

- l'inclusione del framework tramite l'header genesis.h;
- il punto di inizio del programma che in C è dato dal contenuto della funzione main;
- il comando VDP\_drawText che si incarica di scrivere il testo in una determinata posizione dello schermo;
- il loop di gioco individuato dal ciclo while che viene eseguito all'infinito;
- l'esecuzione all'interno del loop di gioco di un non meglio specificato "processo" contestuale al vertical blank

#### COMPRENDIAMO MEGLIO COSA ABBIAMO SCRITTO

Come sempre accade quando si mettono le mani sulle nostre amate retro console/computer, per avere contezza di cosa stiamo facendo e di cosa è possibile fare occorre conoscere le caratteristiche dell'hardware su cui stiamo programmando, e l'SGDK, per quanto "nasconda" una parte di quanto accade dietro le quinte tramite funzioni di alto livello, non fa eccezione.

Quando viene inclusa la libreria del framework, questo si occupa di compiere una serie di azioni preliminari prima ancora che venga eseguita la funzione main:

- vengono svuotati i 64KB che compongono la RAM del megadrive;
- viene preparato il VDP (Video Display Processor), che è il chip video custom utilizzato per la grafica, con una configurazione di default;
- vengono svuotati i 64KB che compongono la VRAM (video RAM), che è una RAM dedicata alla parte grafica;
- vengono caricate quattro palette di default: grigia, rossa, verde, blu. Il megadrive può mostrare a schermo fino appunto a quattro palette di sedici colori ciascuna a scelta tra un totale di 512 possibili;
- viene caricato un font di default (con cui potremo visualizzare il nostro Hello World);
- viene inizializzata la gestione dell'input;
- viene resettato il sistema sonoro (gestito da due chip: Yamaha YM2612 e Texas Instruments SN76489)

Quando viene chiamata la funzione main, il parametro definito dal numero intero senza segno a 16 bit hard indica:

- quando vale zero, che è stato premuto il tasto di reset del megadrive;
- quando vale uno, che la console è stata accesa usando il tasto di accensione

I parametri di posizione del testo che abbiamo inserito col comando VDP\_drawText (10, 13) non sono espressi in pixel ma in tiles, che sono dei quadrati di 8x8 pixel e





0	0	1	1	1	0	0	0
0	1	1	0	1	1	0	0
2	2	0	0	0	2	2	0
2	2	0	0	0	2	2	0
1	1	1	1	1	1	1	0
1	1	0	0	0	1	1	0
1	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0

**Fig. 4 - Rappresentazione in bit di una tile, che in memoria occupa 8 bytes, dove ciascun byte rappresenta una riga e il valore del singolo bit rappresenta il colore**

rappresentano l'unità grafica minimale che il VDP può rappresentare, quindi il testo sarà collocato alle coordinate in pixel (80, 10). Dietro le quinte, il metodo `VDP_drawText` si occupa di trasferire i dati grafici inerenti il testo che vogliamo visualizzare (vedi nota in fig. 3) nella VRAM del VDP (che può ospitare fino a 2048 tiles) utilizzando un piano (di default il piano A) e un metodo di trasferimento (di default tramite CPU). I piani, i metodi di trasferimento e altre caratteristiche del VDP saranno trattate nel prossimo articolo dedicato agli sprites e al comparto grafico.

Quando dentro il loop di gioco viene chiamato `SYS_doVBlankProcess` l'SGDK si mette in attesa del vertical blank (cioè del tempo che il raster impiega per tornare alla prima linea dello schermo una volta che ha finito di proiettare l'ultima) per eseguire in quel lasso di tempo una serie di attività, (come il trasferimento di dati nella VRAM del VDP) tra cui anche alcune che devono essere necessariamente effettuate frame per frame come il polling dell'input ecc.

Tornando al nostro codice in cui visualizziamo del testo a schermo, ci si potrebbe aspettare che eliminando il comando `SYS_doVBlankProcess` non vedremo più il testo a schermo poiché non eseguiamo le attività legate al vertical blank tra cui il trasferimento dati nella VDP, ma in realtà il metodo di trasferimento tramite CPU utilizzato di default dal comando `VDP_drawText` effettua un trasferimento immediato dei dati alla VDP e pertanto il testo sarà comunque visibile; al contrario se avessimo

specificato un metodo di trasferimento che attua il trasferimento al vertical blank il testo non sarebbe visibile se non eseguendo il comando `SYS_doVBlankProcess` (potete provare sostituendo `VDP_drawText` con questo comando: `VDP_drawTextEx(BG_A, "Hello RetroMagazine", TILE_ATTR(PALO, FALSE, FALSE, FALSE), 10, 10, DMA_QUEUE);`

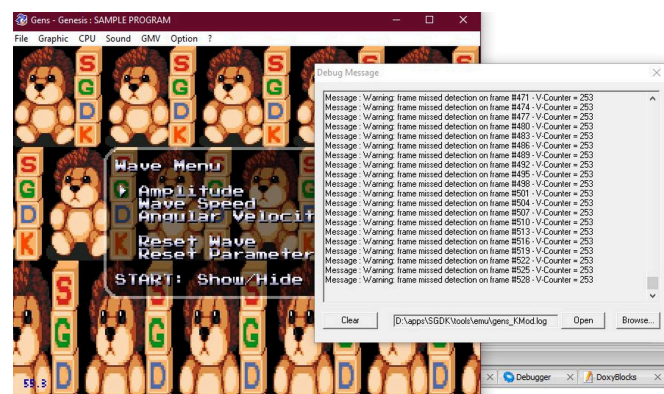
## UNO SGUARDO AGLI EMULATORI IN CHIAVE DI SVILUPPO

L'emulatore che ritengo per esperienza il più accurato, in cui ho trovato una per ora praticamente perfetta corrispondenza con la macchina vera (il gioco andava in crash sia sull'emulatore che sulla macchina reale allo stesso punto!) è sicuramente il BlastEm, che offre anche la possibilità di personalizzare le impostazioni e, ad esempio, far partire l'emulatore in full screen e renderlo praticamente "invisibile" così da poter generare degli eseguibili che facciano partire il gioco in maniera diretta, senza dover utilizzare l'interfaccia dell'emulatore per caricare la rom.

L'altro emulatore che utilizzo e che trovo molto carino per gli effetti grafici è il Kega Fusion, mentre ai fini di debug (tematica che approfondiremo) possono inoltre essere utilizzati il Gens\_KMod, che un tempo veniva incluso tra gli stessi files del framework, e il Regen Debug Version (v0.972).

## RISORSE UTILI

- <https://github.com/Stephane-D/SGDK>
- <https://plutiedev.com/>
- <https://danibus.wordpress.com/>
- <https://www.copetti.org/writings/consoles/mega-drive-genesis/>
- [https://segaretro.org/Sega\\_Mega\\_Drive/Technical\\_specifications](https://segaretro.org/Sega_Mega_Drive/Technical_specifications)



**Fig. 5 - L'emulatore Gens\_KMod con caricato un esempio incluso nel SGDK e la finestra con i messaggi di debug**





# Casi di identità... - parte I

di Marco Pistorio

Bentrovati, amici lettori di RetroMagazine World!  
Come forse saprà qualcuno dei nostri più fedeli lettori, mi occupo da sempre di informatica nella vita che conduco tutti i giorni.

I miei primi passi in questo tortuoso sentiero li compii studiando la programmazione insieme al mio fidato "biscottone", ovvero il mio Commodore 64, ed ancora oggi, nel (poco) tempo libero che riesco a ritagliarmi di tanto in tanto, provo a realizzare qualcosina per questa macchina che continuo ad adorare.

C'è da dire che sono in buona compagnia. Gli estimatori di queste stupende macchine sono tanti e crescono ancora, incredibile a dirsi. E molti sono davvero bravi, talentuosi, davvero in gamba.

Forse perché le informazioni a disposizione oggi sono numerose e varie ed è facile ed immediato accedervi. Forse gioca la sua parte anche la nostalgia di quegli anni verdi, quando queste macchine erano nuove fiammanti e tanti di noi erano ragazzini o poco più. Probabilmente ci saranno ulteriori ingredienti che rendono così appetibili queste macchine vetuste e che rendono la loro programmazione interessante e piacevole ancora oggi, nonostante si tratti di hardware sicuramente non recente ed estremamente limitato, ad esempio se si confrontano le loro caratteristiche tecniche con quelle degli attuali PC.

Vi anticipo che questo articolo è molto tecnico e si rivolge, in particolare, a tutti coloro che hanno qualche esperienza nello sviluppo su Commodore 64 (o su macchine similari) e che impiegano per i loro lavori strumenti moderni di programmazione, come IDE, cross-assemblatori, strumenti di compressione dati, ecc.

Di cosa si tratta più precisamente? Si tratta principalmente di un'esperienza che ho deciso di condividere con Voi, cari lettori di RetroMagazine World.

Probabilmente alcune delle conclusioni alle quali sono arrivato potranno essere utili (spero) anche a Voi.

Da cosa è nato tutto ciò di cui Vi voglio parlare?

Da un 'grosso' problema che mi sono, ahimè, posto. Ho trovato in Rete una scherzosa definizione di informatico

che secondo me calza davvero a pennello, in questo specifico caso in particolare. :D

La definizione recita più o meno così: *"L'informatico è una persona che risolve costantemente problemi che non sapevi nemmeno esistessero, in modi che non capisci"*.

A questo punto Vi espongo il mio dilemma.

Ormai qualche anno fa, sviluppai dei lavori, dei semplici giochini, delle semplici "intro" in assembly per C64. Decisi poi di pubblicare buona parte di questi lavori sul sito CSDB. Per chi fosse curioso di vederli Vi posto il link da cliccare immediatamente qui di seguito:

<https://csdb.dk/scener/?id=29165>

Per chi di Voi predilige i video su YouTube, posto di seguito il link del mio canale personale, dove potrete trovare alcuni di questi lavori che ho pubblicato su CSDB:

<https://www.youtube.com/@mpsoftita-retroprogramming8233/>

Ebbene, il problema che mi si è presentato è stato il seguente: "come faccio a capire che l'eseguibile che ho pubblicato corrisponda ESATTAMENTE a quello che ottengo ricompilandone il codice sorgente in mio possesso?"

Molti di Voi penseranno in questo momento: "beh, se possiedi il codice sorgente, dove sta il problema? Senza dubbio è immediato ricavarne il codice eseguibile..."

Qui, da fan sfegatato di Star Trek, mi viene da pensare ad una frase pronunciata dal Sig. Spock nel film "Star Trek- Rotta verso l'ignoto": *"La logica è solo la premessa della saggezza, Valeris, non il suo epilogo."*

Vi spiego ora il perchè di questa frase che mi torna in mente. E' verissimo ed immediato che, ricompilando il codice sorgente, si ottenga alla fine il relativo codice eseguibile. Chi sviluppa lo sa anche fin troppo bene.

Tuttavia, ottenere un eseguibile "perfettamente uguale" ad un altro, partendo dal medesimo codice sorgente, non è proprio così semplice.

Il compilatore che si adopera durante le due diverse operazioni DEVE essere esattamente LO STESSO, e ciò





deve valere per eventuali ulteriori passaggi successivi (ad esempio, il tool che si adopera per comprimere i dati al fine di ottenere un codice eseguibile più ridotto in termini di occupazione di memoria).

Ma a volte tutto ciò non basta.

Solitamente si deve lavorare con lo stesso Sistema Operativo per ottenere una perfetta compatibilità binaria tra i due eseguibili. A volte con gli stessi aggiornamenti di sistema, che modificano determinate librerie e le portano alla stessa precisa versione affinché il codice eseguibile ottenuto in seconda istanza risulti perfettamente analogo a quello ottenuto in precedenza.

Inoltre è bene precisare che questi ultimi elementi (aggiornamenti di sistema e librerie) rivestono la loro importanza quando si genera codice eseguibile destinato a funzionare all'interno del Sistema Operativo di un PC.

Quindi, per tornare alla massima del Sig. Spock, il poter ricompilare il codice sorgente originario è solo la premessa per riuscire ad ottenere un eseguibile che sia l'esatto duplicato di un eseguibile precedentemente generato.

Ecco un problema da "vero" programmatore, no? Mi riconduco qui alla definizione scherzosa di programmatore che Vi enunciavo poc'anzi. :D

Mi si è presentato un particolare caso di una mia intro che realizzai nel 2017 e che ho provato a comprimere (dopo averla ricompilata, naturalmente) impiegando diverse versioni del tool di compressione dei dati che adoperai a quel tempo, che si chiama "EXOMIZER", e non sono riuscito ad ottenere lo stesso, identico risultato che ottenni allora.

Dopo diversi ed infruttuosi tentativi decisi di rinunciare. E' chiaro che, lanciando l'eseguibile che ottengo dalla compilazione, la intro si comporti nel modo atteso.

Tuttavia il codice che ottengo, dopo averlo compresso, non ha la stessa identica "firma" di quello che pubblicai nel 2017.

C'è poco da fare. Le firme che ottengo sono sempre diverse. Mi ricordai che la mia macchina di produzione era allora una macchina Linux e quindi, molto probabilmente, l'EXOMIZER funziona in modo diverso quando gira su una piattaforma Windows piuttosto che su una piattaforma Linux.

Fortunatamente, in diversi altri casi, esaminando altri miei lavori pubblicati in tempi diversi, l'esito è stato favorevole :) (vedi fig. B).

Ecco quindi un altro concetto che volevo condividere con Voi. Parlo di "firma" poc'anzi. Di cosa si tratta?

Riflettiamo intanto su un aspetto fondamentale, direi basilare. Un programma eseguibile, dal punto di vista del nostro PC, è sempre e comunque un file.

Un file è un insieme di caratteri (ovvero bytes) memorizzati nella memoria del PC.

E' possibile pertanto esaminare qualsiasi programma eseguibile che si trovi in tale forma, byte per byte, ovvero carattere per carattere, al fine di calcolare un valore, un checksum, una "firma" di tale file, appunto.

Tale "firma" dovrà essere diversa nel momento in cui anche un solo byte esaminato nell'insieme in analisi risulti differente.

E' possibile adoperare diversi metodi per calcolare queste firme.

Personalmente prediligo l'MD5, che si può facilmente implementare anche in semplici applicazioni .Net (vedi Riquadro 1 e Fig. A).

Per chi volesse approfondire l'argomento:

<https://it.wikipedia.org/wiki/MD5>

Ora che sappiamo che è possibile generare "MD5 checksum" (o "MD5 Hash") dai nostri programmi eseguibili, parliamo di cosa è bene evitare se ci interessa analizzare i nostri lavori tramite questi metodi.

Una di queste cose che è da evitare è l'inclusione sistematica di uno o più caratteri ottenuti in maniera casuale nei nostri programmi ad ogni sua compilazione.

Specifiche di questo genere in un nostro programma assembly (l'esempio è in sintassi Kick Assembler):

```
.....
.....
casuali:
.fill 256,round(15*random())
.....
```

renderebbero le firme dei nostri eseguibili diverse ogni qualvolta li andassimo a generare.

Perchè? Perchè introdurremmo al loro interno una sequenza di X-caratteri (256 nell'esempio) casuali, che verrebbero ricalcolati ad ogni ricompilazione. Risultato? "Firme" sempre diverse.

Un possibile rimedio? Generare un segmento contenente tali bytes una sola volta e poi collegarlo al nostro programma







sorgente, in maniera tale che adoperi sempre lo stesso insieme di bytes random anche se lo dovessimo ricompilare più volte.

Un'altra pratica che Vi suggerisco caldamente di adoperare è quella di segnare SEMPRE, magari all'interno di apposite note testuali all'interno del codice sorgente dei vostri programmi, il nome e la versione del compilatore utilizzato, nonché i nomi di tutti i tools eventualmente impiegati in cascata unitamente ai loro numeri di versione ed infine la data di generazione e la piattaforma utilizzata per lo sviluppo. Vi sarà così più semplice "rimettere insieme i pezzi" all'occorrenza.

Ma, supponiamo di avere due eseguibili con due firme tra loro diverse. Come fare a stabilire se possano essere riconducibili al medesimo codice sorgente oppure no? Questo è l'argomento che proverò ad approfondire nella prossima puntata. Per il momento c'è già carne al fuoco a sufficienza! :D

Un caro saluto a tutti Voi, amici lettori. Alla prossima!

