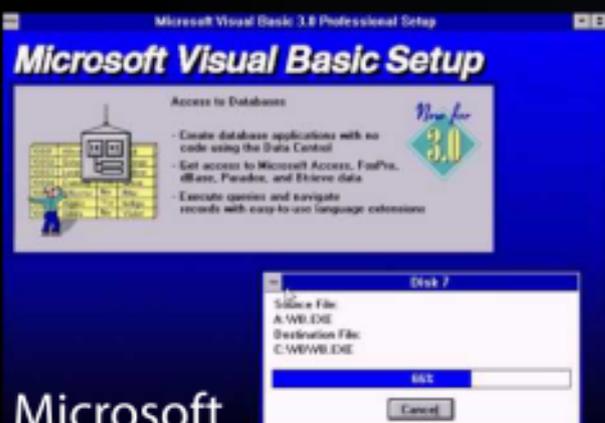




# RetroMagazine

semplicemente retro

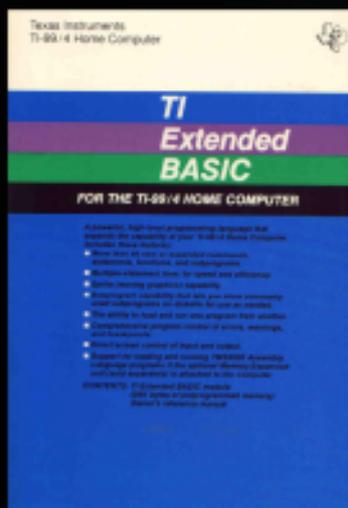
Numero 18 - Anno 3 - Ottobre 2019 - WWW.RETROMAGAZINE.NET - Pubblicazione gratuita



Microsoft  
VISUAL BASIC



Dracula the undead  
(Atari Lynx)



Dark Side of 16 bit Ti99:  
Il Basic esteso



GANBARE GOEMON 2  
KITERETSU SHŌGUN (SNES)



Retro Magazine a  
Once Upon a Sprite  
2019

- Introduzione ad HOLLYWOOD - 2ª parte
- Impara l'arte... Il formato D64 - 3ª parte
- Giappone 6ª puntata:  
Il Giapostroano caso di Obachan button
- Retroeventi



*Happy*

# HALLOWEEN

*from Retro Magazine*

## Aggiungi un posto a tavola che c'è una zucca in più!

Contrariamente a quanto molti credono, Halloween non è una festa americana, è in realtà una festività di origine irlandese che affonda le sue radici nella notte dei tempi. Il nome deriva da All Hallows' Eve Day, la vigilia di Ognissanti. Inizialmente celebrata nel mese di maggio, nell'834 a.C. fu spostata alla notte tra il 31 ottobre ed il 1 novembre per coincidere con la festa druidica del Samhain, durante la quale si celebrava l'inizio del nuovo anno...

L'anno scorso abbiamo dedicato un numero intero a questa ricorrenza, quest'anno invece abbiamo deciso di dedicarle soltanto la copertina e qualche articolo. L'idea di fondo era quella di non ripetersi per non diventare prevedibili, ma allo stesso tempo non potevamo esimerci dal trattare l'argomento che offre spunti decisamente interessanti per chi abbia voglia di coglierli.

Edoardo ha deciso di raccogliere la sfida, presentando un articolo interamente dedicato ai giochi da giocare durante la notte delle streghe! Provate a dare soltanto un'occhiata ai titoli nella lista da lui proposta... Ovviamente per molti dei titoli è facile intuire il perchè siano lì, ma almeno un paio di essi mi hanno sorpreso. Dopo aver letto la motivazione invece ho fatto il canonico due più due ed ho dato ragione all'autore. :-)

Anche Daniele e Starfox si sono lasciati sedurre dall'atmosfera lugubre del periodo ed hanno presentato la recensione di due giochi assolutamente in tema. Che poi entrambi riguardino i vampiri è soltanto un caso.

Li avete giocati? No? Dovete recuperarli!

Oltre agli articoli a tema trovano spazio in questo numero le consuete rubriche, alcuni nuovi argomenti che potrebbero avere un seguito nelle prossime pubblicazioni di RM ed un paio di interviste che avevamo in programma da tempo ma che ancora non eravamo riusciti a portare a termine.

Continuano inoltre le collaborazioni con gli amici del DumpClub 64 e di AmigaGuru e trovano spazio alcune nuove firme, Andrea Pastore e Carlo Nithaiah Del Mar Pirazzini, che ci auguriamo possono presto diventare collaboratori fissi di RM.

Concludo ringraziando tutti voi lettori per continuare a seguirci così numerosi e per stimolarci sempre a fare meglio. Buona lettura!

**Francesco Fiorentini**

## SOMMARIO

◇ Impara l'arte...Il formato D64 - terza parte	Pag. 3
◇ Microsoft Visual Basic	Pag. 6
◇ Dark Side of 16 bit Ti99 - Il Basic esteso	Pag. 9
◇ Introduzione ad HOLLYWOOD - seconda parte	Pag. 12
◇ Calcolare in multipla precisione - parte quarta	Pag. 16
◇ Il dumping da nastro	Pag. 23
◇ RetroMath: Memorie associative e reti di Hopfield	Pag. 25
◇ Intervista a Vincenzo Scarpa	Pag. 30
◇ M4 Interface per Amstrad CPC	Pag. 35
◇ Sedai Linux	Pag. 38
◇ Dracula The Undead (Atari, Lynx)	Pag. 42
◇ Ganbare Goemon (SNES)	Pag. 44
◇ Ganbare Goemon 2 (SNES)	Pag. 46
◇ Super Mario Bros (C64)	Pag. 48
◇ Speciale Retrohalloween	Pag. 50
◇ Nosferatu (Multi)	Pag. 53
◇ Rally Bike (Arcade)	Pag. 54
◇ Giappone 666 ^puntata: Il Giapoprostrano caso di Obachan button	Pag. 55
◇ Firenze Vintage Bit 2019 - Conferenza	Pag. 58
◇ Once Upon a Sprite 2019 - resoconto breve	Pag. 60