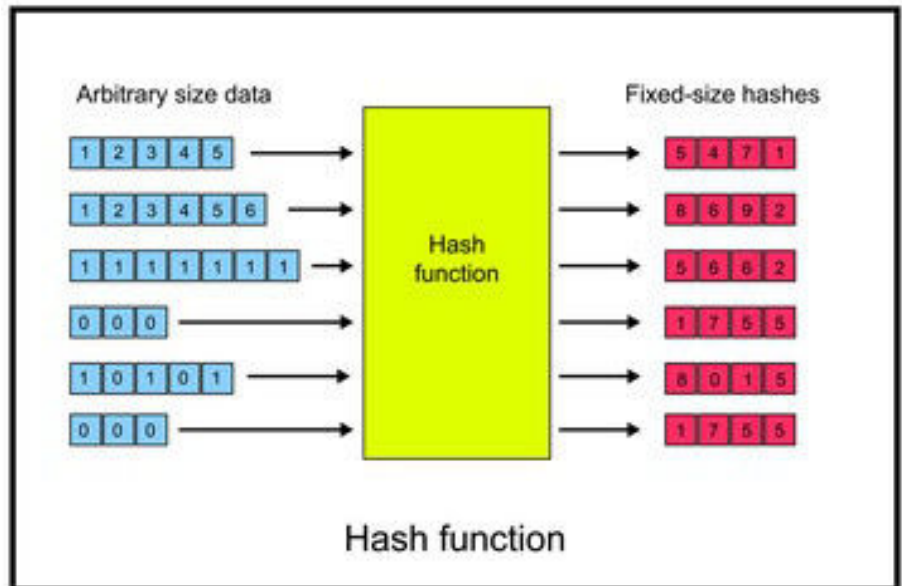


Fig. A - Schema concettuale della funzione di HASH (fonte sito <https://stock.adobe.com/search?k=md5>)

Elem contiene il nome del file, completo del suo path, da esaminare

Public Function calcola\_hash(elem As String) As String

```
Dim fileBytes() As Byte
Dim Md5 As New MD5CryptoServiceProvider()
Dim byteHash() As Byte
fileBytes = File.ReadAllBytes(elem)
byteHash = Md5.ComputeHash(fileBytes)
Return Convert.ToBase64String(byteHash)
```

End Function

Riquadro 1. Esempio di codice VB.net per determinare il calcolo dell'MD5 HASH di uno specifico file

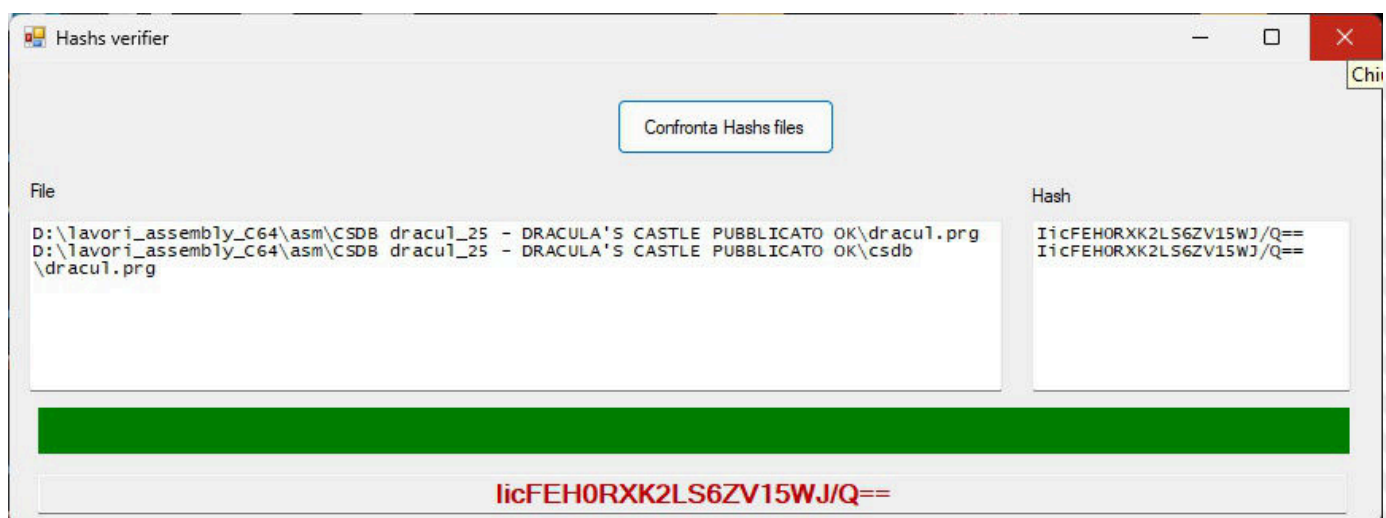


Fig. B - Confronto dell'MD5 HASH del gioco che pubblicai su CSDB "Dracula's Castle" nel 2017 con quello che ho ricompilato di recente. L'esito del confronto è positivo, quindi i due files .prg sono assolutamente identici, fino all'ultimo byte.





# PunyInform, una nuova libreria per scrivere avventure testuali per vecchi computer.

di Fredrik Ramsberg, tradotto ed adattato da Gianluca Girelli

Nota iniziale: questo articolo fa parte di una serie di cinque tutorials scritti da Fredrik Ramsberg, e qui riprodotto con il consenso dell'autore.

Fredrik, con l'amico e collega Johan Berntsson, ha sviluppato sia PunyInform (una libreria compatta per sviluppare text adventures su macchine a 8-bit) che Oozmo (un interprete Z-code che può essere utilizzato per rendere giocabili i giochi PunyInform su molte macchine Commodore). Fredrik e Johan sono anche gli organizzatori di PunyJam (ormai giunta alla 4<sup>a</sup> edizione), una competizione amichevole (ma non per questo meno agguerrita) dove, entro un tempo limite, si sviluppano avventure testuali. Il soggetto dell'avventura cambia da jam a jam e viene comunicato appena prima il suo inizio.

PunyInform è una libreria per il linguaggio di programmazione Inform 6, sviluppata per facilitare la creazione di avventure testuali semplici o avanzate in grado di funzionare bene anche su computer a 8 bit come Commodore 64, Amstrad CPC, Apple II, Spectrum +3 ecc. La prima versione di PunyInform è stata rilasciata il 5 luglio 2020.

## Le radici

Il linguaggio di programmazione, il compilatore e la libreria

```
Ventilation Shaft      Score: 0/0
HIBERNATED 2
A Science Fiction Story by Stefan Vogt
(Type "about" for game information.)
Release 1 / Serial number 200702 /
PunyInform v1.0 / Inform v6.34

Ventilation Shaft
It takes a moment before you realize
that you must have missed your
destination. Instead you find yourself
in a narrow shaft, lying on your
stomach, because there is not enough
room to stand or turn around. Solid,
impenetrable metal surrounds you. An
object is preventing you from going
backwards, you can feel it, but you
cannot see what it is.

The way north is blocked by a rusty
grating. Artificial light from the
location behind floods this place.
> ■
```

Fig. 1: Hibernated 2, un'avventura testuale commerciale di prossima uscita di Stefan Vogt, che gira su un Commodore 64 (emulato)

standard Inform sono stati sviluppati da Graham Nelson negli anni Novanta, per fornire un modo per scrivere nuove avventure testuali e compilarle in Z-code, il formato di gioco inventato dai fondatori della società di avventure testuali Infocom nel 1979. Alla fine degli anni Novanta, Inform aveva raggiunto la versione 6 ed era un linguaggio piuttosto maturo, simile al C, specializzato per il compito di scrivere avventure testuali. Anche la libreria standard era molto buona. In quel periodo, pochi erano interessati a utilizzare i lenti computer degli anni Ottanta, quindi la libreria non fu mai sviluppata o testata per quei vecchi computer. Due decenni dopo, il retro computing è diventato una cosa importante e poter utilizzare questo linguaggio e questa libreria eccellenti per scrivere giochi per computer a 8 bit era un goal da raggiungere. Purtroppo, però, il linguaggio funzionava bene ma la libreria è troppo grande e troppo lenta per essere adatta ai computer a 8 bit.

## Una libreria più piccola e più veloce

PunyInform è una libreria in cui le parti principali sono state scritte da zero, con particolare attenzione alle dimensioni ridotte e al codice efficiente. Dal punto di vista di un programmatore di giochi, è molto simile alla libreria standard di Inform 6 e molte routine sono state copiate da detta libreria, ma ogni routine è stata esaminata attentamente per vedere se fosse possibile renderla più veloce e più compatta. Mentre il gioco più piccolo possibile utilizzando la libreria standard è di 59 KB, il gioco più piccolo possibile utilizzando PunyInform è di soli 24 KB. Ha la maggior parte delle funzionalità della libreria standard, incluso un parser avanzato, un buon insieme di verbi standard e un sistema di attributi e proprietà che decidono il comportamento di ciascun oggetto nel mondo di gioco. Alcune funzionalità sono state rese opzionali, per consentire all'autore del gioco di saltare alcune parti meno necessarie per risparmiare spazio. È disponibile il supporto per l'aggiunta di NPC (persone, animali ecc. con cui il giocatore può interagire), conversazioni, porte,





oscurità, timer/fusibili (codice che viene eseguito dopo un determinato numero di turni), daemons (codice che viene eseguito ogni turno) e molto altro ancora.

Il principio alla base della libreria è che tutto ciò di cui un autore di avventure ha bisogno regolarmente dovrebbe essere fornito dalla libreria sin dall'inizio, quindi lo sviluppatore non deve generalmente scrivere codice ad-hoc, ma può anche scegliere di modificare tutte queste funzionalità e aggiungerne di nuove.

### Programmazione in PunyInform

Scrivere un gioco minimo comporta la definizione di due costanti per il nome del gioco e il titolo, una costante per la posizione iniziale del giocatore, due istruzioni "Include" per includere i file della libreria, la definizione di un oggetto per la posizione iniziale e una routine "Initialise" per stampare un file iniziale testo. Questo è tutto.

```
Constant Story      "Minimal game";
Constant Headline  "^A PunyInform demo game, by Fredrik Ramsberg.^";

Constant INITIAL_LOCATION_VALUE = Library;

Include "globals.h";

Include "puny.h";

Object Library "The Library"
  with
    description "You are in a library."
    has light;

[Initialise;
  print "^^And so the story begins...^^";
];
```

**Fig. 2: Un gioco minimalista**

Supponiamo che si voglia creare una cucina vuota e una camera da letto con un armadio. Il giocatore dovrebbe essere in grado di aprire e chiudere l'armadio e persino bloccarlo e sbloccarlo se ha la chiave giusta, può mettere delle cose al suo interno, ma esso deve risultare troppo pesante da spostare. Inoltre, posizioniamo la chiave all'interno dell'armadio. Questo è un gioco da ragazzi in PunyInform.

Utilizzando le proprietà standard (descrizione, nome, with\_key ecc.) e gli attributi (apribile, bloccabile, luce ecc.) si può far funzionare la maggior parte delle cose senza necessità di scrivere codice particolare. Tuttavia, quando si ha un oggetto che non può essere definito in questo modo, si può aggiungere codice all'oggetto per

```
Object Kitchen "Kitchen"
  with
    description "Everything is painted blue.",
    e_to Bedroom,
    has light;

Object Bedroom "Bedroom"
  with
    description "It's your bedroom.",
    w_to Kitchen,
    has light;

Object -> Cupboard "cupboard"
  with
    description "It's heavy wooden cupboard.",
    name 'heavy' 'wooden' 'cupboard',
    with_key SmallKey,
    has openable lockable container static;

Object -> SmallKey "small key"
  with
    name 'small' 'key';
```

**Fig. 3: Utilizzo di proprietà e attributi**

farlo reagire nel modo desiderato. Ciò consente un'incredibile flessibilità. Supponiamo che si voglia che la porta dell'armadio sia bloccata all'inizio del gioco, quindi l'apertura fallirà al primo tentativo ma sarà possibile al secondo.

"General" è un attributo che si può utilizzare per i motivi più disparati. Si tratta di una variabile su misura, definita automaticamente in locale per ogni oggetto. In questo caso significa "L'armadio si aprirà facilmente?" Ecco come appare il gioco risultante in Windows Frotz, uno degli interpreti Z-code più popolari.

### Documentazione

PunyInform viene fornito con la documentazione completa. Dato che è molto simile alla libreria standard di Inform 6, si può iniziare leggendo il Manuale ufficiale di Inform Designer, 4a edizione (DM4), e poi dare un'occhiata al manuale di PunyInform per vedere cosa è diverso. Oppure, ovviamente, si può sempre immergersi direttamente nella programmazione e cercare volta per volta le cose che ci

```
Object -> Cupboard "cupboard"
  with
    description "It's a heavy wooden cupboard.",
    name 'heavy' 'wooden' 'cupboard',
    with_key SmallKey,
    before [;
      Open:
        if(self hasnt general) {
          give self general;
          "The door seems to be stuck. You tug at it hard,
            but it only opens a tiny bit. Maybe if you
            try again?";
        }
    ],
    has openable lockable container static;
```

**Fig. 4: Aggiunta di una regola "before"**





```

Windows Frotz
File View Help
Score: 0 Moves: 4
Bedroom
And so the story begins...

Minimal game
A PunyInform demo game, by Fredrik Ramsberg.
Release 1 / Serial number 230831 / Inform v6.41 PunyInform v4.7 R

Kitchen
Everything is painted blue.

> e
Bedroom
It's your bedroom.

You can see a cupboard (which is closed) here.

> open cupboard
The door seems to be stuck. You tug at it hard, but it only opens a tiny
bit. Maybe if you try again?

> again
You open the cupboard, revealing a small key.

> look
Bedroom
It's your bedroom.

You can see a cupboard (which contains a small key) here.

> █

```

servono man mano che si procede.

## Conclusioni

In questo primo articolo abbiamo analizzato le basi del linguaggio Inform6 e dalla libreria PunyInform, vedendo come, con una struttura snella e pochi semplici comandi, sia possibile costruire un'avventura testuale. Nel prossimo tutorial ci concentreremo su come organizzare ed inizializzare un ambiente di sviluppo. Successivamente, entremo nel vivo dell'azione sviluppando dei semplici (ma altamente riconfigurabili) giochi.

Buon divertimento!

## RISORSE:

**Main project page per PunyInform:** <https://github.com/johanberntsson/PunyInform>

**PunyInform releases:** <https://github.com/johanberntsson/PunyInform/releases>

**PunyInform manual:** <https://github.com/johanberntsson/PunyInform/wiki/Manual>

**Inform Designer's Manual, 4<sup>a</sup> edizione:** <https://www.inform-fiction.org/manual/html/>

**Un forum amichevole dove chiedere aiuto:** <https://intfiction.org/>

**Un server Discord per supporto e discussioni su PunyInform:** <https://discord.gg/zyYvb8JJE6>

**Ozmoo, un interprete Z-code utilizzabile per rendere giochi PunyInform fruibili su Commodore 64/128/Plu4 e MEGA65:** <http://microheaven.com/ozmooonline/>

**Pagina ufficiale PunyJam#4:** <https://itch.io/jam/punyjam-4>





# Omaggio a Doriath (Merlino)

di Eugenio Rapella

A partire dai primi anni ottanta, le edicole avevano cominciato a riempirsi di riviste, collegate a Vic20 – Spectrum – C64 ecc., prima con listati, poi con cassette e infine con floppy disc.

I miei primi contatti con un calcolatore elettronico, risalivano ai primi anni dell'università (siamo intorno al ... 1973!); a quei tempi non avrei proprio immaginato che, in futuro, avrei potuto disporre di un “personal computer”; proprio “personal”. Tutto mio! Da utilizzare a casa mia!

Naturalmente, appena si è reso disponibile, mi sono procurato un Vic20 e sono poi passato ad un Commodore 128 che ho quasi sempre usato in versione C64.

Mentre mi dilettao con i miei programmini (mi piaceva soprattutto utilizzare il Commodore nelle “simulazioni

stocastiche”: spesso mi inventavo un gioco “probabilistico”, determinavo la strategia ottimale e la mettevo alla prova facendo giocare al C64 migliaia di partite per controllare se la mia strategia si rivelava vincente anche ... sul campo), compravo sistematicamente qualche rivista e provavo a cimentarmi con alcuni dei giochi della cassetta o floppy allegato.

Intendiamoci, anche allora non ero un ragazzino; non ero particolarmente interessato alla maggior parte dei giochi (come gli “sparatutto”) e, di solito, giocavo solo un paio di partite prima di abbandonare il gioco, tranne che per ... **Merlino!**

All'interno della rivista n. 10, al gioco identificato con il nome di “Merlino”, erano dedicate poche righe:



La presentazione è molto approssimativa e mancano alcune informazioni essenziali. Lanciando il gioco si otteneva::

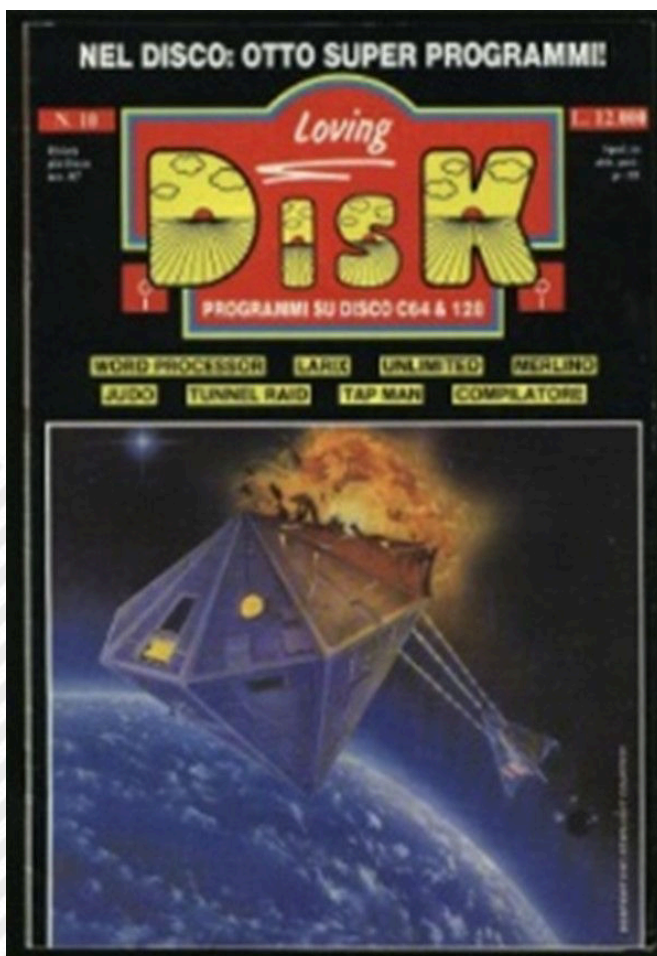
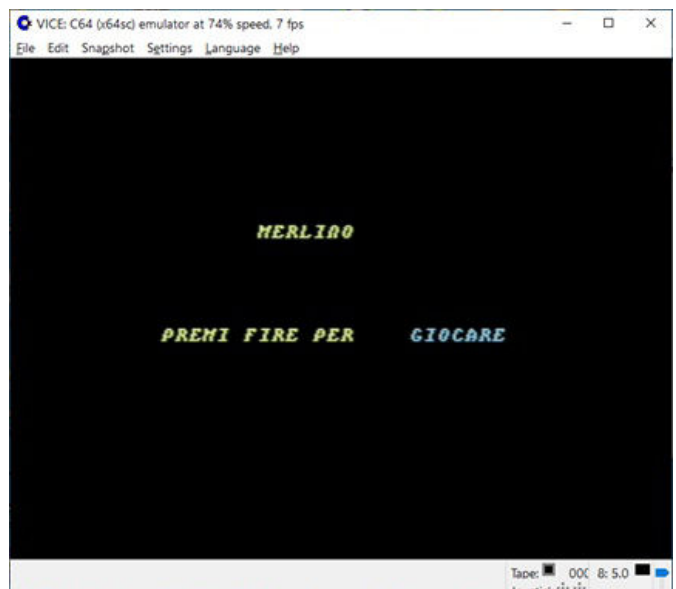
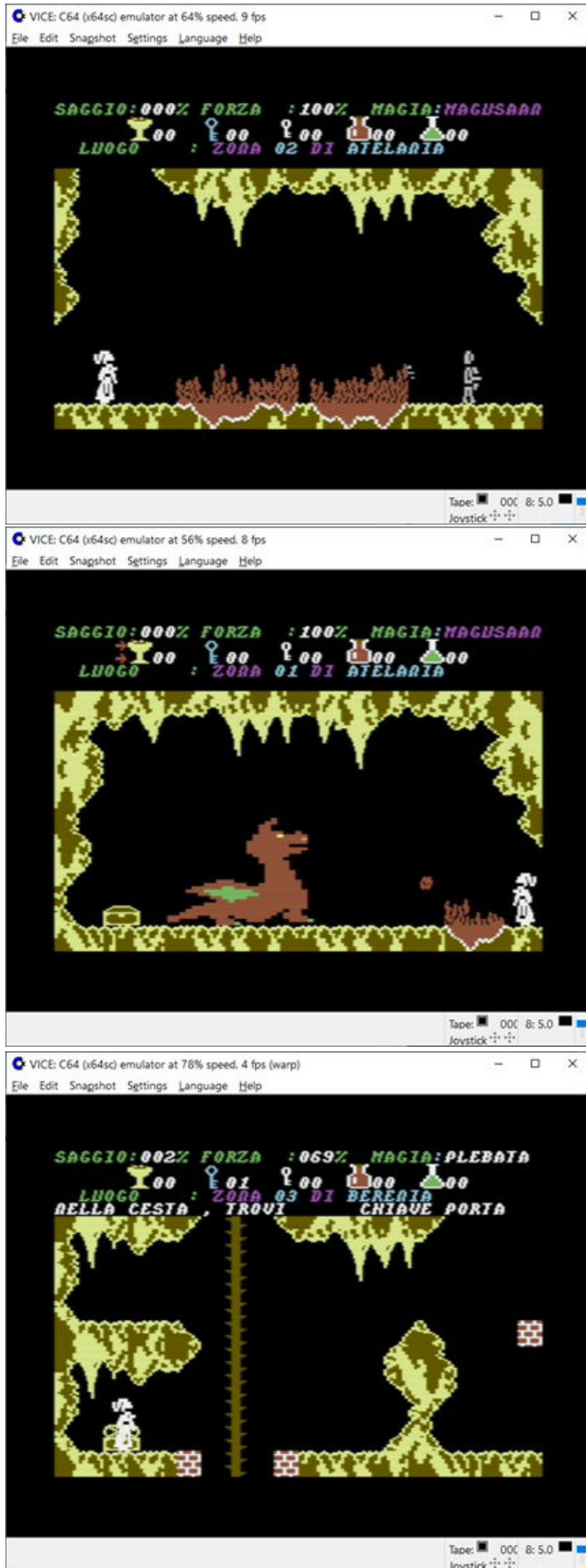


Fig. 1 - La copertina del “Loving Disk” n. 10; siamo nel 1987





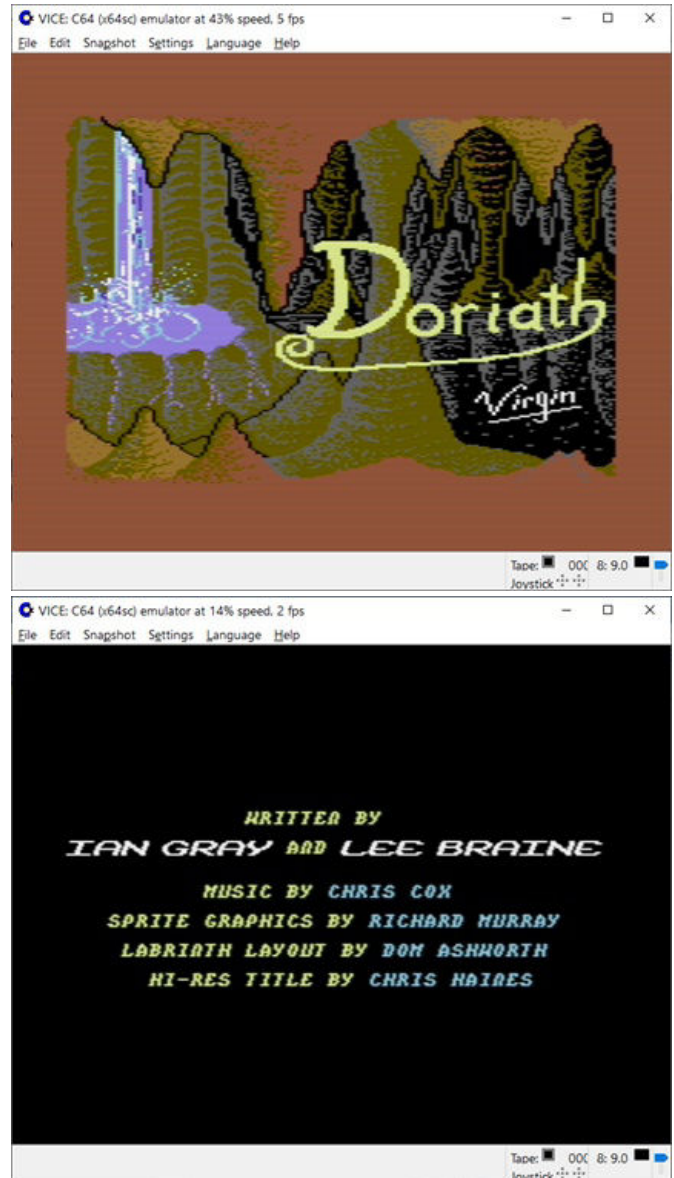
Ecco tre schermate ad inizio gioco:



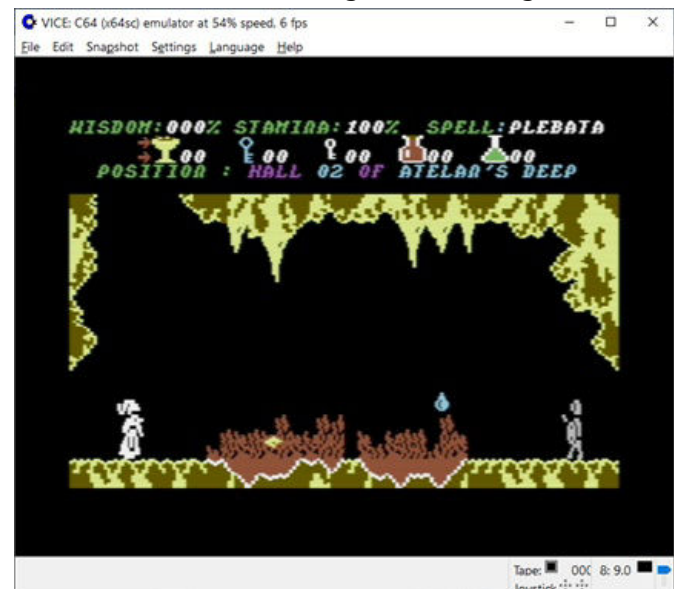
Sappiamo tutti che, ai tempi, eravamo in una specie di far west: sulle riviste dell'epoca apparivano giochi "clonati" da giochi acquistati all'estero senza che gli editori si preoccupassero troppo di diritti d'autore o cose simili.

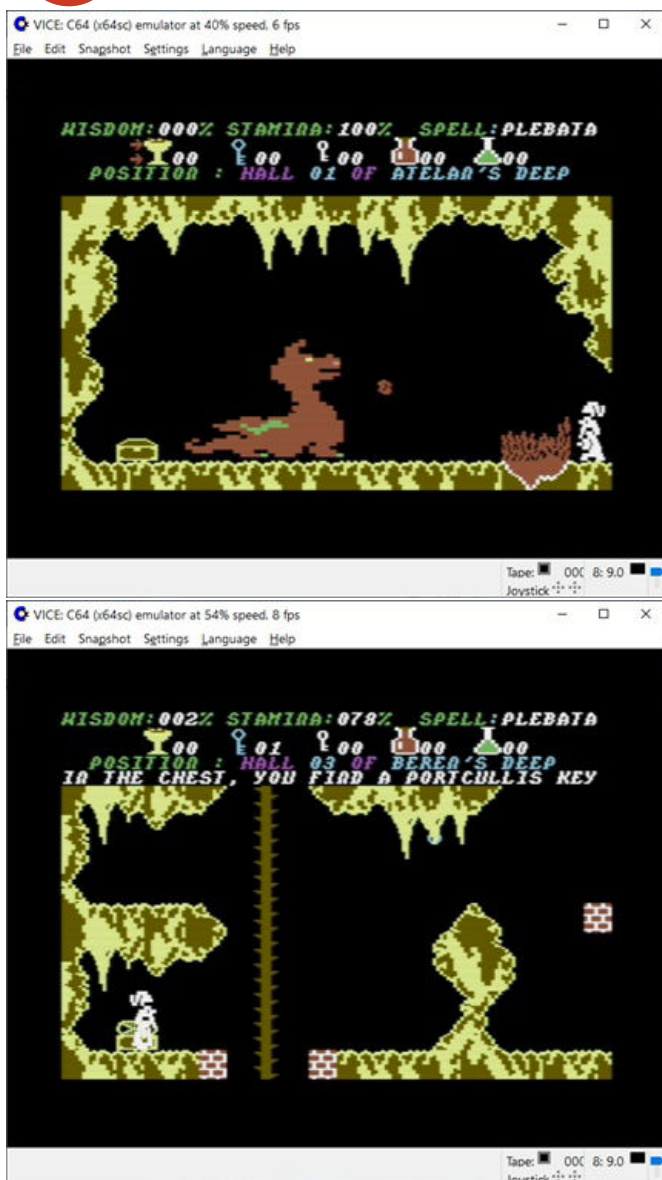
Diversi anni più tardi ho faticato un po' prima di scoprire che l'originale di Melino era ... **Doriath**.

Le schermate iniziali di Doriath, ovviamente assenti in Merlino, sono queste:



Ed ecco le tre schermate originali ad inizio gioco:





Come si nota, in Merlino le scritte in inglese sono state sostituite da quelle in italiano in modo da mantenere la stessa lunghezza.

Ma torniamo a “Merlino”: il meccanismo di gioco era interessante e la colonna sonora era ... ipnotica. Io e due miei amici ci ritrovavamo per quella che chiamavamo una “merlinata”. Occorreva accordarsi preventivamente per avere a disposizione un buon intervallo di tempo perché ... non era possibile “salvare il gioco” e, udite udite, c’era (e c’è) una sola vita a disposizione: una distrazione e bisognava ripartire dall’inizio, raccogliere da capo tutti gli amuleti indispensabili per distruggere i vari nemici. Nelle “ceste” si recuperano chiavi, pozioni ristoratrici, amuleti e anche dei misteriosi “scroll fragments” (nell’articolo “I soldati del dragone” non c’è alcun riferimento a questi scroll fragments).

Una volta, a gioco inoltrato, nel disperato tentativo di scoprire se c’era un modo per registrare su un dischetto la situazione, avevamo premuto il tasto “S” per un eventuale comando “Save”. Sorpresa! Premendo “S” compare un messaggio che, si scopre poi, prende forma man mano che ci si procura questi benedetti frammenti.

Qual è l’obiettivo del gioco? Nell’articolo associato alla rivista si parla di “uccidere il drago che vive nei ghiacci”. Anche se non è detto da nessuna parte, sappiate che dovete prima raccogliere tutti gli “scroll fragments” e raggiungere il 100% di saggezza.

Doriath mantiene il suo fascino anche oggi? Se volete provare a cimentarvi, sospendete la lettura e andate a scaricare il gioco (potete giocare a “Merlino” scaricando il “Loving Disk” n.10 da

[https://ready64.org/download/scheda\\_download.php?id\\_download=814](https://ready64.org/download/scheda_download.php?id_download=814)

oppure scaricare “Doriath” da

<https://csdb.dk/search/?seinsel=releases&search=Doriath> se scegliete “Doriath +15”, avete anche la possibilità di ... barare);

disegnate voi stessi la mappa che si viene a creare man mano che procederete nel gioco. Il Joystick va in porta 2. Per passare da una “magia” all’altra (segnalate in alto a destra) si utilizza la barra spaziatrice. Le magie disponibili sono scritte in colore bianco; all’inizio c’è solo quella denominata “Plebata”.

Per utilizzare una magia disponibile si sposta il joystick verso il basso premendo “Fire”: la magia compare e, mantenendo attivo il Fire, la si posiziona sull’avversario (avversari di diverso tipo richiedono magie differenti) rilasciando poi il Fire.

La doppia freccia rossa indica l’elemento utilizzabile fra quelli segnalati nella seconda riga; si passa da uno all’altro premendo F1 e lo si utilizza premendo F7.

Il primo dei cinque elementi, il calice, segnala le pozioni ristoratrici a vostra disposizione: bevendone una, la vostra forza tornerà al 100%. Se la forza scende a 0 % la partita



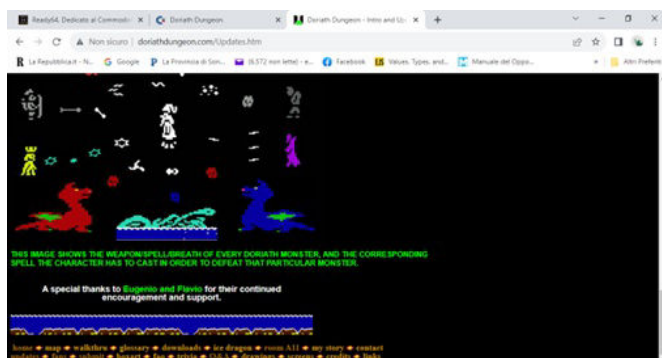


ha termine e dovrete ricominciare da zero. Occhio dunque a questa percentuale! Le pozioni ristoratrici disseminate sono numerose: non è il caso di lesinare. Come si diceva prima, premendo S si leggono gli scroll fragments fino a quel punto ritrovati.

In realtà ci sono anche un po' di ... misteri ("niente Quasilin Amulet?"; "come si accede alla parte destra della mappa?"; "stanza A11 irraggiungibile?"), ma i giocatori della prima ora ne sono venuti a conoscenza ... giocando.

In ogni caso, prima o poi, dovete assolutamente visitare: <http://www.doriathdungeon.com/> dove troverete tutto, ma proprio tutto, quello che riguarda Doriath.

C'è anche una sezione "fan submission" con i contributi di appassionati da varie parti del mondo; c'è anche qualcosa da parte mia e di mio figlio Flavio (il vero esperto di Doriath in famiglia):



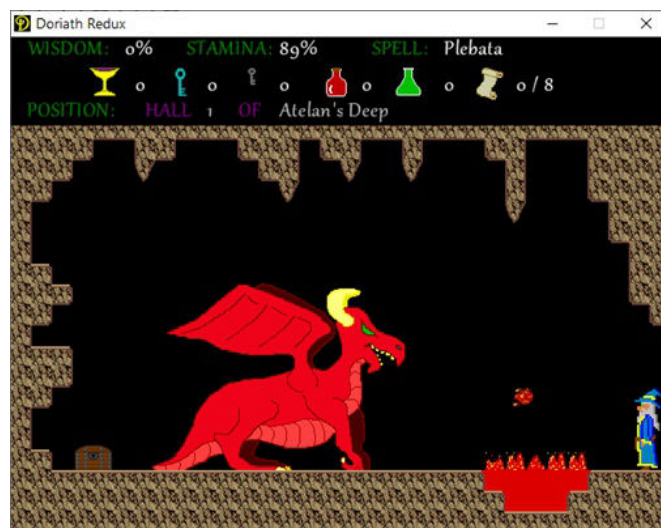
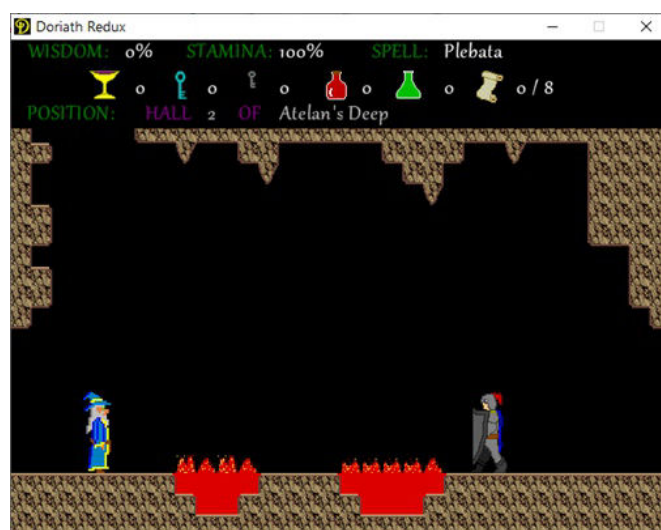
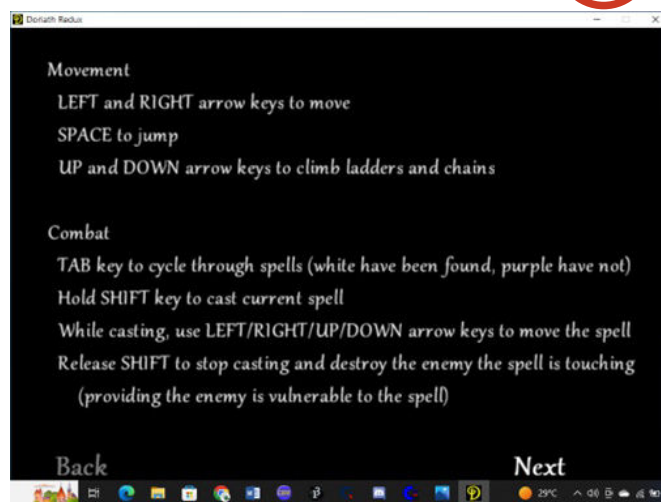
Su <https://www.lemon64.com/game/doriath> sono disponibili gli articoli dedicati a Doriath apparsi sulle riviste dell'epoca, si può ascoltare e scaricare la colonna sonora, leggere commenti e molto altro.