### Hanno collaborato alla stesura di questo numero di RetroMagazine

- Marco Pistorio
- Francesco Fiorentini
- Ermanno Betori
- Gianluca Girelli
- Alberto Apostolo
- The Dump Club 64 Team
- Giuseppe Fedele
- David La Monaca (Cercamon)
- Andrea Pastore
- Giorgio Balestrieri
- Fox Mulder
- Carlo Nithaiah Del Mar Pirazzini
- Edoardo Ullo
- Daniele Brahimi
- Querino Ialongo
- Michele Ugolini
- Leonardo Vettori
- Federico Gori
- Copertina a cura di Flavio Soldani





## Impara l'arte... Il formato D64 - terza parte

di Marco "The Ripper" Pistorio

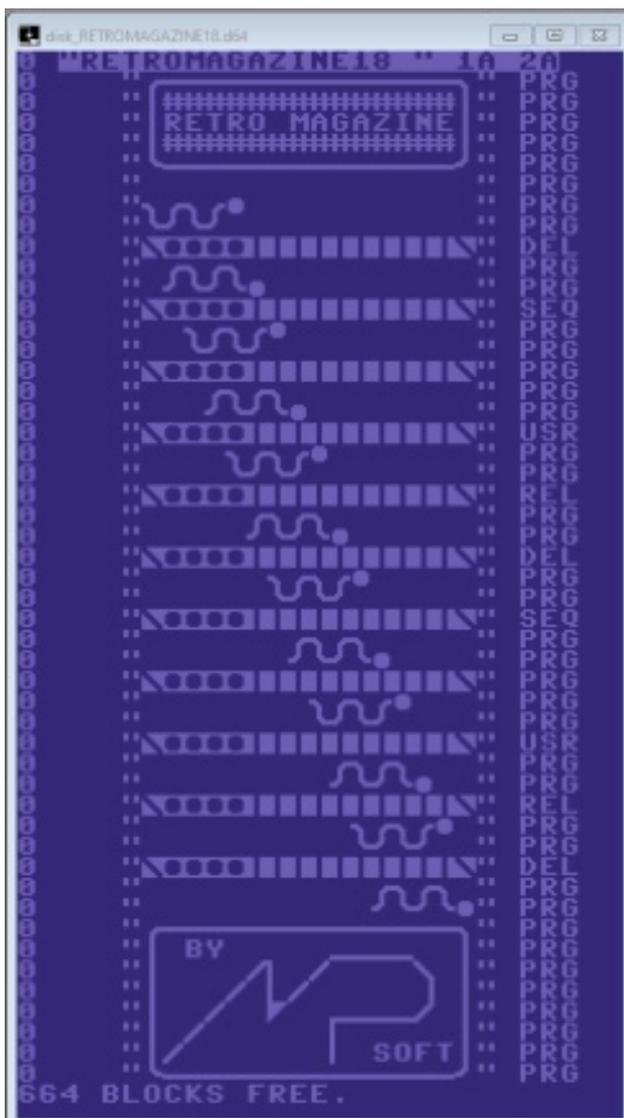
Bentrovati amici lettori, in questo nuovo numero di RetroMagazine, dove approfondiremo alcune peculiarità probabilmente poco note e/o poco sfruttate del formato con il quale vengono memorizzate le informazioni contenute nei preziosi floppy disk del nostro biscottone, ovvero il formato ".D64".

Invito chi non avesse letto i precedenti articoli sull'argomento, a firma di Francesco Fiorentini, a leggerli con attenzione in quanto in questo articolo considererò le nozioni precedentemente esposte come già acquisite e quindi non mi ci soffermerò troppo.

Sono certo che le informazioni contenute in questo articolo stuzzicheranno l'interesse dei lettori più curiosi e "smanettoni".

Desidero ringraziare **Holger Aurich** per il suo video su FB relativo all'argomento (vedi riquadro riferimenti WEB), video dal quale sono riuscito a trarre buona parte degli elementi che esporrò in questo articolo.

Adesso non mi resta che augurarvi buona lettura!



### GIUSTO UN PIZZICO DI TEORIA

Sappiamo già che la traccia 18 del floppy contiene la BAM (acronimo di Block Availability Map) e la directory.

La directory contiene, in particolare, il nome di ciascun file memorizzato sul disco ed il tipo di file cui appartiene (PRG, SEQ, REL,USR, DEL).

Saprete che è possibile utilizzare diversi caratteri per formare il nome di un file.

Forse non saprete invece che è possibile inserire, con un pò di attenzione, anche alcuni codici PETASCII "speciali", ad esempio quelli relativi al movimento del cursore (CRSR UP, CRSR DOWN, CRSR LEFT, CRSR RIGHT) all'interno dell'area destinata a contenere i nomi dei files su disco. A cosa potrebbero servire tali caratteri speciali?

Per creare delle vere e proprie animazioni mentre si lista sullo schermo il contenuto di una directory, giusto per fare un esempio...(!)

Accennavo però al fatto che è necessario prestare un pò di attenzione per fare ciò.

E' facile verificare in prima persona che, digitando un carattere " (doppi apici) e premendo successivamente il tasto cursore su, il cursore non si sposterà di una riga verso l'altro come ci si potrebbe aspettare. Verrà stampato invece a video un carattere speciale, una pallina piena in reverse-image, che è il carattere che corrisponde al codice PETASCII relativo a questo specifico tasto (\$91, ovvero 145 in notazione decimale).