Su YouTube è visibile un longPlay di oltre due ore (ma Doriath si può risolvere in meno tempo):

<https://www.youtube.com/watch?v=RPMYIXvsaJM&t=339s>

Un porting di Doriath per PC è **Doriath Redux** facilmente recuperabile da Internet: <https://gamejolt.com/games/doriath-redux/87317>

Ecco alcune schermate di Doriath Redux:



Non posso concludere questo omaggio a Doriath senza proporvi la mappa realizzata, ai tempi, da Luigi, uno dei tre partecipanti alle "merlinate".

Non è un capolavoro?





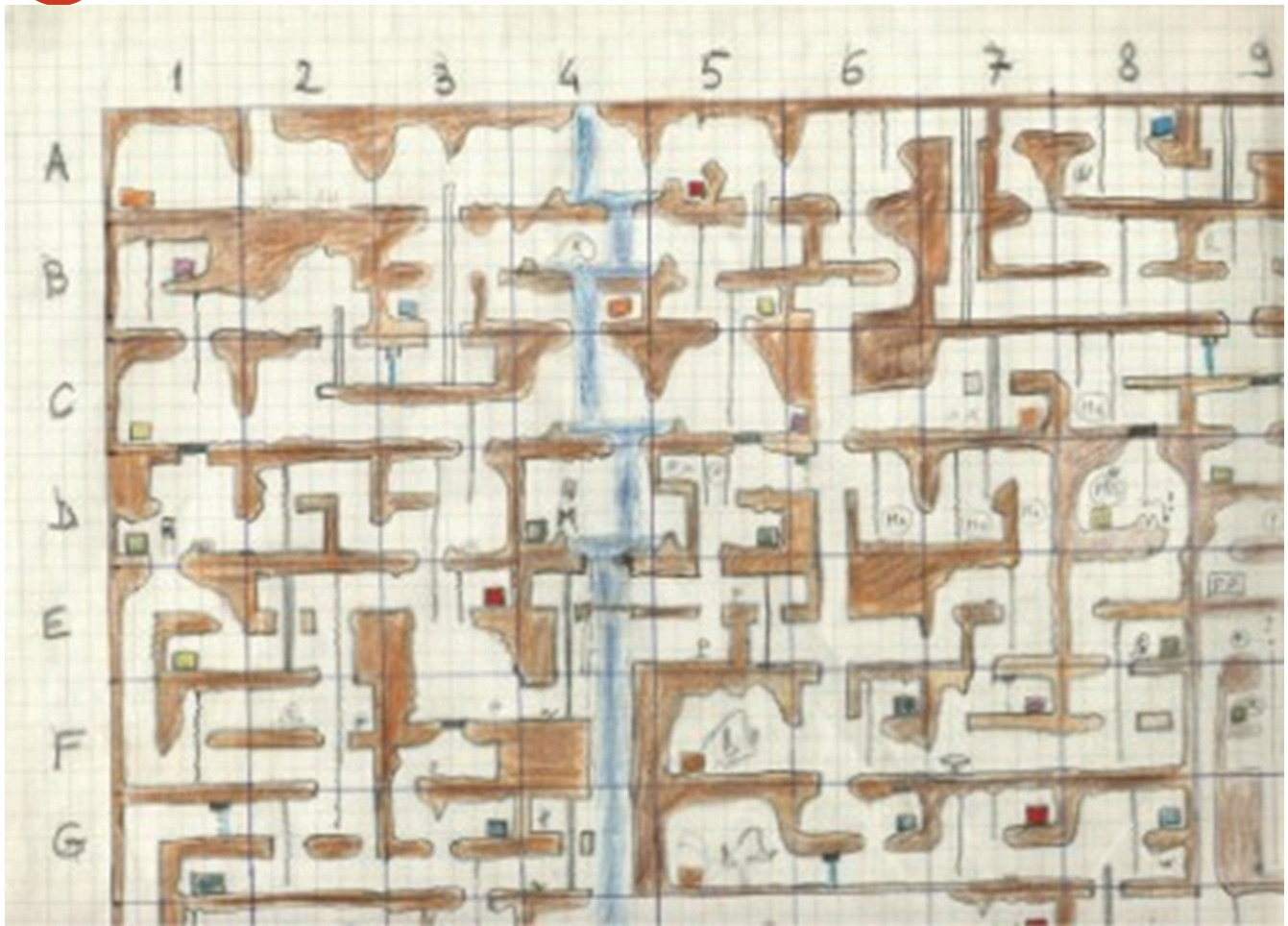


Fig. 2 - Angolo superiore sinistro della mappa

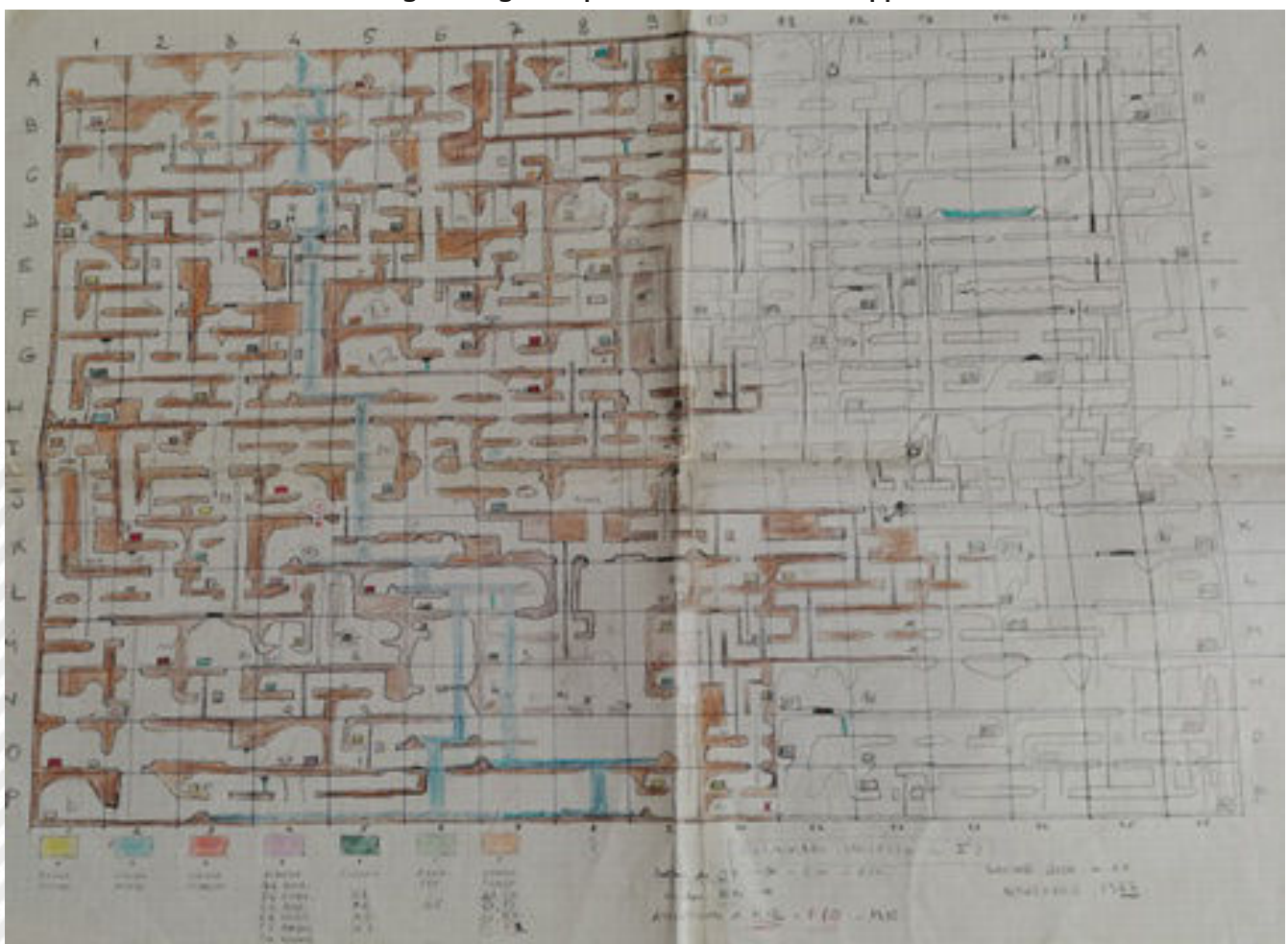


Fig. 3 - La mappa completa realizzata da Luigi (sono ben  $16 \cdot 16 = 256$  ambienti!)





**NEW GAME**

# DRACULA X: NOCTURNO IN THE MOONLIGHT – ULTIMATE

Anno: 2024

Editore.: Konami/Medusa Team

Genere: Metroidvania

Piattaforma: Sega Saturn

Sito web: [http://meduza-](http://meduza-team.ucoz.net/news/dracula_x_nocturne_in_the_moonlight_ultimate_sega_saturn/)

team.ucoz.net/news/dracula\_x\_nocturne\_in\_the\_moonlight\_ultimate\_sega\_saturn/2024-01-01-48

Castlevania: Symphony of the Night è uno dei gioielli nel catalogo della prima Playstation.

Pubblicato nel 1997, riprese il gameplay della saga di Castlevania e lo combinò abilmente con l'esplorazione non lineare di Super Metroid aggiungendo elementi tipici del gioco di ruolo con tanto di punti esperienza ed equipaggiamento.

Il risultato finale fu una rivoluzione che cambiò un genere, anzi ne inventò uno nuovo, il "Metroidvania".

Ma non si trattava di un'esclusiva per il sistema Sony; Konami lo pubblicò anche per il Sega Saturn, completo di tutto e con alcuni livelli aggiuntivi, ma le scarse vendite della console in occidente fecero sì che non venisse mai rilasciato al di fuori del Giappone. In un certo senso, è stata una sorta di benedizione in quanto la versione Saturn presentava alcuni difetti e mancanze rispetto a quella Playstation, a causa di un porting gestito dallo studio di Nagoya di Konami, realizzato con molta fretta.

Ora però è qui tra noi! Abbiamo una versione in lingua inglese e con alcune migliorie davvero eccellenti.

La community del retrogaming ha finalmente trovato un modo per rendere questo gioco fruibile a chi non conosce il giapponese tramite una bella e corposa patch che non solo traduce il testo, ma aggiunge alcuni elementi migliorativi davvero necessari.

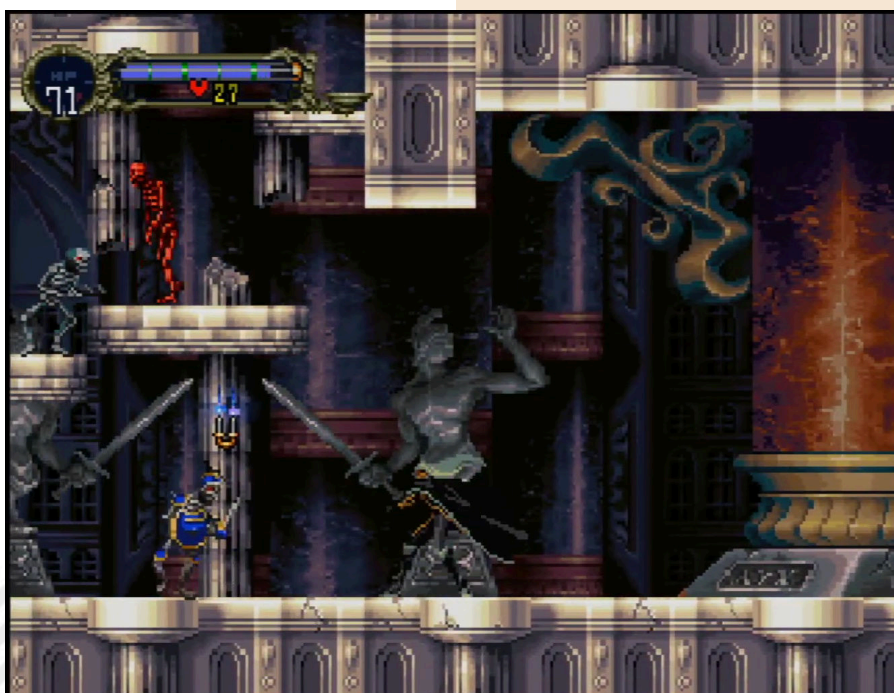
Il progetto di traduzione ha richiesto molto tempo, ma la versione

"definitiva" vale la pena di essere attesa e testata.

Utilizzando la versione PSP come modello, questa patch per Dracula X: Nocturn in the Moonlight (questo il titolo giapponese del gioco) riduce i tempi di caricamento, corregge alcuni evidenti bug e consente persino di utilizzare la cartuccia RAM da 4 MB di espansione del Saturn.

Sono stati aggiunti tutti gli effetti di trasparenza che mancavano nel port originale, ridimensionati gli sprite di protagonisti e sistemate le animazioni.

Il gioco è sempre lo stesso. Utilizza un gameplay bidimensionale a scorrimento laterale. L'obiettivo è quello di esplorare il castello di Dracula per abbattere l'entità chiamata Shaft che controlla il corpo di Richter Belmont, autoproclamatosi signore del castello ed eroe assoluto.





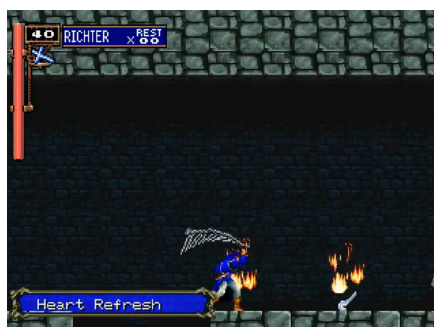
Una volta sconfitto è possibile accedere alla seconda porzione del gioco, il Castello Rovesciato, una versione capovolta con nuovi nemici e boss.

Il gioco non lineare, con la maggior parte del castello inaccessibile finché non vengono raccolti i vari oggetti e abilità, rimane sempre coinvolgente. Un titolo da riscoprire se siete possessori di Saturn o se volete provarlo in emulazione.

La patch va applicata ad una rom del gioco giapponese ed è ovviamente consigliato il possesso del titolo originale.

Ora non ci resta che attendere anche il demake/port per Sega Megadrive di questo gioiello, ma intanto godiamoci questo super lavoro.

di Carlo Nithaiah Del Mar Pirazzini



## GIUDIZIO FINALE



### » Giocabilità 90%

Un titolo che ha aperto un genere che ora è possibile giocare anche su Saturn in una bella veste comprensibile e migliorata.

### » Longevità 95%

Difficile ma appagante l'esperienza di gioco.





**NEW GAME**

# BRILEY WITCH CHRONICLES 2

Anno: 2023

Editore/Sviluppatore: Sarah Jane Avory

Genere: JRPG

Piattaforma: Commodore 64

Sito web: [https://](https://sarahjaneavory.itch.io/briley-witch-chronicles-2-c64)

sarahjaneavory.itch.io/briley-witch-chronicles-2-c64

Ritorna la saga della streghetto Briley e ritornano i personaggi del villaggio di Meapole.

Qualcosa di funesto sta per accadere nella vita della giovane strega e purtroppo il suo destino verrà cambiato per sempre.

Presentato il 1° dicembre, Briley Witch Chronicles 2 si tratta del seguito diretto del premiato primo capitolo realizzato da Sarah Jane Avory nel 2021.

La saga videoludica si rifà alla serie di libri Briley Witch Chronicles Book, e nel caso specifico di questo secondo episodio, ai volumi 5 e 6.

In questa nuova avventura ritroviamo Briley e il suo amico Smokey molto cresciuti. La nostra eroina è ora un'esperta maga.

Il gioco parte in effetti con un bel po' di punti esperienza in più (ben 55000) rispetto al capitolo 1 che ci vedeva giovani neofite, e avremo a disposizione un grimorio ricco di incantesimi e 160 punti magia. Una bella quantità di mana che ci permetterà di affrontare la terribile accusa che ci verrà posta di fronte durante lo svolgersi della storia.

Il gioco parte con un'introduzione ben realizzata e piuttosto lunga che ci spiega cosa sta accadendo e subito ci mette in gioco come... panettiera! Ebbene sì! Si comincia consegnando pane attraverso il villaggio fino ad arrivare al culmine della storia.

Tutto questo mentre ci ritroverà a combattere con ragni, serpenti e mostri di vario tipo.

Gameplay in puro stile JRPG, come il predecessore, con scontri dove potremo selezionare il tipo di attacco, la difesa e la magia ricordandoci



tantissimo i primissimi Final Fantasy. Come in ogni titolo del genere che si rispetti anche qui sono aspetti classici come i negozi di equipaggiamento, magici e le immancabili quest secondarie.

Il motore del titolo è lo stesso del precedente ma con alcune migliorie





## GIUDIZIO FINALE



## » Giocabilità 91%

Stesso gameplay del primo capitolo e una storia decisamente intrigante.

## » Longevità 90%

Più difficile del primo episodio ma con un grado di sfida bilanciato.



come il secondo pulsante per aprire il menù (implementato anche nel primo capitolo in un aggiornamento).

L'aspetto grafico è sempre piacevole e ben animato così come la colonna sonora azzeccata al contesto di gioco. Dimenticavo... è compatibile Pal e NTSC e gira su tutti i possibili tipi di configurazione per Commodore 64 senza intoppi.

Rispetto al primo capitolo il livello di difficoltà è leggermente più alto. Questo dovuto anche a come la nostra amata streghetta parte all'inizio del gioco.

E' tutto proporzionale al livello di potenza iniziale.

Titolo in formato crt (cartuccia) che non si discosta molto dal precedente ma che offre qualche ora di svago e avventura.

Un titolo davvero ben fatto che fa comprendere quanto sia abile (per chi ancora non se lo ricordasse) la prode Sarah Jane Avory dietro ad una tastiera. Da comprare al volo!

di **Giampaolo Moraschi**





**NEW GAME**

# YETI MOUNTAIN

Anno: 2024

Editore/Sviluppatore:

Protovision

Genere: Action Multi-Genere

Piattaforma: Commodore 64

Sito web: [https://](https://protovision.itch.io/yeti-mountain)

protovision.itch.io/yeti-mountain

Yeti Mountain è un gioco d'avventura ed esplorazione multigenere, come tanti titoli dei tempi d'oro, sviluppato da un talentuoso team guidato da Russel Mills.

In tre episodi distinti dovremo indagare sulla scomparsa del nostro migliore amico e di altri sciatori.

Il primo episodio è un in stile RPG su missioni. Tanta interazione coi personaggi, partecipazione ad eventi sciistici e esplorazioni.

Il villaggio sciistico creato è molto interessante ed è sviluppato con molta cura per i dettagli.

Nel secondo episodio saremo impegnati in un gioco d'azione/platform con scorrimento multiplo. Entreremo nella tana dello Yeti ed exploreremo diversi ambienti. Trappole e fauna ostile a gogo!

L'episodio finale invece ci farà scendere sulle piste da sci fino ai piedi della montagna dello Yeti. Durante la discesa saremo coinvolti in diversi enigmi ed esplorazioni.

C'è un po' di tutto in questo titolo: minigiochi sugli scii, piattaforme, esplorazione, puzzle, mostri, trappole, ecc... ecc...

La narrazione mi ricorda alcune storie di stampo Lovacraftiano. E' scritta molto bene e scorre liscia e gradevole attraverso a numerose scene di intermezzo ben realizzate.

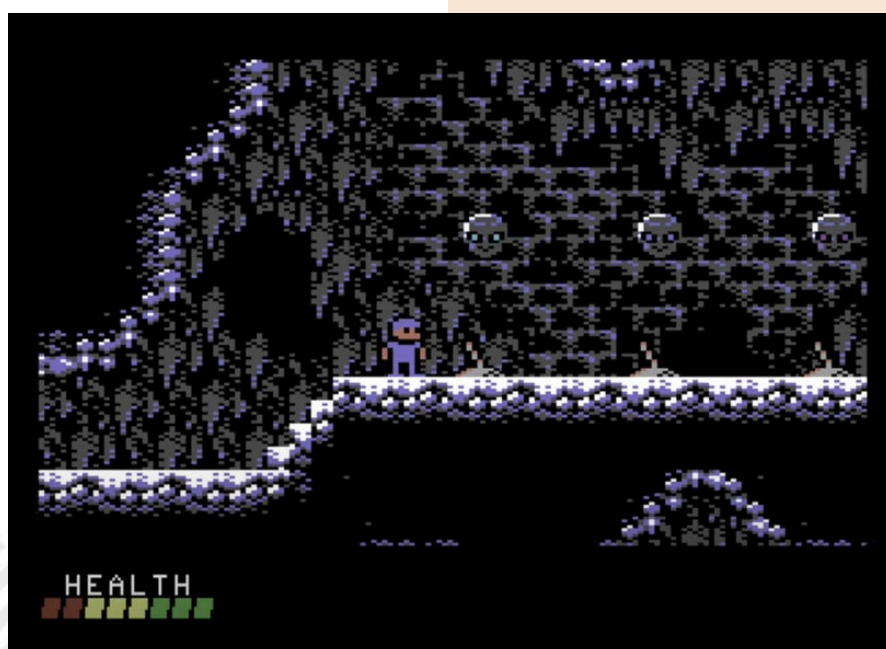
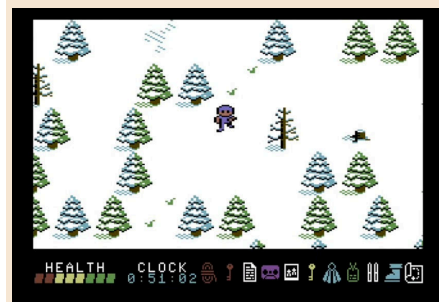
Il file è in formato CRT e comprende la possibilità di salvataggio in gioco. E' compatibile PAL e NTSC e costa circa 9 dollari nello store.

Sono soldi ben spesi perché Yeti Mountain è un titolo divertente. Mi



sono divertito per la narrazione e per lo stile di gioco. Non amo moltissimo i multigenere, ma qui c'è qualità.

Qualità nella storia, qualità nel comparto grafico per un buon uso dei colori e delle animazioni e tanta qualità nella colonna sonora con 21 brani realizzati con cura.





## GIUDIZIO FINALE

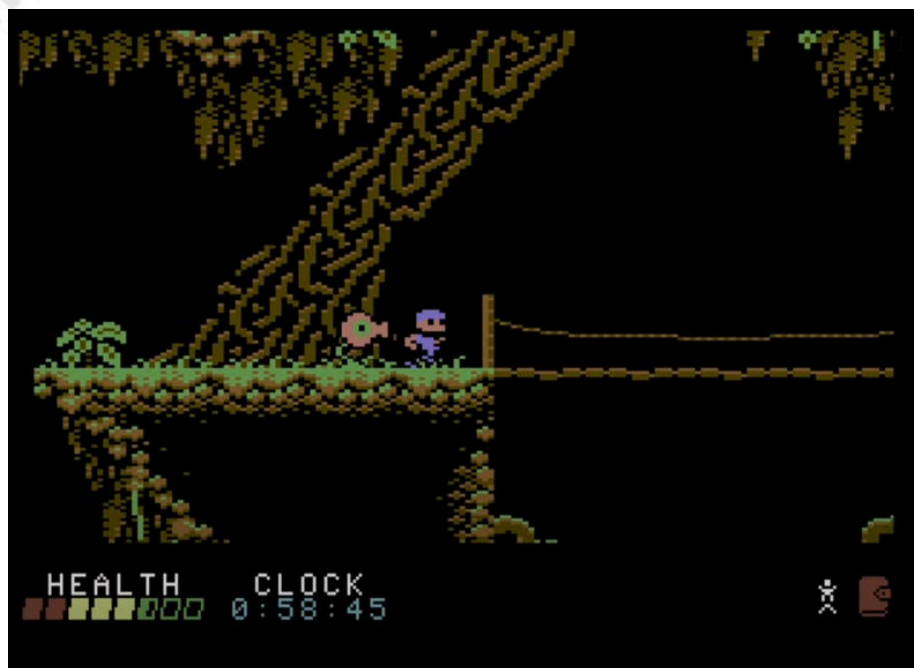


## » Giocabilità 90%

Tre differenti scenari di approccio e tantissimi generi di gioco in un solo titolo. Bellissime le sezioni di esplorazione.

## » Longevità 90%

E' un titolo impegnativo ma mai snervante. La narrazione è davvero eccellente.



Le missioni secondarie tengono alto l'interesse durante il gioco, altro punto a favore.

C'è varietà nello stile di gioco. Ad esempio nelle discese sugli sci, accumulando punti attraverso complicati balzi e acrobazie, oppure partecipando attivamente alle competizioni e agli eventi.

Insomma, siamo di fronte ad un bel prodotto che merita di essere acquistato in formato digitale e, quando uscirà, anche in formato cartuccia.

di Carlo Nithaiah Del Mar Pirazzini





**NEW GAME**

# CORESCAPE

**Anno:** 2024

**Sviluppatore:** drMortalWombat

**Genere:** Shoot em up

**Piattaforma:** Commodore 64

**Sito web:** [https://](https://drmortalwombat.itch.io/coreescape)

[drmortalwombat.itch.io/  
coreescape](https://drmortalwombat.itch.io/coreescape)

Avvertenza: questo gioco non è facile, non rende sereni. E' stimolante ma è dannatamente frustrante. Non si può affrontare senza conoscerne a memoria ogni singola wave. Servono riflessi pronti, manualità... e l'utilizzo di alcuni trainer contenuti appositamente nel titolo.

Corescape è uno sparattutto per hardcore gamers della prima guardia. Abituati a battaglie spaziali con difficoltà immonde. Ad inserire gettoni su gettoni nei cabinati per poter vedere i livelli successivi.

È così dannatamente difficile... da essere bello!

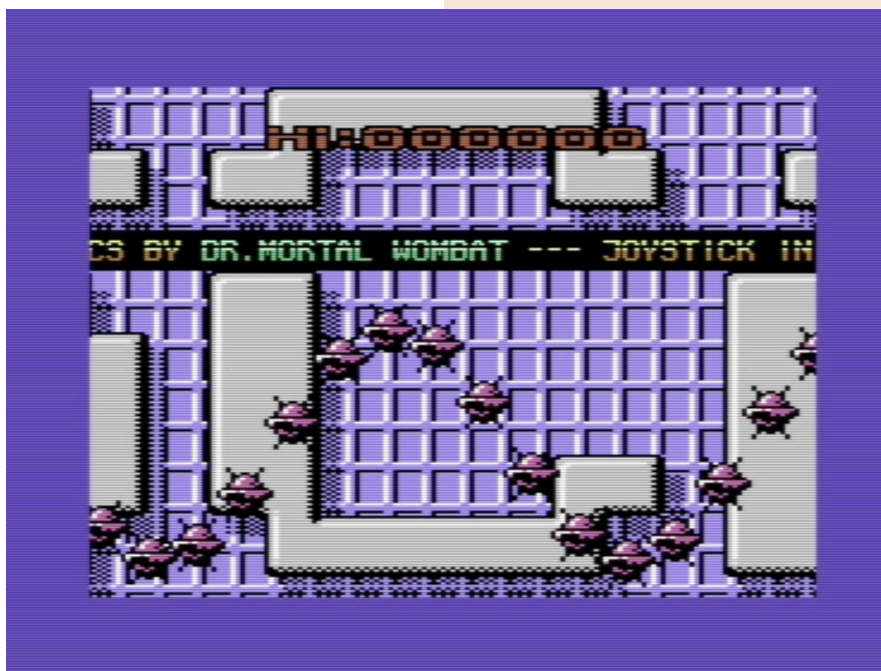
Non è il più bello ma il suo level design intriga e ti fa giocare e rigiocare spesso. Il vero problema di questo gioco non sono i nemici, ma le barriere e gli elementi dello sfondo.

Sono indistruttibili e richiedono una forte memoria visiva e un grande sangue freddo per essere superati. Non potete immaginare le imprecazioni che ho rilasciato ogni volta che morivo malamente contro un elemento del livello.

Sono otto livelli e sono tutti una grande impresa da portare a termine.

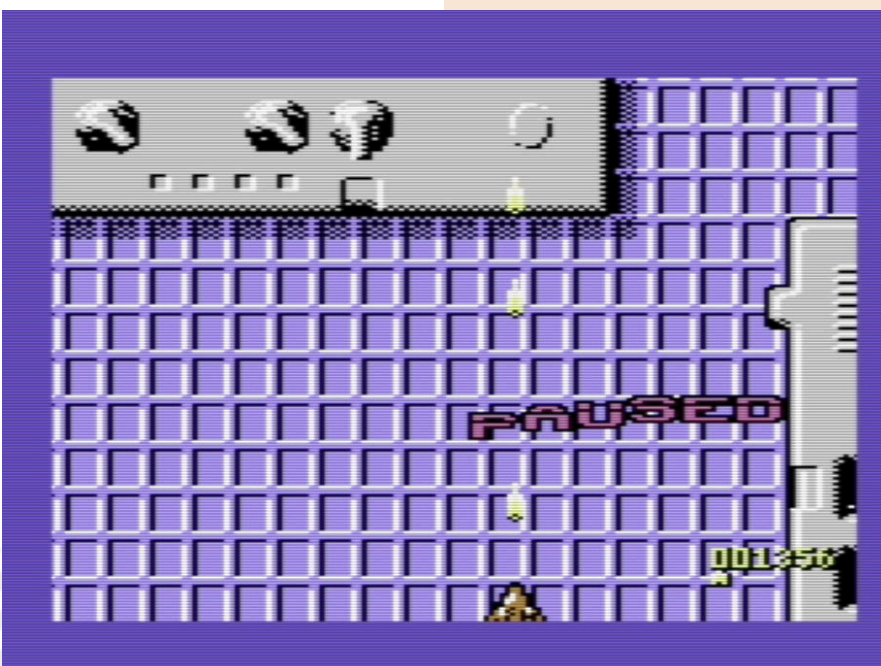
Se vi piacciono le sfide "quasi" impossibile Corescape è il gioco perfetto per voi.

Ho apprezzato non solo lo stile brutale



del design di gioco, ma anche l'aspetto tecnico.

Veloce, ben animato con un aspetto grafico "metallico" che mi ha ricordato alcuni classici del passato.







## GIUDIZIO FINALE

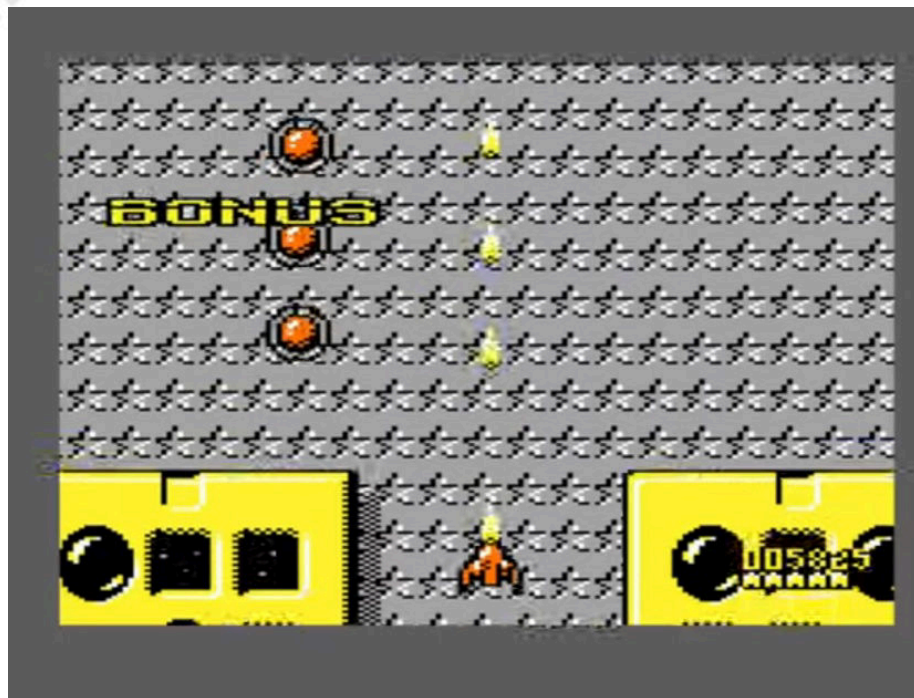


## » Giocabilità 80%

La struttura dei livelli è intrigante e il sistema di controllo perfetto.

## » Longevità 65%

Personalmente gli avrei dato 90%, ma devo essere equo. È un gioco davvero difficile e adatto ai veterani delle sfide impossibili.



I livelli non sono vari, ma sono belli da vedere e curati.

rallenta mai. Un chiaro esempio di buon lavoro.

Il Sid suona alla grande, pompando una colonna sonora decisa ed orecchiabile condita da buoni effetti sonori.

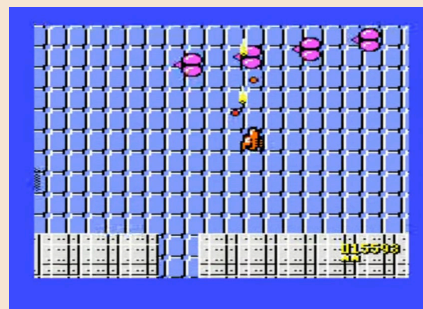
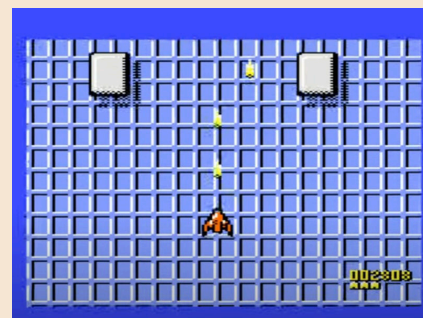
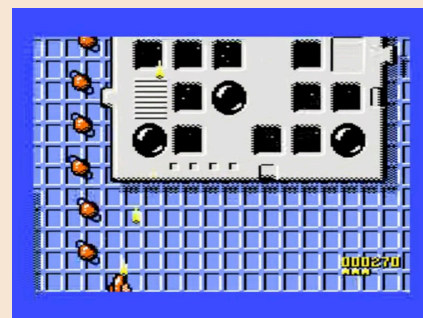
Ripeto l'avvertenza iniziale: non è un gioco per tutti. Ci voglio attributi e coraggio per portarlo a termine.

Promosso

Il file è in formato prg compatibile su tutti i sistemi possibili ed è progettato e testato PAL, ma gira leggermente più veloce su Commodore NTSC.

di **Giampaolo Moraschi**

Si spara tanto, non ci sono potenziamenti ed è pieno di cose sullo schermo. Non





# MAGICAL POP'N

Anno: 1995

Editore/Sviluppatore: Pack-in-video/Polestar

Genere: Platform/Metroidvania

Piattaforma: Super Famicom/  
Super Nintendo

Mi piace scandagliare le librerie di importazione per le vecchie console, rimango continuamente sbalordito dall'enorme numero di gemme nascoste di cui la maggior parte delle persone non ha mai sentito parlare. Titoli su licenza di Gundam, mille pubblicazioni di giochi di ruolo mai localizzati, bizzarri giochi sportivi. Ma sono le pubblicazioni degli editori più piccoli ad essere interessanti.