Il nome di un file all'interno di una directory sarà sempre all'interno di una coppia di doppi apici. Di conseguenza occorre fare qualcosa per permettere l'uso di caratteri speciali come, appunto, i tasti cursore di cui vi accennavo. Qual'è la soluzione di questo rebus? Se digitate un carattere " (doppi apici) e premete insieme i tasti SHIFT+RETURN, il cursore si posizionerà alla estrema sinistra della riga immediatamente successiva a quella dove vi trovavate inizialmente. Da questo momento in poi, premendo i tasti per spostare il cursore nelle 4 possibili direzioni (ALTO, BASSO, SINISTRA e DESTRA), il cursore si sposterà come atteso, senza ottenere sfilze di caratteri in reverse-image. Ho preparato per questo articolo una semplice animazione. Listando il contenuto della directory del file .d64 relativo a questo articolo, vedrete una prima parte scorrere normalmente, quindi un "serpentello" muoversi da sinistra verso destra ed infine una parte finale scorrere anch'essa normalmente.

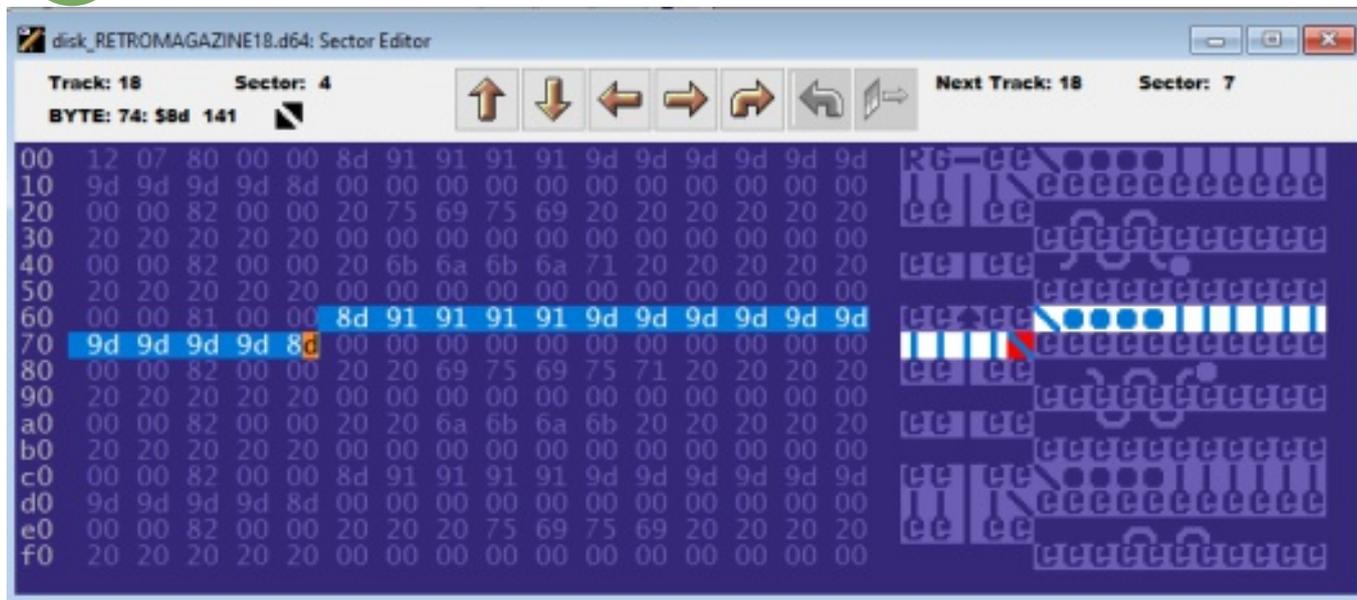
L'animazione del serpentello è creata sfruttando i caratteri grafici predefiniti del C64, ed ogni "frame" viene visualizzato su due righe di schermo.

Ciascun frame viene separato dal successivo da un file con un nome studiato appositamente.

**Il nome è costituito così: 1 carattere SHIFT+RETURN (\$8D), 4 caratteri CRSR UP (\$91), 10 caratteri CRSR LEFT (\$9D), 1 carattere SHIFT+RETURN (\$8D).**

Il primo carattere, SHIFT+RETURN (\$8D), serve per far





sì che i caratteri cursore successivi spostino il cursore e non compaiano invece come caratteri in reverse-image, come già spiegato.

I 4 caratteri CRSR-UP (\$91) servono per riposizionare il cursore 4 righe sopra quella dove si trova inizialmente. Considerate che 2 righe vengono utilizzate per stampare ciascun frame del "serpentello" in movimento, quindi subito dopo ci si troverebbe alla terza riga, ma a causa dello SHIFT+RETURN il cursore si troverà ancora giù di una riga, quindi dobbiamo recuperare esattamente 4 righe per far sì che i frames del serpentello compaiano sempre in corrispondenza della stessa zona dello schermo. I 10 caratteri CRSR LEFT (\$9D) ed il conclusivo SHIFT+RETURN (\$8D) servono per far posizionare il doppio apice alla fine del nome del file "costruito" il più a sinistra dello schermo possibile, insieme all'indicazione

del tipo file, informazione quest'ultima che viene sempre stampata in corrispondenza di ciascun nome di file presente nella directory.

Si potrebbero adoperare meno caratteri CRSR LEFT, ottenendo un diverso posizionamento del doppio apice e dell'indicazione del tipo di file.

Lo stesso risultato, ovvero un diverso posizionamento di questi due elementi, si otterrebbe omettendo il secondo SHIFT+RETURN alla fine dell'area del nome del file.