Pack-In-Video non è certamente un nome famoso che associo a software di alto livello. Quindi è ironico che abbiano pubblicato uno dei migliori giochi per il 16 bit Nintendo di tutti i tempi.

Magical Pop'n è un titolo fenomenale che merita di essere nella stessa bacheca di Super Mario World o Metroid ed è un titolo da non perdere. Il modo migliore per descrivere Magical Pop'n è quello di associarlo ad una strana combinazione tra Mario, Metroid e Zelda, in poche parole un Metroidvania.

Si tratta di un mix leggendario e il gioco riesce ad estrarre gli elementi giusti da ciascuno per creare qualcosa di unico.

La protagonista, una tenera e paffuta principessa, è un personaggio estremamente agile e si muove a ritmo sostenuto. È armata di una spada che può usare in quasi ogni direzione.

Come in Zelda, anche qui abbiamo un inventario di armi secondarie e incantesimi da utilizzare. Come in un bel action che si rispetti, si parte con una semplice arma ma si può potenziare tutto al massimo: attacchi, bombe, arrampicate, spin attack stile Sonic... Insomma c'è tanta varietà.

La connessione con Metroid è ben



radicata nella struttura del gioco. Sebbene sia suddiviso in sei fasi con tanto di boss intermedio e finale, ciascuna è piuttosto ampia e con segreti ed oggetti da trovare in qualsiasi momento.

Ogni livello contiene un nuovo incantesimo o oggetto necessario per superare i blocchi e/o potenziarsi





## GIUDIZIO FINALE

### » Giocabilità 95%

Vario, elegante, graficamente bello.

### » Longevità 90%

Lungo al punto giusto per essere apprezzato al meglio.



ulteriormente. A differenza della serie Zelda, queste abilità non sono cruciali per sconfiggere il boss di quel determinato livello, ma sono utili per poterlo affrontare con più sicurezza. Non ci sono limiti di tempo e le fasi sono abbastanza grandi da fornire un senso di “mondo aperto”, ma non così grandi da richiedere una mappa.

Vi sembrerà strano, ma mi ha ricordato molto di più un titolo per Sega Megadrive che un classico gioco per SNES. Si muove velocemente. Il ritmo di gioco è eccellente e gli ottimi controlli rendono il passaggio da un'abilità all'altra un gioco da ragazzi nel bel mezzo del combattimento.

Per coloro che desiderano una certa rigiocabilità, di solito ci sono più percorsi fino alla fine di un determinato livello. Tutto questo porta ad un finale spettacolare che lascia a bocca aperta. Le sei fasi di gioco sono di durata moderata e personalmente apprezzo questo. Non sono presenti password o funzionalità di salvataggio su batteria

perché non necessarie.

Ogni fase è perfettamente calibrata con quello che si deve fare e affrontare. Ad essere onesto, l'unica vera pecca è la sua mancata distribuzione in occidente. Il titolo fu lanciato solo per il mercato nipponico.

La lingua non incide in nessuna maniera sulla giocabilità.

Un titolo fenomenale.

di **Roberto Del Mar Pirazzini**





# THE ADVENTURE OF LITTLE RALPH

**Anno:** 1999/2007  
**Editore/Sviluppatore:** Sony/  
New Corporation  
**Genere:** Platform  
**Piattaforma:** Playstation  
(versione recensita),  
Playstation 3 e PSP

Alla fine degli anni '90, il platform 2D era un genere che stava andando fuori moda.

Super Mario 64 aveva reso il 3D mainstream e non aiutava il fatto che Sony preferisse i titoli che spingevano i giochi tridimensionali, rispetto ai titoli che si basavano ancora sugli sprite. Spesso, ad alcuni giochi venne negata la localizzazione occidentale se si trattava di un gioco basato su sprite 2D.

L'Avventura del Piccolo Ralph è uno di quei titoli che l'Occidente si è perso. Ralph ha un problema. La sua ragazza è super sexy ed è stata rapita dal malvagio Valgo; un cattivo folle e arrapato.

Come se ciò non bastasse, questo gigantesco Adone giallo ha anche trasformato il nostro eroe in un bambino. Ralph non solo deve affrontare il malvagio rivale in forma fanciullesca, ma deve dimostrare di essere un tipo abbastanza tosto sopravvivendo a otto livelli infernali che hanno vari punti di diramazione. Spada in mano, Ralph si farà strada tra robot, diavoli, maiali satanici e orsi polari...

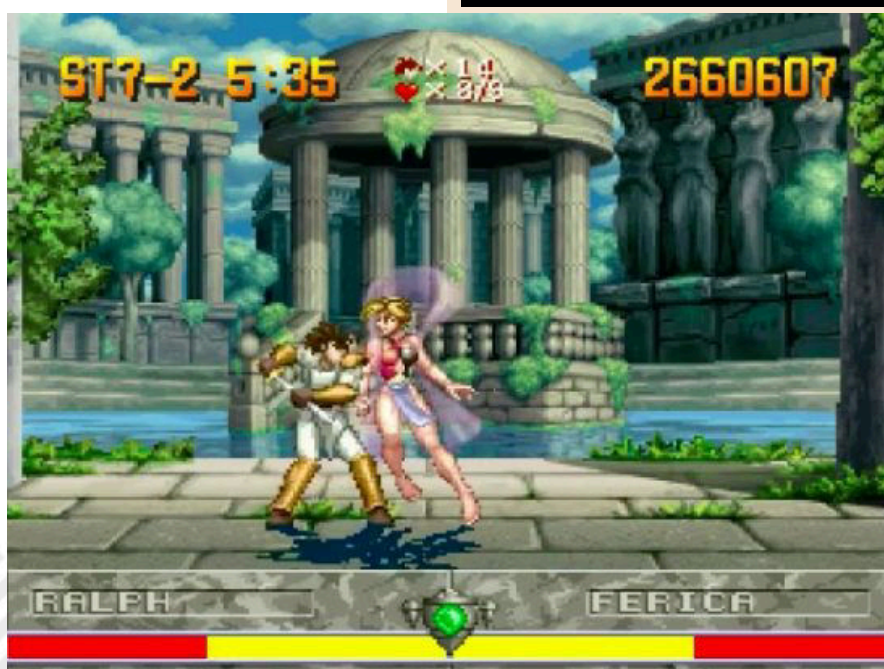
In una serie di coloratissimi e bislacchi mondi di gioco il nostro eroe si troverà impegnato nel più classico dei giochi di piattaforme ma dotato di un raffinato



2D.

Un titolo che nel 1999 passò inosservato in occidente ma che conserva un fascino tecnico meraviglioso.

I ragazzi dietro a L'avventura del piccolo Ralph hanno prestato molta





## GIUDIZIO FINALE

### » Giocabilità 90%

C'è tutto quello che serve ad un buon platform game. Controlli reattivi e tantissimo da vedere, fare e giocare.

### » Longevità 85%

Titolo ostico con alcune sfide al limite della resistenza umana, ma vi incolla al pad.



attenzione alla pixel art e alle animazioni. I personaggi sono dettagliati, i fondali ispirati da uno stile che mi ha ricordato il compianto Shotaro Ishinomori e c'è un uso magistrale dei colori.

L'estetica fa davvero da padrone con un look magistrale.

Nel 2007 fu riportato su PS3 e su PSP ma passò inosservato ancora una volta. A questo punto non mi resta che consigliarvelo in questo 2024.

Se riuscite a trovarlo in versione fisica ad un costo accettabile non fatevelo scappare.

Davvero un gioiello di grafica, sonoro e giocabilità.

di **Marta Rossman**





**NEW GAME**

# HUNTER GIRLS

**Anno:** 2023

**Editore/Sviluppatore:** PSCD Games

**Genere:** Running Game

**Piattaforma:** Sega Megadrive

**Sito web:** [https://pscdgames.com/index.php?route=product/product&product\\_id=15](https://pscdgames.com/index.php?route=product/product&product_id=15)

Girl power!!! Tre donne con abilità speciali si gettano all'avventura in un running game di ultima generazione per Sega Megadrive.

Un'avventura fantasy con tante piattaforme da saltare, abilità da utilizzare e mostri da uccidere ed evitare.

Yeeeee!!! Ok, no... in realtà Hunter Girls è un titolo che mi ha creato tantissima aspettativa e alla fine si è rivelato piuttosto banale e tecnicamente mediocre.

Eppure il PSCD team è un gruppo di sviluppatori in gamba, ma questa volta mi ha lasciato con l'amaro in bocca.

Banale nella storia e banale nelle meccaniche.

Un running game piatto che abbiamo già visto in tutte le salse.

Certo sono carini i filmati introduttivi come è interessante la musica d'accompagnamento. Posso

ammettere che persino i menù e la possibilità di settare tantissime lingue sono cose ben gradite e ben fatte ma è un gioco piatto e con un alto grado di difficoltà a causa di una non perfetta gestione dei salti e delle collisioni.

Agnes, Kim e Flora hanno tutte e tre diverse capacità. C'è chi è più abile nella difesa e chi invece è più capace nel salto.

In alto, durante il gioco, abbiamo tre indicatori: il primo ci indica il mana che una delle ragazze possiede



all'inizio, il secondo il numero di proiettili dell'arco di un'altra ragazza e il terzo la capacità di difesa.

Tutte queste abilità si esauriscono quando vengono utilizzate e quindi occorre farlo con parsimonia.

Il titolo dura in effetti una trentina di minuti. Si porta a termine con una sola condizione: quella di memorizzare ogni percorso perfettamente.





## GIUDIZIO FINALE



## » Giocabilità 50%

Un running game ripetitivo e con alcuni problemi di design.

## » Longevità 30%

Giocabile in trenta minuti dopo aver memorizzato alla follia ogni singolo salto. Decisamente folle!



Qualsiasi errore di posizionamento, di salto o di utilizzo errato delle abilità porta a morte certa. Decisamente troppo limitante e frustrante.

A tutto questo aggiungiamo un comparto tecnico non eccellente. Gli sprite animati bene ma privi di dettagli e soprattutto l'azione di gioco è "scarna". C'è poco da vedere. Si saltano i goblin, si parano le frecce, si prende il tempo per zompare sulle piattaforme per evitare i dirupi. Fondali semplici e con temi di colori autunnali molto blandi.

Insomma, non mi sono divertita e avrei preferito giocare a qualche altro titolo dei PSCD.

Il gioco è disponibile all'acquisto in formato "cartuccia" per 55 dollari sul sito che vi ho linkato in descrizione.

Cinquantacinque dollari sono decisamente tanti per così poca sostanza.

di **Marta Rossmann**





NEW GAME

# SERAPHIMA

Anno: 2023

Editore/Sviluppatore: Zosya Entertainment

Genere: Metroidvania

Piattaforma: ZX Spectrum 128k

Sito web: [https://](https://www.zosya.net/2023/11/03/new-game-seraphima/)[www.zosya.net/2023/11/03/new-game-seraphima/](https://www.zosya.net/2023/11/03/new-game-seraphima/)

Siamo di fronte al miglior gioco Spectrum di sempre? Non lo so, sinceramente non lo so.

Quello che so è che Seraphima è uno dei migliori Homebrew per questa macchina degli ultimi 10 anni e il migliore uscito nel 2023.

Un titolo sopraffino, ben sviluppato e che si lascia giocare piacevolmente. Guardiani e predoni hanno convissuto segretamente sulla Terra. I guardiani (o Grigori) proteggono la vita, mentre i predoni gli danno la caccia.

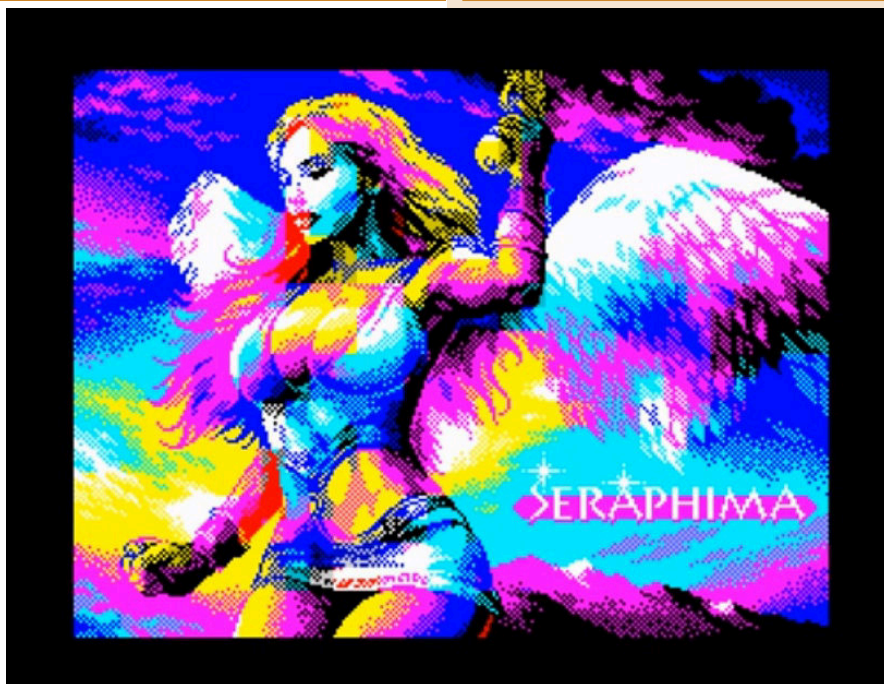
La loro eterna lotta mantiene l'equilibrio. Tuttavia, quando Seraphima emerge come ultimo guardiano, il destino della Terra diventa incerto.

I Predoni stanno eliminando tutti i guardiani.

Il compito della nostra alata e prosperosa eroina è di preservare l'equilibrio delle ere e combattere coraggiosamente per garantire la pace e la salvezza.

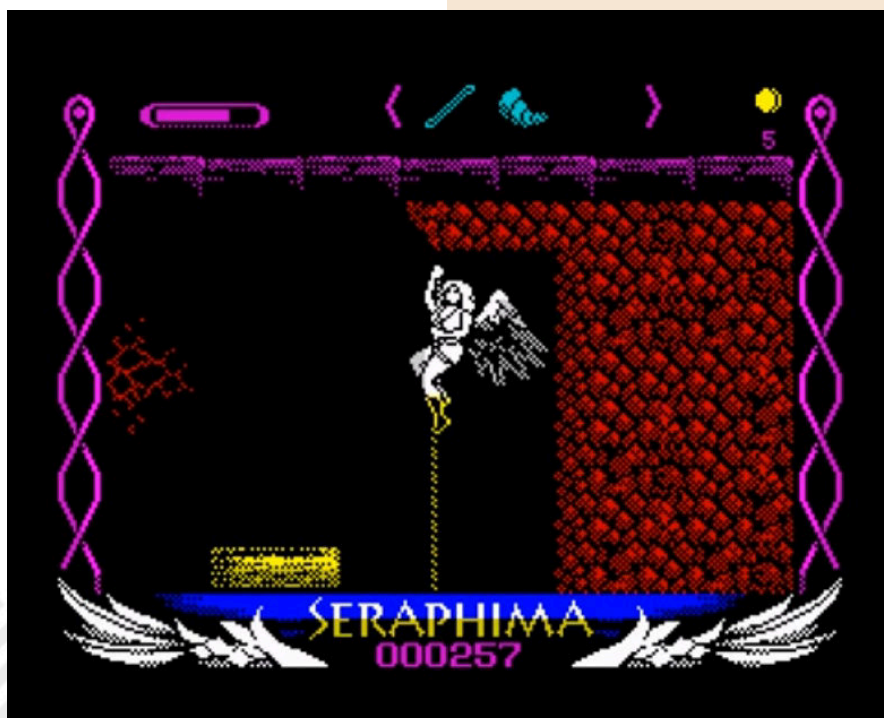
Trama (quasi) scontata a parte, Seraphima è un Metroidvania che ci invoglia ad essere esplorato e vissuto. La nostra eroina si muove inizialmente in una specie di tutorial, guidata dalla regina Madre. In questo campo di addestramento imparerà a sparare con le sue doppie pistole magiche, ad utilizzare oggetti e chiavi per aprire le porte e a scontrarsi con i bellicosi mondi che esplorerà.

Ogni mondo è collegato attraverso porte ed ogni mondo è differente in



termini di impatto e difficoltà.

Un titolo che possiede tutto. C'è una bella grafica. Animata bene, colorata bene e con il giusto tocco. Si vede la mano di un team esperto come quello di Zosya.







**GIUDIZIO FINALE** 

**» Giocabilità 95%**

Bilanciato, godibile, ottimo tutorial e ottimi comandi (consigliato il joystick ovviamente). C'è del concetto nello sviluppo dei livelli e una bella storia.

**» Longevità 95%**

E' un titolo divertente, non troppo punitivo ma che richiede attenzione per essere gustato al meglio. Dura il tempo che deve durare un titolo di questo genere e non stanca.



Lo sprite della nostra "poppata" eroina è meraviglioso per movenze ed eleganza. Lo stesso discorso vale per il compartimento audio, musicale e per gli effetti.  
Ma il punto di forza del gioco è la sua incredibile capacità di tenerti incollato all'azione.

Seraphima è bello da giocare, da esplorare e da gustare. Non è lunghissimo ma è piacevole. Un pezzo da 90 che merita di essere collezionato.

Il titolo funziona su tutti gli ZX Spectrum 128k e ovviamente in emulazione. Capolavoro!

di Carlo Nithaiah Del Mar Pirazzini





**NEW GAME**

# HERMANO

Anno: 2023

Sviluppatore: PAT MORITA  
Team

Genere: Platform

Piattaforma: Game Boy

Sito web: [https://](https://patmorita.itch.io/hermano-game-boy)

[patmorita.itch.io/hermano-game-boy](https://patmorita.itch.io/hermano-game-boy)

Durante El Dia de los Muertos, il piccolo Mano si era recato come sempre sulla tomba di suo fratello Nano.

Si dice da sempre che durante questa notte, le porte di inferno e paradiso si aprano affinché i trapassati possano ricongiungersi con i vivi.