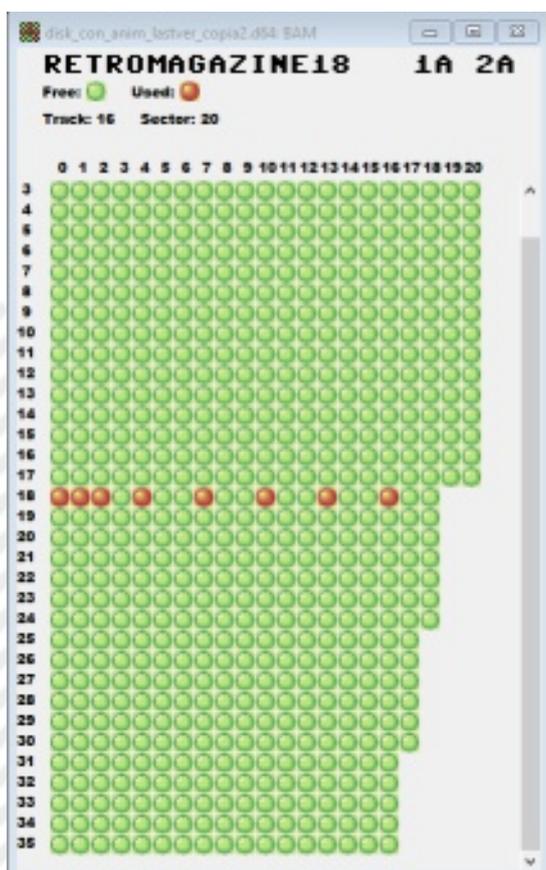
#### DALLA TEORIA ALLA PRATICA

Tutti gli elementi presenti nella directory contenuta nel file .d64 di esempio sono stati creati manualmente, utilizzando uno strumento molto famoso tra i retroappassionati del Commodore 64, un programma chiamato **DirMaster**, già presentato nei precedenti articoli relativi al formato .d64

I vari elementi stati inseriti come normali voci di directory dal menù "File" di DirMaster, scegliend poi "Insert" ed infine inserendo opportunamente i caratteri che compongono i vari frames del "serpentello".

Non è possibile però editare direttamente ciascuna riga di separazione dei vari frames, a causa dell'impiego di caratteri inconsueti all'interno dell'area preposta a contenere il nome del file. Per queste righe speciali conviene editarne il loro contenuto sfruttando la funzione "BAM EDITOR", che è presente al di sotto del menù "View" di DirMaster, e facendo click sul blocco della traccia 18 contenente la voce da editare. I dati relativi alla directory del disco si trovano, come già detto, all'interno della traccia 18.

Inoltre i blocchi già allocati perchè utilizzati vengono già evidenziati appena si richiama la funzione "BAM EDITOR". Il modo migliore per editare il contenuto delle righe di separazione tra un frame ed il successivo della animazione è quello di impostare i codici carattere che compongono tali righe direttamente in formato esadecimale. Ciascuna di queste righe di separazione apparirà quindi come una sequenza così composta: **8D 91 91 91 91 9D 9D 9D 9D 9D 9D 9D 9D 8D**.







## Microsoft Visual Basic

di Francesco Fiorentini

E' il 1993, sto frequentando il primo anno di Universita' a Siena insieme ad altri 3 amici, quando la mia attenzione viene catturata da una locandina. Si tratta di un bando per un corso di 'Programmatore GUI' di 2000 ore che si svolgera' a Firenze, completamente sovvenzionato dalla Regione Toscana. Non ricordo di preciso dove vidi la locandina, probabilmente nei locali dell'Universita', perche' anche Andrea, uno dei miei compagni di Universita', si accorse della cosa.

GUI significa Graphical User Interface, cioe' programmatore di interfacce grafiche. All'epoca la programmazione era il mio chiodo fisso e la possibilita' di realizzare delle interfacce grafiche nei programmi non era cosi' scontata come adesso. Il grosso degli applicativi e dei gestionali era ancora ancorato alle interfacce a carattere (Unix, Xenix, DOS, AS400...), quindi un simile corso risultava ai miei occhi come una finestra aperta sul futuro e, molto probabilmente, una reale possibilita' di impiego in una software house.

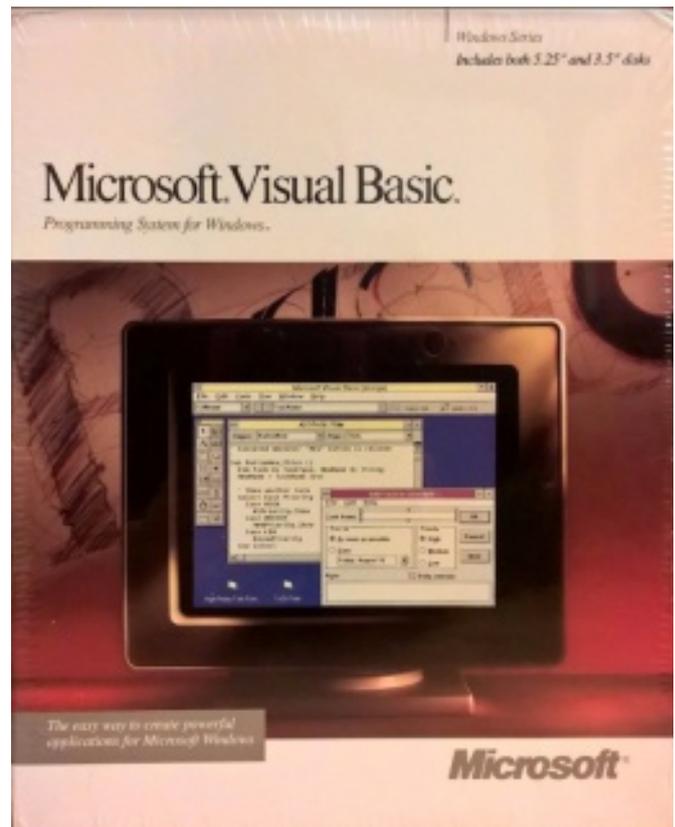
Non ci pensai due volte, mi iscrissi al corso e poco dopo scoprii che anche Andrea aveva fatto lo stesso. Ovviamente tra il dire ed il fare c'e' di mezzo il mare. In questo caso piu' che il mare, una marea... una marea di persone che aveva avuto la nostra stessa idea.