Mentre il fratello piccolo parlava alla tomba di Nano, un terribile demone sbucò dal nulla rapendolo.

Proprio mentre Nano stava tornando in superficie.

Questa è la storia dietro ad Hermano, il nuovo platform game per Game Boy che ho appena scaricato dal sito ufficiale.

Noi saremo alla guida di Nano. Per essere un defunto non se la cava per niente male. Combatte a colpi di capocciata/teschiata contro i nemici e può fare affidamento ad alcune bombe che trova in giro per i livelli. È un titolo classico che mescola elementi alla Ghost n' Goblins con Super Mario e lo fa bene.

Piacevole da controllare e comandare e molto carino da guardare.

L'ho provato sia sul mio GB che in emulazione e devo ammettere che rende alla grande in tutti e due i casi. Il titolo presenta anche alcune simpatiche idee di gioco come la possibilità di giocare a "gravità invertita" nel sottosuolo.



Diversi livelli di gioco con numerose ambientazioni (cimitero, città





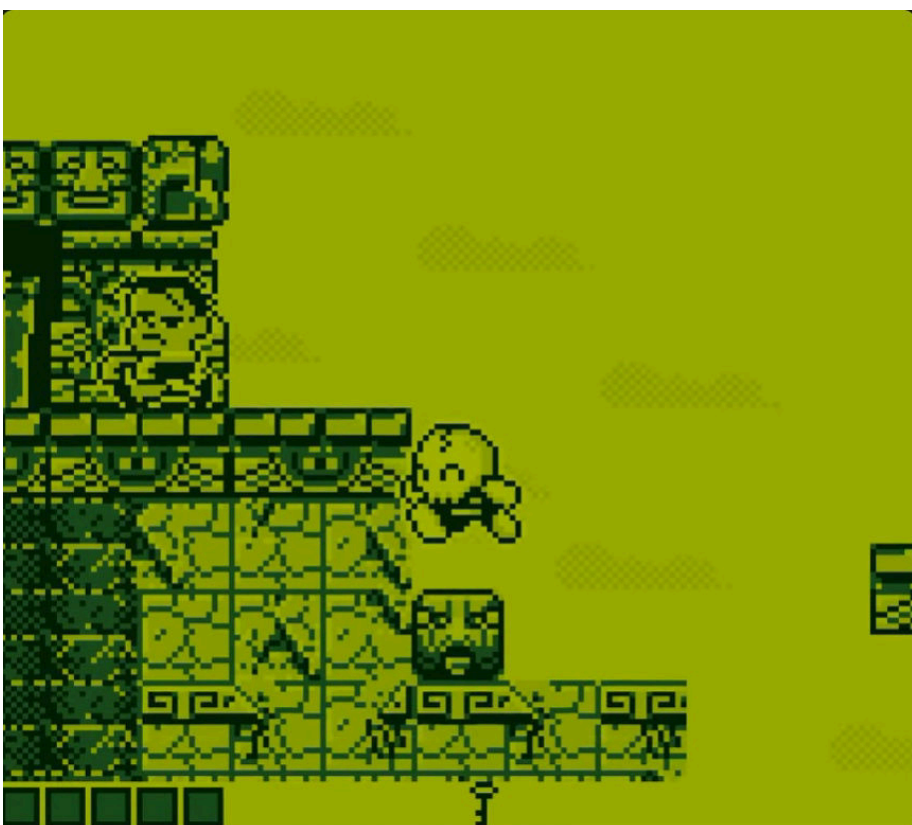
abbandonata, paludi, piramidi) e tanti mostri della tradizione da sconfiggere. Lo scopo è trovare la chiave per passare al livello successivo.

Ho letto sul sito che esiste anche una versione fisica su cartuccia che farò presto mia.

Bella la grafica super pulita e animata bene e davvero bellissimo il sonoro. Le musiche mi sono piaciute un sacco.

Un altro successo su Game Boy.

di **Marta Rossman**



## GIUDIZIO FINALE

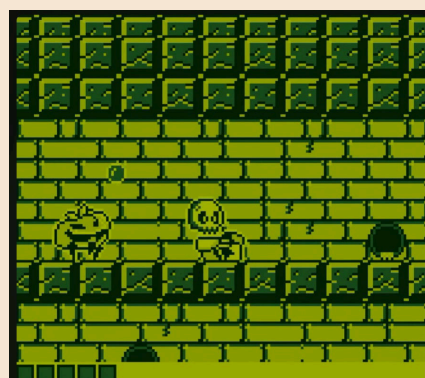
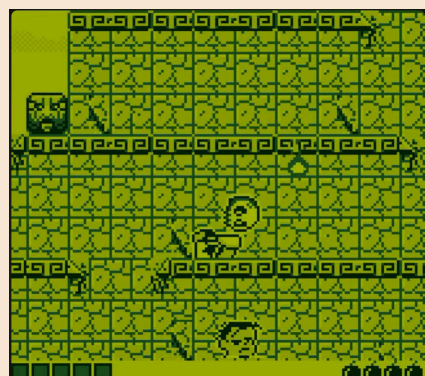


### » Giocabilità 85%

Divertente e ben strutturato. L'idea di giocare "sottosopra" è intrigante.

### » Longevità 85%

Diversi livelli tutti ben caratterizzati che ci porteranno via un po' di tempo di gioco. Piacevole.





**NEW GAME**

# TETRIS 1200

Anno: 2024

Editore/Sviluppatore: Gamtec,  
Super Fighter Team

Genere: Puzzle Game

Piattaforma: Amiga 1200 o  
superiore

Sito web: [https://](https://jotd666.itch.io/tetris1200)

jotd666.itch.io/tetris1200

Lo sviluppatore JOTD666 ci sta abituando bene! Direi benissimo. Da un po' di tempo ci regala conversioni di titoli complessi del passato.

Arcade come Xevious o Super Bagman e non solo. Tutti titoli riportati perfettamente sui nostri Amiga.

Anche nel caso di Tetris siamo di fronte al porting 1:1 del titolo di Aleksej Leonidovic Pazitnov del 1984. Anzi in realtà ci troviamo di fronte alla versione più nota per noi appassionati, quella di Atari che trovavamo in sala giochi nel 1988.

Una versione nata durante una mega diatriba tra Atari e Nintendo.

Mentre quella Nintendo era più simile alla versione realizzata da Pazitnov su Electronica-60, quella Atari portava alcuni cambiamenti.

Il gioco si basa su turni dove il giocatore avanza cancellando il maggior numero di linee target per quel determinato livello. Al termine del round, il campo viene svuotato e inizia il livello successivo con maggiori difficoltà.

Accanto al punteggio tradizionale del giocatore c'è una barra chiamata misuratore arcobaleno. Ogni quattro linee cancellate aggiunge 1 misuratore alla barra.

Il punteggio di ogni tetramino piazzato dal giocatore è  $d * \min(r*(r+h), 250)$  dove:

$d = 1$  per gravità normale o  $2$  per caduta morbida quando il tetramino atterra

$r =$  il numero di barre sul misuratore arcobaleno più  $1$

$h =$  riga su cui è stato posizionato il tetramino, meno  $1$ .



Dopo aver terminato ogni round il gioco assegna un "Bonus" fino a 2100 punti in base al numero di righe vuote sopra il blocco più in alto. La formula è  $5^e * (e + 1)$ , dove  $e$  è il numero di righe vuote, che corrisponde a 10 punti per la prima riga, 20 punti per la seconda riga, 30 punti per la terza riga e così via, con 10 punti in più per ogni riga.

Curiosità sul titolo Atari... non supportava gli sprite. L'intero gioco





## GIUDIZIO FINALE



## » Giocabilità 95%

Non credo che ci sia bisogno di dire molto di più. Perfetto nelle dinamiche e nel gameplay.

## » Longevità 99%

Infinita. Letteralmente infinita.

funzionava in modalità "testo", proprio come facevano le prime versioni rilasciate su computer.

Questa versione Amiga è un port del codice 6502 a 68000 e ha richiesto diverso tempo e numerose beta per essere perfetta.

Si perché ci troviamo di fronte alla migliore versione per questo gioco e la migliore trasposizione di Tetris su Amiga.

Richiede un processore 68020 e il chipset AGA, quindi come configurazione base è necessaria una Amiga 1200. Funziona perfettamente in emulazione ed è disponibile in formato ADF o WHDLoad.

Non mi resta che consigliarvi di scaricare il titolo e divertirvi.

Tetris appartiene a quei giochi letteralmente infiniti in termini di giocabilità.

Immortale!

di Carlo Nithaiah Del Mar Pirazzini





**NEW GAME**

# BREAKTHRU

**Anno:** 2024

**Editore/Sviluppatore:**

Acidbottle

**Genere:** Shoot em up

**Piattaforma:** Amiga

**Sito web:** [https://](https://acidbottle.itch.io/breakthru)

acidbottle.itch.io/breakthru

Un altro titolo storico dell'era arcade arriva su Amiga e lo fa grazie alle manine d'oro di Acidbottle, che tempo fa ci aveva deliziato con Wonderboy.

Il gioco è uno sparattutto a scorrimento orizzontale in cui il giocatore controlla un veicolo armato e ha il compito di recuperare un potente caccia da combattimento chiamato PK430 rubato da alcuni terroristi assetati di sangue.

Per portare a termine la missione si devono "attraversare" cinque diverse roccaforti nemiche ubicate in diversi scenari.

Le forze avversarie sono composte da soldati, carri armati, elicotteri, jeep e altro ancora.

Il gioco, conosciuto in Giappone come Forced Breakthrough, venne pubblicato nel 1986 dalla Data East e ottenne un discreto successo. Nello stesso anno, in licenza US Gold, furono rilasciate le versioni home computer per ZX Spectrum, Amstrad CPC e Commodore 64.

Questa versione Amiga ricalca perfettamente lo spirito, la struttura e le capacità tecniche della controparte da sala giochi.

C'è tutto il gioco, compreso l'utilizzo del doppio pulsante (un pulsante fa fuoco e il secondo permette di saltare con la moto).

Ci sono tutti i quadri disponibili, i nemici e i bonus.

Alla versione attuale (1.0) mancano ancora gli effetti sonori che verranno implementati, secondo Acidbottle, con le due prossime release, ma c'è comunque la colonna sonora ben realizzata.

È un titolo del passato. Un classico mangia gettoni dell'epoca dall'alto tasso di difficoltà. I livelli vanno memorizzati il più possibile e sono decisamente ostici fin dal primo.

Questo però non toglie il fascino del gioco e la voglia di cliccare sui "continue" presenti per vedere come il gioco andrà a finire.





## GIUDIZIO FINALE

### » Giocabilità 85%

Il titolo di Data East in tutto e per tutto, difficoltà compresa.

### » Longevità 75%

Gioco ostico che nasceva con l'obiettivo di "mangiare monetine". La versione Amiga è fedele alla linea.



Il gioco è stato rilasciato in formato ADF e richiede 1Mb di Chip Ram e 1Mb di altra Ram supplementare.

Sono previste una versione WHDLoad e una versione per ACD32.

Amiga sta rivivendo una seconda vita per quanto riguarda le conversioni di titoli da sala giochi e tutto questo non può che essere un bene.

di **Giampaolo Moraschi**





**NEW GAME**

# FINAL FIGHT ENHANCED

Final Fight ha quel “non so che” di mitologico. Un titolo che ci fece amare le sala giochi. Un gioco che ci faceva sentire eroi.

Liberare una fanciulla dalle mani della terribile organizzazione criminale Mad Gear era quanto di più “eroico” un bambino dei primi anni 90 potesse fare. Lo sviluppatore Yoshiaki Okamoto realizzò un gran bel prodotto nel 1989 per la Capcom.

Nato come possibile sequel del discutibile Street Fighter, il gioco prese una sua strada grazie al gameplay dinamico e alla storia e generando diversi seguiti e titoli simili (come Captain Commando o Cadillacs and Dinosaurs ad esempio). Come tanti giochi del tempo venne convertito per numerosi sistemi da casa. Chi non può non ricordarsi la terrificante conversione di US Gold per C64, massacrata da Zzap! Con un sonoro 22%? Oppure la legnosa versione per Amiga, che da ferma sembrava perfetta ed invece in movimento era il delirio?

Purtroppo gli home computer non vantavano lo sviluppo diretto come le console (Capcom curò personalmente lo sviluppo sulle piattaforme Nintendo e Sega con due discrete conversioni), ma si affidavano alle mani (non troppo esperte) dei team di US Gold.

Sono passati anni dalla stramba versione Amiga e siamo qui, davanti ai nostri 16 bit Commodore per parlare

ancora una volta della storia di Cody, Guy e Haggar.

Questa versione Enhanced cosa porta di nuovo a quello già visto?

Intanto porta risorse diciamo strane. Per prima cosa gira su Amiga ECS e non richiede chipset superiori o RTG. Buona notizia per tutti i possessori di macchine base vero? Purtroppo no. Per funzionare correttamente vuole 2Mb di Chip Ram e ulteriori 512k di altra Ram (veloce o lenta). Un po' tanto per sistemi base ma perfetti su sistemi avanzanti.

Ovvio che tutto questo su WinUAE non ha alcun problema, ma già su un A1200 il prodotto ha discrete difficoltà. Tra le novità c'è la possibilità di selezionare MAKI, personaggio femminile di Final Fight 2. Personaggio interessante da giocare per velocità e varietà di colpi.

Anno: 2024

Editore/Sviluppatore: Brick Nash/Prototron

Genere: Beat em Up

Piattaforma: Amiga

Sito web: [https://](https://drive.google.com/drive/folders/1GydNnHIY93p00ePcHNwD7xtn-zmtPPE0)

[drive.google.com/drive/folders/1GydNnHIY93p00ePcHNwD7xtn-zmtPPE0](https://drive.google.com/drive/folders/1GydNnHIY93p00ePcHNwD7xtn-zmtPPE0)







## GIUDIZIO FINALE

### » Giocabilità 80%

Il titolo ha sempre un certo fascino e la possibilità di giocare con Maki con l'aggiunta del secondo tasto sono una cosa davvero buona e giusta.

Peccato per la configurazione "anomala" per poter essere giocato su real hardware.

### » Longevità 85%

Un signor gioco ancora oggi nonostante siano passati anni e di beat em up ne siano usciti centinaia di migliaia. Sempre bello da giocare.

Abbiamo una maggiore velocità di reazione nel gioco rispetto alla "legnosità" del titolo originale.

Il titolo è stato sviluppato in assembly 68000 da parte dell'autore con lo scopo di apprendere questo linguaggio.

Devo ammettere che è un buon lavoro di elaborazione e tutto sommato riuscito bene. Non abbiamo incertezze nella fluidità dei livelli e, nonostante i piccoli bug presenti qua e là, tutto fila liscio e in maniera godibile.

Bene l'implementazione dei due pulsanti di gioco che rendono l'esperienza sicuramente più simile all'originale da sala giochi.

Personalmente avrei preferito una versione ottimizzata per le piattaforme AGA o superiori. Con delle richieste così alte in termine di memoria, meglio passare ad una tavolozza più "importante" e simile alla versione arcade. Ma si tratta di semplici gusti personali.

L'originale di US Gold era un prodotto triste, questo è decisamente meglio.

Il titolo è disponibile in due formati: il primo in adf su due file disco e il secondo in formato WHDLoad.

Il progetto è sempre in via di "rinnovamento" a detta dell'autore. Parla di future correzioni e questo non può che far piacere.

Tirando le proverbiali somme possiamo dire che questa versione di Final Fight è promossa nonostante alcuni difetti e che possiamo dimenticarci l'orrore di US GOLD.

di **Giampaolo Moraschi**



Oh Dad!  
I was so scared...

I'm so glad to see they didn't hurt you.  
I'm so sorry Jessica. I thought I'd lost  
you like I lost your mother.  
I'll never let anything bad happen to  
you again.

I love you Dad!





NEW GAME

# DOTTIE FLOWERS

Anno: 2024

Editore/Sviluppatore: Goldlocke

Genere: Platform

Piattaforma: Super Nintendo

Sito web: [https://](https://goldlocke.itch.io/dottie-flowers)[goldlocke.itch.io/dottie-flowers](https://goldlocke.itch.io/dottie-flowers)

La streghetta Dottie torna a fare capolino sui nostri Super Nintendo dopo le avventure in Dottie dreads nought.

Il nuovo platform di Goldlocke è bello, colorato e bizzarro.

Bizzarro nel modo di distribuzione. Per poter giocare in modo completo bisogna terminare la demo presente sul sito web e leggere attentamente le istruzioni.

La giovane streghetta Dottie si trova coinvolta in un impiccio di grave entità; l'invasione del pianeta terra da parte di una civiltà aliena robotica.

Dopo una simpatica intro animata ci si imbarca nell'avventura attraverso sei fasi di gioco.

La prima fase è una sorta di tutorial che ci mostra le capacità della piccola Dottie. Ci verrà spiegato come saltare, come utilizzare la scopa magica, il super spin e molto altro.

La struttura del gioco è quella di tutti i classici platform di sempre. Una bella mappa e i livelli da affrontare in puro stile Super Mario World.

Come dicevo è super colorato, animato

bene e con una grafica che ricorda il capitolo precedente e lo stile grafico "kawai" del team Goldlocke.

Mi è piaciuta molto la scelta cromatica e la colonna sonora.

Come specificato nel sito non è un progetto commerciale e la versione fisica è disponibile solo terminando il demo.

Un titolo semplice ma godibile che si porta a termine in circa 30/60 minuti.

Avrei voluto vedere sicuramente più azione, ma considerando l'esperimento degli sviluppatori va bene così.

di **Ingrid Poggiali**

## GIUDIZIO FINALE

### » Giocabilità 80%

Un simpatico platform game divertente da scoprire e giocare.

### » Longevità 70%

Si termina troppo velocemente.





**NEW GAME**

# TURBO OUTRUN

Anno: 2024

Editore/Sviluppatore: TCFS

Genere: Driving

Piattaforma: Plus/4



Turbo Outrun è un classico dei bei tempi andati delle sale giochi.