Il corso aveva un numero limitato di posti a disposizione, mi sembra di ricordare tra i 15 ed i 20, quando invece gli aspiranti superarono abbondantemente le 700 unita'. Un test attitudinale avrebbe quindi sancito l'ammissione al corso. Mi ricordo che per il test d'ammissione furono utilizzate le aule dell'Universita' di Firenze ed io ed Andrea, complice il fatto della vicinanza dei nostri cognomi, ci ritrovammo a compilare il test attitudinale nella stessa aula. Ve la faccio breve, venimmo scelti entrambi ed ancora oggi penso che il corso di Programmatore GUI dell'ELEA sia stata una delle cose piu' interessanti a cui abbia mai preso parte!

Come mai ho voluto raccontarvi tutta questa storia in un articolo dedicato al Visual Basic? Semplice, il Visual Basic e' stato uno degli argomenti trattati durante il corso ed io ne ho fatto la conoscenza proprio grazie al corso GUI. Qualcuno dei lettori potrebbe obiettare che Windows ed i suoi tool (quindi anche il Visual Basic) non facciano parte del retrocomputing vero e proprio, ma per come la vedo io, qualsiasi cosa legata all'informatica che abbia piu' di 25 anni puo' essere considerata retro. Un auto puo' essere considerata d'epoca a patto che sia stata costruita almeno 30 anni prima dell'iscrizione ad uno dei Registri dell'ASI ma prima della Legge di Stabilita' del precedente governo bastavano 20 anni, quindi ho deciso di dare nel mezzo...

Altra doverosa precisazione; con questo articolo mi limitero' a dare un po' di informazioni generali riguardo il Visual Basic ed il VB 3, ma se ce ne fosse l'interesse si potrebbe anche fornire informazioni piu' dettagliate e perche' no, anche un vero e proprio corso su come uti-

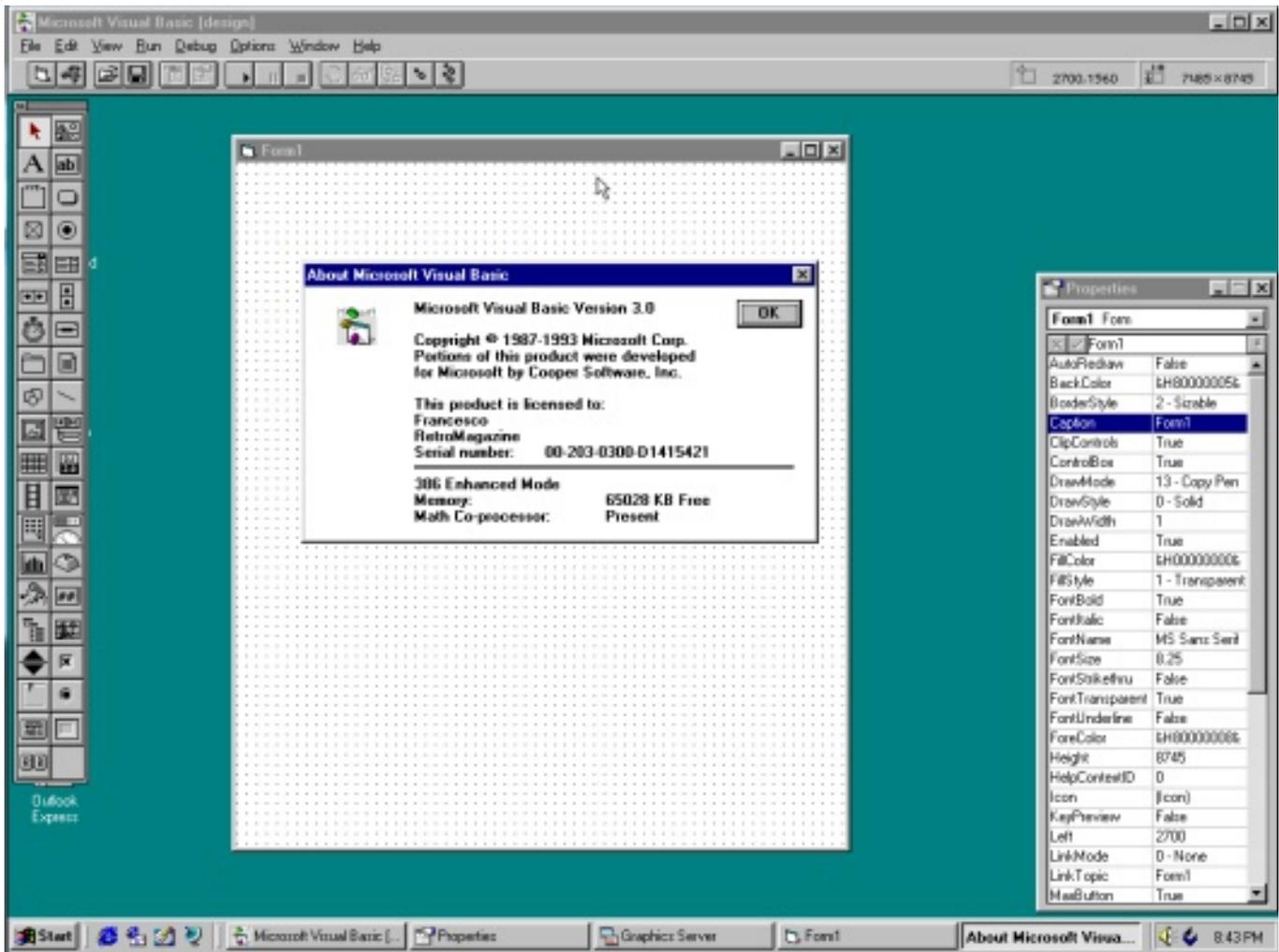
lizzare questo strumento. Come abbiamo visto nel numero 14 nell'articolo sulla struttura dati del D64, il VB puo' essere velocemente utilizzato per creare tool o accedere ai dati memorizzati sui supporti virtuali dei nostri amati 8 bit (e non solo...), quindi perche' non introdurre l'argomento in RM?



### Un po' di storia

La prima versione di Visual Basic e' la 1.0 (ma va? :-)) del 1991. Il tutto nasce da un prototipo di shell drag and drop chiamata Tripod realizzato da Alan Cooper. Inizialmente la shell ha soltanto un semplice linguaggio di scripting, niente a che vedere con il Basic, ma il miracolo avviene quando Cooper decide di mostrare il suo prototipo a Bill Gates nel 1988. Bill Gates ne intuisce subito le potenzialita' e, una volta acquisiti i diritti come MS, decide insieme ad Alan di creare un ambiente piu' evoluto. Fu cosi' che dal matrimonio di Quick Basic e Tripod, nel 1991 nasce il Visual Basic: un linguaggio di programmazione di terza generazione orientato agli eventi e con un ambiente di sviluppo rapido per applicazioni Windows. Prima pero' di arrivare ad una versione matura del prodotto devono passare un paio d'anni. Le versioni 1.0 e 2.0 infatti sono piuttosto acerbe e vengono velocemente sostituite fino ad arrivare nel 1993 alla versione 3.0. Visual Basic 3.0 sara' la prima release ad essere distribuita in due versioni, Standard e Professional ed ad includere un motore di Database in grado di leggere e scrivere file





Access.

Visual Basic 3.0, che permette di creare applicazioni a 16 bit per Windows, resiste per 2 anni. Nel 1995 viene sostituita dalla versione 4.0 che permette di creare applicazioni a 16 o 32 bit. Ricordo la versione 4.0 con orrore per una serie di problemi che ne minavano fortemente la stabilita' e le prestazioni.

Nel 1997 viene quindi distribuita la versione 5.0. Si tratta di una versione stabile e che raggiungera' una grande popolarita'. La vera rivoluzione della versione 5 e' la possibilita' di compilare il codice in eseguibile Windows nativo rispetto agli eseguibili in P-code creati dalle versioni precedenti. Nonostante la stabilita' raggiunta, la versione 5 viene velocemente soppiantata nel 1998 dalla versione 6.0 che sancira' il canto del cigno del Visual Basic. Visual Basic 6.0 e' l'ultima versione 'classica' prima che Microsoft passi alla versione .NET. La versione .NET e' infatti una completa rivisitazione del linguaggio VB per renderlo compatibile con il framework .NET ma che ne sancisce di fatto la mancata compatibilita' con le versioni precedenti.

Visual Basic 6.0 ha una vita piuttosto lunga per un prodotto informatico, ben 10 anni. La sua vita commerciale va dal 1998 al 2008, quando viene di fatto abbandonato come prodotto ufficialmente supportato da Microsoft. La popolarita' del linguaggio pero' e' tale che ancora oggi il team di sviluppo di VB6 e' impegnato a garantirne il funzionamento anche nell'ultimo sistema operativo della casa di Redmond: Windows 10.

### Solo per Windows?

Visual Basic non e' stato pero' soltanto un prodotto Windows. Nel 1992 la Microsoft immette sul mercato il Visual Basic 1.0 per DOS. Si tratta piu' che altro di un esperimento, che verra' abbandonato velocemente. Il linguaggio non era nemmeno compatibile con il VB per Windows e l'interfaccia grafica veniva simulata con caratteri ASCII. Pur trattandosi di un prodotto pienamente funzionante e' misconosciuto ai piu' e soltanto pochi lo hanno utilizzato. Essendo un tool di nicchia, ci sembrava doveroso riservargli un po' di spazio con un articolo dedicato che verra' pubblicato nei prossimi numeri a cura di Giorgio Balestrieri.

### Visual Basic 3.0

Il Visual Basic 3.0, VB3 per gli amici, e' la prima vera versione matura di questo ambiente di sviluppo; introduce infatti dei significativi miglioramenti alle precedenti due versioni e di fatto comincia a farsi apprezzare come strumento per la realizzazione di applicazioni professionali. Vediamo un po' di elencare i punti di forza del VB3 in ordine prettamente sparso.

*IDE o Integrated Design Environment:* in italiano Ambiente di Sviluppo Integrato, e' la possibilita' di avere all'interno dello stesso ambiente di progettazione una serie di strumenti che aiutano il programmatore nella progettazione del software. Generalmente un IDE si compone dei seguenti strumenti: un editor di codice; un compilatore e/





o un interprete; un tool di building automatico; un debugger. L'ambiente IDE del Visual Basic, come dice il nome stesso, si definisce Visual IDE in quanto tutti gli strumenti sono accessibili tramite l'interfaccia grafica.

*RAD o Rapid Application Development:* anche se i puristi del RAD potrebbero storcere la bocca nell'associare questa definizione al Visual Basic, e' indubbio che lo sviluppo di applicazioni con il Visual Basic sia rapido, soprattutto grazie alla comoda interfaccia grafica ed all'ambiente IDE di cui sopra.

*Facilita' nel disegno delle interfacce grafiche o GUI:* e' innegabile che l'ambiente VB3 abbia aperto praticamente a tutti la possibilita' di progettare e disegnare facilmente la propria applicazione grafica in Windows, grazie soprattutto all'ambiente grafico ed ai controlli VBX e OCX integrati. Fino ad allora chi avesse voluto cimentarsi nello sviluppo di applicazioni Windows doveva barcamenarsi tra la complessita' delle SDK di Windows ed il codice in C. Dopo il VB3 chiunque sarebbe stato in grado di piazzare dei controlli nel modulo del programma e, con pochissime righe di codice, creare una semplice applicazione.

*Facilita' nella scrittura del codice:* il Basic e' universalmente riconosciuto come un linguaggio semplice, di facile accesso e comprensione. L'idea della Microsoft di dotare la sua suite Visual IDE con questo linguaggio, oltre al piu' complesso Visual C++, ha di fatto aperto la strada a migliaia di coder e di improvvisati programmatori che avrebbero trovato ostico il meno familiare linguaggio C. Secondo chi scrive, la fortuna del VB e' stata proprio la scelta del linguaggio. Finalmente una cosa complessa come lo sviluppo di un'applicazione grafica in Windows era virtualmente alla portata di tutti. Qui dobbiamo riconoscere tutta la bravura di Microsoft a comprendere il mercato dotandolo di uno strumento semplice, ma allo stesso tempo potente.