Tutti lo volevano giocare a casa sui sistemi a 8 e 16 bit del tempo. Lo volevano come in sala giochi correndo su piste lunghissime a tutta velocità. Nel 1989, stesso anno dell'uscita da sala, venne convertito su diverse macchine, edito dalla US Gold in Europa e sviluppato da Probe software per C64 e Ice Software per Amiga, ST, CPC e ZX.

La versione C64 ricevette una buonissima accoglienza con recensioni positive su tutte le riviste del tempo. Ben fatta, veloce e con una buona somiglianza (in termine di gameplay) al gioco originale.

Inutile dire che la versione di Ice Software per Amiga/ST e per CPC/ZX non fu altrettanto fortunata.

Dal 1989 ci spostiamo ad inizio 2024 e lo facciamo su una piattaforma poco compresa quando fu lanciata sul mercato ma che sta vivendo una "nuova giovinezza": il Plus/4.

Lo sviluppatore TCFS, partendo dall'ottima versione C64, ci ha tirato

fuori dal cilindro una bellissima versione per "l'altro" 8bit di casa Commodore.

E che versione!

Bella da vedere e da giocare.

Veloce il giusto e in tutto e per tutto alla pari con la versione uscita sotto etichetta US GOLD per il C64.

Il gioco è sempre quello: si corre con una visuale in prospettiva alle spalle della macchina, come nel classico Outrun, con la possibilità di selezionare tra cambio automatico o manuale.

Al posto della Ferrari Testarossa abbiamo una più veloce F40 in versione turbo con tanto di turbo attivabile, ci sono le piste variabili, la polizia... insomma c'è tutto Turbo Outrun.

La grafica funziona bene e il TED non fa rimpiangere il SID.

Com'è era la storia del Plus/4 macchina poco potente o poco utilizzata?

Godiamoci questo titolo

di **Giampaolo Moraschi**

## GIUDIZIO FINALE



### » Giocabilità 90%

Si guida bene e lo si fa con una buona conversione.

### » Longevità 90%

Questo è un titolo che ricaricherete ogni volta che ne avrete occasione, fidatevi.





**NEW GAME**

# PAC-MAN ARCADE

Anno: 2024  
 Editore/sviluppatore: Marco Leal ZX  
 Genere: Maze  
 Piattaforma: ZX Spectrum 48k  
 Sito web: <https://marco-leal-zx.itch.io/zx-pacman-arcade>

Iwatani nel 1980 creò un'icona assoluta, il tondeggiante Pac-Man. Nacque per diversificarsi dalla marea di cloni di Space Invaders e Arkanoid del periodo e nacque per gioco.



Le sue innumerevoli evoluzioni furono memorabili, così come le migliaia di conversioni più o meno dirette per tutti i sistemi di gioco dal 1980 ad oggi. Nel 2024 siamo qui ancora una volta a parlare di Pac-Man e lo facciamo su ZX Spectrum attraverso una versione PERFETTA. Perfetta nella forma, nelle logiche di gioco e negli aspetti visivi.



Marco Leal, lo sviluppatore, ha realizzato un prodotto sopraffino nei pochissimi KB a sua disposizione. Ci sono tutti i quadri di gioco, le logiche di assalto dei fantasmi, i bonus e le pillole.



Gira fluido questo PAC-MAN Arcade, con i suoi 50 fps su un semplice ZX Spectrum 48k. Che macchina meravigliosa vero?

Pensate, Marco ha inserito la schermata di Game Over dopo il livello 256 cosa che in arcade causava il crash del sistema. Che perfezionista! Supporto dei due giocatori e scaricabile dal sito in formato tzx compatibile in emulazione e su macchine virtuali.



Ma ci si può ancora divertire con Pac Man nel 2024? Sì. Decisamente sì, ci si diverte come nel 1980 e su Spectrum è ancora più bello.

di Carlo Nithaiah Del Mar Pirazzini

## GIUDIZIO FINALE

» **Giocabilità 90%**  
 È perfetto in tutto e rende benissimo esteticamente

» **Longevità 90%**  
 Richiede manualità come nell'originale e tanta pratica, ma non stanca.





NEW GAME

# THE MANDARIN II

Anno: 2024

Editore/Sviluppatore: Mananuk

Genere: Action/Platform

Piattaforma: Amstrad CPC/ZX Spectrum

Sito web: <https://mananuk.itch.io/the-mandarin-2-amstrad>

Prendete The Mandalorian direttamente dalla saga di Star Wars... cambiategli nome e aggiungete un pizzico di ironia nei testi ed eccovi... THE MANDARIN.

Un simpatico platform d'azione per Amstrad CPC e ZX Spectrum.

Il primo titolo seguiva gli eventi più o meno della prima serie questo secondo titolo prende un po' della seconda serie e la mescola ad una ricerca dei piani rubati dal terribile generale imperiale Limoncello (ebbe sì, abbiamo il nome del secolo...).

Il gioco è composto da 5 diversi livelli/fasi. Avremo una fase simil shoot em up, una fase platform e alcune fasi puzzle.

Tutte sviluppate bene per gameplay e design dei livelli.

Personalmente ho amato la fase di

fuga del quarto livello per velocità e tensione di gioco.

La versione Amstrad CPC è ben realizzata. Bella e colorata la grafica e con un buon sonoro basico fatto di effetti sonori semplici ma efficaci.

C'è un buon uso del colore e tutto ciò che vediamo su schermo è comprensibile e nitido.

La versione ZX soffre di alcuni piccoli glitch grafici ma dal punto di vista della giocabilità siamo sempre sullo stesso livello di quella Amstrad.

Buon titolo, semplice e divertente che forse finisce troppo presto per lo standard medio.

di **Giampaolo Moraschi**

## GIUDIZIO FINALE



### » Giocabilità 85%

Mi piacciono i giochi multi evento e la possibilità di affrontarli con diverse logiche.

Sistema di controllo ben fatto in entrambe le versioni

### » Longevità 65%

Finisce tutto troppo presto. Peccato.





NEW GAME

# SERGIO KIDD

Anno: 2023

Editore/sviluppatore:

PCNONOGames

Genere: Platform

Piattaforma: Game Boy Color

Sito web: [https://](https://pcnonogames.itch.io/sergio-kidd-for-gameboycolor)[pcnonogames.itch.io/sergio-kidd-for-gameboycolor](https://pcnonogames.itch.io/sergio-kidd-for-gameboycolor)

Un re cattivo che impone nuove tasse... cittadini che si ribellano... un mago che li trasforma in mostri... Piante, funghi velenosi, monete da raccogliere... Ma cos'è Super Mario? No è Sergio Kidd, il nuovo platform game di PCNONOGames dal nome decisamente bislacco ma sicuramente dal buon gameplay.

Il titolo è disponibile in versione digitale a 3,95 euro oppure in versione fisica (ordinabile sul sito).



Di che parla... anzi di che si tratta quando inseriamo Sergio Kidd nel nostro GBC.

Allora è un platform dei più classici diviso in 5 mondi diversi con più di 25 livelli da esplorare.

Come in tutti i platform del suo genere che si rispettino ci troveremo di fronte a diversi mostri piuttosto arrabbiati, trappole, piante carnivore, oggetti come scale e monete da raccogliere. Si lascia giocare con piacere e i mondi sono simpatici e ben strutturati.



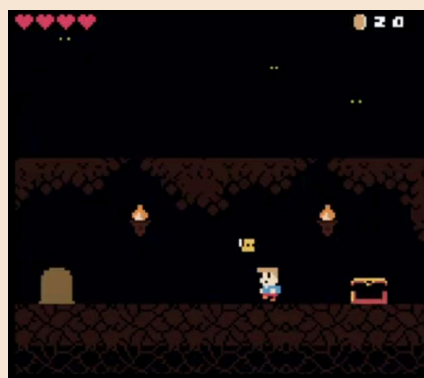
Senza dubbio l'influenza dei giochi di Mario è evidente. Molti mostri e tantissimo design ricordano alcuni capitoli della saga dell'idraulico Nintendo, ma non credo sia un male. C'è una bella grafica colorata e molta varietà di animazioni, c'è un buon sonoro ma forse il piccolo Sergio non ha il carisma adatto per questo gioco... Non saprei... Mi è sembrato fiacco, anonimo.

I comandi rispondono bene e il gioco si lascia giocare con gusto.

Non è un titolo difficilissima da portare a termine, soprattutto se si è dei "pro" di Mario Bros e affini, ma per 3 euro e poco più ci può stare benissimo.

Sono contenta di portarmelo all'università... in treno è il gioco perfetto per passare il tempo di viaggio.

di **Ingrid Poggiali**



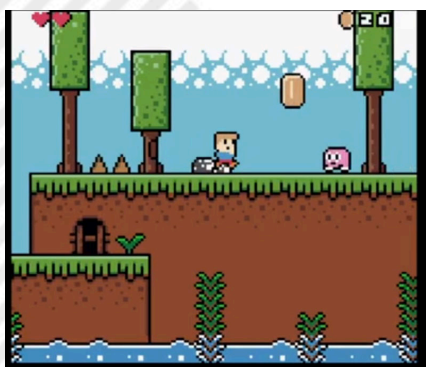
## GIUDIZIO FINALE

### » Giocabilità 80%

Buoni comandi e struttura solida da platform ben rodato.

### » Longevità 70%

Non troppo impegnativo se si è padroni di questo tipo di giochi.





**NEW GAME**

# DUCK HUNT

Anno: 2024

Editore/Sviluppatore: Marggines

Genere: Shoot em up

Piattaforma: Amiga

Sito web: <https://www.ppa.pl/forum/strefa-gier/46460/duck-hunt-kolejna-nowa-gra#m807596>



Dite la verità? Quando giocavate a Duck Hunt su NES, non provavate un senso di odio assoluto verso il cane da caccia? Con le sue risatine malefiche quando non si abbattevano le anatre? Malefico quadrupede...

Duck Hunt è stato uno dei titoli must per la console a 8 bit Nintendo assieme a Super Mario. Capace di far vendere una marea di Action Set e di pistole Zapper.

Semplice, divertente e ben congeniato. Assolutamente irritante quando si veniva "percolati" dal braccio al nostro seguito.

Questa inaspettata versione Amiga arriva direttamente dalla Polonia. Il suo sviluppatore ci fornisce due file (adf e lha) dove troveremo una colorata trasposizione del gioco NES.

La grafica non è opera sua ma, ci tiene a precisarlo, presa da questo sito: [https://www.sprites-resource.com/custom\\_edited/duckhuntcustoms/sheet/63915/](https://www.sprites-resource.com/custom_edited/duckhuntcustoms/sheet/63915/).

Non ha rilasciato specifiche tecniche. No il test lo abbiamo fatto su una A600 con 2 mb di Ram e in emulazione: tutto filato liscio come l'olio.

Il gioco c'è tutto e si gioca col mouse: ci sono le due modalità (singola anatra e doppia anatra), c'è il cane, c'è la musichetta.

Alla lunga stanca ma ha due grandi pregi: non richiede chissà che sforzo mentale e soprattutto è gratis... almeno fino a quando MAMMA NINTENDO non lo farà cancellare dalla rete. Divertente.

di Carlo Nithaiah Del Mar Pirazzini



## GIUDIZIO FINALE



### » Giocabilità 90%

Sparare ai pennuti è ancora divertente... anche su Amiga!

### » Longevità 85%

Un titolo simpatico da caricare ogni tanto per una veloce partita..





# KEYSTONE KAPERS

Oggi rivivremo la nostra infanzia giocando a guardie e ladri anche se solo virtualmente. Ammetto che non mi dispiacerebbe rigiocare dal vivo aggiungendo questo ingrediente ai nostri nostalgici svaghi fatti di console e home computer retro.

Il gioco in questione è Keystone Kapers, pubblicato da Activision nel 1983 per l'Atari 2006 e l'anno successivo per altre console Atari e non solo. In quegli anni è mancata la conversione per il biscottone, è vero. Ma nel 2019 eccola! Grazie ad un volenteroso e bravissimo programmatore, appassionato e nostalgico come noi, siamo riusciti ad ottenerla a distanza di oltre trenta anni! Le sorprese come vedete non finiscono mai.

Nel gioco impersoneremo un buffo poliziotto, di nome Keystone Kelly, all'interno di un centro commerciale stile anni 80 e saremo all'inseguimento di un detenuto evaso, con la classica divisa a righe, di nome Harry Hooligan. Harry scorazza per il centro commerciale cercando vie di fuga e facendoci macinare chilometri sui vari piani e scale mobili. Tutto ciò che dovrà fare il poliziotto sarà prenderlo e non sarà affatto facile su ben tre piani pieni di ostacoli come aeroplanini, palle rimbalzanti e carrelli della spesa che sfrecciano a tuta velocità. Oltre ad essi ci sarà anche un tempo limite (tipico dei giochi arcade dell'epoca) a complicare le cose, dato che ad ogni collisione con gli ostacoli (tranne l'aeroplanino che vi farà perdere una vita) verranno sottratti alcuni preziosi secondi e una volta arrivato a zero la partita terminerà direttamente.

Sui piani ci sono anche oggetti bonus da raccogliere, indispensabili per

arrivare ai 10.000 punti per guadagnare una vita extra. Per fortuna dalla nostra parte ci sono gli ascensori per spostarci più velocemente sui piani, ammesso che non si trovino bloccati come in un condominio di 300 famiglie. Ah ho dimenticato di dirvi che perderemo una vita anche se faremo scappare l'evaso attraverso il tetto.

Il gioco non ha un finale e solo una volta perse tutte le vite potremo dire di averlo "terminato". Questo gioco lo trovo abbastanza longevo per il semplice fatto che la voglia di prenderlo in mano per una partitina ogni tanto c'è sempre, che sia durante una pausa lavorativa o una serata in cui non c'è niente di bello in tv. La musica di accompagnamento è molto simpatica.

Rivedendo la primissima versione per Atari 2006 l'ho trovata molto carina e giocabile e devo dire che la conversione sul biscottone è riuscitissima, non parlo solo di elementi grafici. Per la console Atari uscì anche un secondo capitolo, non ancora presente nella versione home computer e spero di vederla presto realizzata...

Questa conversione mi ha un po' viziato, come le ciliegie: se non avete avuto ancora modo di giocarla, fatelo al più presto perché la firma Activision è più di una garanzia e in quegli anni ci ha fatto molto divertire e sognare con i suoi titoloni. Per cominciare in bellezza il 2024 direi che è proprio l'ideale, fino a quando non arriveranno le giornate calde che ci faranno uscire di casa ancora una volta e con le giornate che iniziano già ora ad allungarsi.

di **Daniele Brahimi**

**Anno:** 1983

**Editore:** Activision

**Genere:** Platform

**Piattaforma:** Commodore 64/  
Atari 2600



## GIUDIZIO FINALE

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

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





## La calma è la virtù dei forti?

Come probabilmente avrete notato, questo numero di RMW si è fatto attendere un po' più del previsto. Tranquilli, non siamo a secco di contenuti e nemmeno a corto di idee. Abbiamo soltanto voluto prenderci un po' di tempo per adempiere ad alcune operazioni che stavamo rimandando da troppo tempo e per passare del tempo prezioso con le nostre famiglie.

In questi mesi di assenza siamo tornati in pari con l'impaginazione dei numeri in inglese. E allora perché non avete ancora pubblicato quei numeri? Semplicemente per dare il tempo ai lettori internazionali di leggere una rivista alla volta e lasciare il tempo tecnico di vita per ciascun numero. Se avessimo pubblicato tre numeri contemporaneamente avremmo rischiato di inflazionare la piazza perdendo così di appeal. E noi questo non lo vogliamo!

Tornando a questo numero 46, devo ammettere che io stesso sono rimasto stupito dalla quantità e qualità dei giochi presenti. Basta dare un'occhiata al box dei voti per capire immediatamente come il livello qualitativo delle produzioni ad 8 e 16 bit di questo periodo sia incredibilmente alto. "Se è un sogno non svegliatemi", direbbe qualcuno, ma per fortuna è la pura e semplice realtà dei fatti. Programmatori, grafici e musicisti talentuosi hanno riscoperto il piacere di scrivere codice, disegnare e comporre musica per macchine di 30/40 anni fa.

Ed a noi sta bene così, dato che spesso possiamo godere del frutto delle loro fatiche gratuitamente o ad un costo irrisorio.

Prima di chiudere definitivamente questo numero, voglio puntare la vostra attenzione all'ultima pagina delle riviste.

Non voglio rovinarvi la sorpresa, ma, come anticipato nel numero 45, ci sono alcune novità divertenti che stiamo preparando per i prossimi numeri e... non ne rimarrete delusi, ne sono sicuro.

Ovviamente ci prenderemo tutto il tempo necessario per fare le cose nel miglior modo possibile.

D'altronde, la calma è o non è la virtù dei forti?

*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 46 - FEBBRAIO 2024

**Direttore Responsabile**

*Francesco Fiorentini*

**Vice Direttore**

*Marco Pistorio*

**Coordinatore Redazione/Editing**

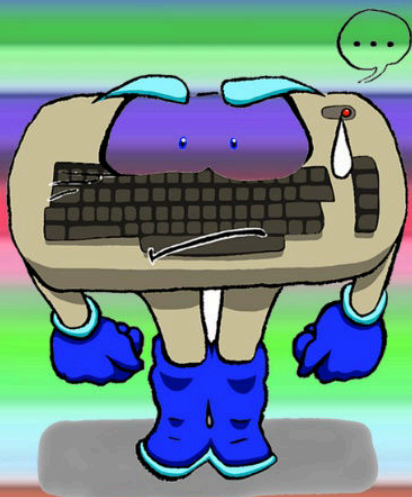
*David La Monaca*

**Responsabili Area Web**

*Giorgio Balestrieri*



# MR. JACK



**Sarà uno dei protagonisti dai prossimi numeri delle recensioni... assieme a Sir Clive, NISHI SAN e tanti altri.**

**Opera del nostro Fabio De Renzis**