*Facilita' di accesso alle basi dati:* ogni programma la necessita' di memorizzare informazioni da qualche parte per poi recuperarle successivamente. Generalmente si usano file e/o basi dati (database). Alla Microsoft pensarono bene di dotare il VB3 di un accesso facile ad un semplice, ma allo stesso tempo efficace (se ben progettato), database. Stiamo parlando di Microsoft Access. Nel 1993, quando esce il VB3, Access e' ancora fermo alla versione 1.1: una versione funzionante ma non ancora del tutto matura. L'accoppiata vincente si avra' con la versione 2.0 di Access, nel 1994. Access merita sicuramente un discorso a parte e sicuramente trovera' spazio tra le pagine future di RetroMagazine.

## Programmazione ad eventi

Un'altra cosa di cui dobbiamo rendere grazie al Visual Basic e' l'apertura del concetto di programmazione ad eventi alla massa. A differenza della programmazione ad oggetti, che per molti programmatori abituati a lavorare con la programmazione strutturata puo' rimanere ostica, la programmazione ad eventi e' stato un concetto facilmente assimilabile. In pratica nella programmazione ad eventi il flusso logico del programma segue il verifi-

carsi degli eventi esterni che scateneranno l'esecuzione delle istruzioni legate a quel determinato evento.

Facciamo un esempio pratico: immaginate di avere una form dove avete piazzato un pulsante (button). Gli eventi legati al pulsante possono essere molteplici (click, double\_click, mouse\_down...) e ad ognuno di essi potrete legare del codice da eseguire quando l'evento si verifichera'. Nel caso di click, l'evento verra' scatenato quando con il mouse farete click con il tasto sinistro sul pulsante (button) ed il codice relativo verra' eseguito.

La cosa interessante e' che gli eventi possono anche essere definiti dal programmatore, ma questa e' un'altra storia.

## Tutto bello? Certo che no...

Ovviamente non era tutte rose e fiori. Visual Basic 3 funzionava abbastanza bene, ma per un utilizzo in ambito professionale mostrava ancora il fianco ad alcune limitazioni che facevano arrabbiare non poco gli sviluppatori.

*La distribuzione del programma:* una volta creato il codice e testato in ufficio, il tutto doveva essere impacchettato per essere installato sui computer dei clienti. E qui cominciarono i grattacapi. La distribuzione "dell'eseguibile" e delle librerie che lo avrebbero fatto funzionare non era il punto di forza del VB3. Spesso e volentieri qualche componente non veniva incluso nel pacchetto, altre volte le stesse librerie ma con versioni differenti erano gia' presenti nel computer ospite e sostituirle avrebbe reso inefficaci altri programmi gia' presenti, altre volte i dischetti con cui trasportavate il tutto si danneggiavano... Insomma una pena!

*Le stampe:* volenti o nolenti tutti i programmi negli anni '90 dovevano offrire la possibilita' di stampare. Diciamo che le stampe non erano proprio il forte di Visual Basic. Microsoft aveva gia' in casa un ottimo motore di stampa realizzato per Access che, opinione personale, avrebbe potuto essere facilmente incluso in Visual Basic. Inspiegabilmente questo non e' mai avvenuto (forse a causa di accordi commerciali con terze parti), lasciando le prime versioni di Visual Basic praticamente in 'brache di tela' per la gestione delle stampe e le successive versioni in mano a tool esterni come Crystal Report.

Per questo primo articolo non mi voglio dilungare oltre, ma, se siete interessati a provare (se non lo avete mai fatto prima) od a riutilizzare il Visual Basic 3.0 vi segnalo che potete scaricarlo dal sito winworldpc all'indirizzo: <https://winworldpc.com/product/microsoft-visual-bas/30>. Nello stesso sito sono presenti anche il database Access: <https://winworldpc.com/product/microsoft-access/10> ed il Crystal Report: <https://winworldpc.com/product/crystal-reports/3x>.

Il tutto puo' essere tranquillamente installato su una macchina virtuale con il S.O. Windows 95. Se non sapete come fare e volete una mano, chiedete pure.

A questo punto non mi resta che augurarvi buon divertimento con VB3, Access e Crystal Report e... appuntamento al prossimo articolo dove vedremo come scrivere qualche semplice programmino!





## Dark Side of 16 bit Ti99 - Il Basic esteso

di Ermanno Betori

Come già ampiamente descritto nel numero 16 di Retromagazine, la genesi del basic sul TI99/4A fu diversa rispetto agli altri computer dell'epoca. In questo articolo approfondiamo l'evoluzione del linguaggio partendo dalla versione del 1978 fino alla ultima che sta ancora avvenendo. Come già letto sappiamo che il TI99 aveva fundamentalmente due basic creati dalla casa madre Texas Instruments, il TI-Basic (1979) inserito di default nella consolle e l'Extended Basic (1981) venduto come cartuccia il quale aumentava le prestazioni in velocità quasi del 30-40%, estendeva di oltre 40 comandi il basic dando la possibilità dell'uso degli sprite, del floppy disk, miglior gestione del testo ecc.. del quale ne esistono due versioni la 100 che era piena di errori e la versione 110 dove molti di questi errori erano corretti ma molti altri rimanevano insoluti.



Altri dialetti Basic funzionanti sul TI99 creati da mamma T.I. furono il Cortex BASIC, che è una evoluzione del PowerBASIC usato sulla serie dei mini computer TI-990 e sostituisce in toto l'originale Basic del TI99, e delle estensioni dei comandi del TI-Basic che furono integrati sulle seguenti cartucce:

- Terminal Emulator II BASIC, presente nella cartuccia TE2 creata specificatamente per usare il modem, la quale addizionava al TI-Basic un numero limitato di comandi specialmente nell'uso dello speech synthesizer come si vede nei pochi programmi scritti appositamente per tale modulo.

- Mini-Memory, questa cartuccia permette la creazione di programmi in linguaggio assembler senza aver bisogno del PEBox, della espansione di memoria da 32K, del floppy disk come unità di memoria di massa ecc.. al contrario delle cartucce Editor Assembler, TI Writer word processor . All'epoca specialmente nel mercato tedesco furono creati molti programmi BASIC che sfruttavano le peculiarità della cartuccia Mini Memory al contrario di quello che accadeva negli U.S.A.



- Personal Record Keeping e Statistica furono cartucce create per creare un database personale e manipolazione dati le quali addizionavano al TI-basic le stesse istruzioni. Di programmi che hanno usato tali comandi supplementari ne abbiamo pochissime copie;



Per completezza anche la cartuccia del linguaggio LOGO aveva i suoi comandi basic supplementari;



Dopo il ritiro della T.I. dal mercato Home Computer avvenuto nel 1983 e conseguente abbandono del supporto sul TI99, molte ditte svilupparono una loro versione dell' Extended basic che vennero vendute specialmente nel mercato americano e tedesco. Menzioniamo le cartucce denominate MicroPal Extended BASIC, Exceltek Extended BASIC, e Mechatronic Extended BASIC che di fatto erano dei cloni della versione TI Version 110 Extended BASIC senza alcuna miglioria.





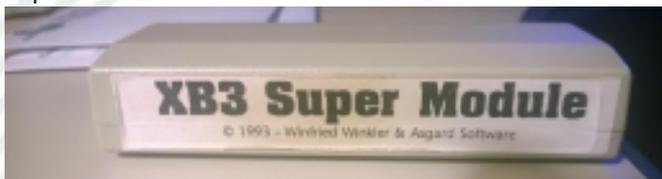
La cartuccia Extended Basic fu costruita usando sia delle EPROM/ROM che le GROM che erano delle ROM create esclusivamente dalla T.I. aventi al loro interno delle librerie di programmazione scritte in GPL (altro linguaggio proprietario T.I. che potremmo definire un Java antidiluviano).

Per migliorare le prestazioni e correggere i bugs che affliggevano l'XB originale bisognava modificare o ricreare le GROM ma non esisteva più il supporto da parte della T.I. pertanto varie ditte stavano cercavano una soluzione funzionale quando l'americana Miller Graphics presentò sul mercato la Gram Kracker.



Questa speciale cartuccia oltre ad avere una memoria supplementare di 80Kbyte permetteva la modifica delle GROM, il salvataggio su floppy del contenuto delle cartucce e loro riversamento ad altri supporti.

Grazie a questo device varie software house cominciarono a mettere le mani sul contenuto dell'extended basic, menzioniamo il Winfried Winkler's Extended BASIC III che fu la migliore implementazione e riscrittura del XB.



Creata da una software house tedesca e rivenduto negli U.S.A. dalla ditta Asgard correggeva i rimanenti errori presenti nel XB originale, ne migliorava le prestazioni fino al 50% e aggiungeva molti comandi supplementari. Tale XB3 venne inizialmente venduto su floppy e per funzionare aveva bisogno di un sistema TI completo di tutte le periferiche più la GRAM



Kracker. Solo nel 1993 riuscirono finalmente a creare la cartuccia stand alone cosa che ne limitò la diffusione. Contemporaneamente al Winfried Winkler's XB3 negli U.S.A. veniva commercializzato dalla Triton il Super Extended BASIC (che si identificava con la Version 120, IIRC, generalmente conosciuto come SEB) il quale aggiungeva molte estensioni presenti sulla GRAM Kracker al TI Extended BASIC.



Con il multi-module add-on creato da John Guion questa versione del XB come cartuccia divenne una delle più utili presenti sul mercato negli anni 80.

Ed./Assembler	Math Games VI	Touch Typing
DiskManager III	BlackJack & Poker	Hunt the Wampus
TI-Writer	Blasto	Meteor Multi...
Diagnostics	Verb Viper	Hangman
Video Game I	Card Shark	Yahtzee
Space Journey	Connect Four	Astro-Fighter
Adventure	Car Wars	

Bisogna aspettare fino al 2013/2014 per riavere sul mercato una cartuccia multi funzione dove l'utente ha tutti i programmi necessari a disposizione. L'ideatore della multi cartuccia Anthony Knerr ricreò prima delle cartridge contenenti gli originali Super Extended Basic (Triton), Winfried Winkler's XB3, e dopo la buona riuscita fece una multicartuccia contenente la sua versione personalizzata del Super Extended Basic cambiando e aggiungendo molte routine specialmente nella gestione dei floppy.

Come mostrato nella figura sottostante vediamo il XB27 insieme all'editor assembler (assemblatore), al ti writer (editor di testo) ed altre utility il tutto avente anche dei giochi.





Per completare le versioni del XB menzioniamo il J&KH Software's Super eXtended BASIC venduto su floppy (generalmente conosciuto come SXB da non confondere con il SEB by Triton) che aggiungeva un buon numero di routine all'originale TI Extended BASIC usando lo spazio della low memory presente nella espansione di memoria da 32K.

Del Myarc's Extended BASIC (1985) ne sono state create varie versioni fino ad arrivare all'ultima Vn 2.12 che è stata usata per creare sul Geneve l'Advanced Extended Basic che dopo varie vicissitudini è ancora in fase di sviluppo. Particolarità è che era insieme al XB3 il basic esteso che eseguiva i programmi TI-Basic anche molto lunghi e una buona gestione della grafica bitmap. Lo svantaggio di questa versione del basic è che per funzionare aveva bisogno della espansione di memoria da 512k creata sempre dalla Myarc perciò per funzionare l'utente doveva avere un sistema TI99 completo floppy,peb ecc.. più l'espansione da 512K!

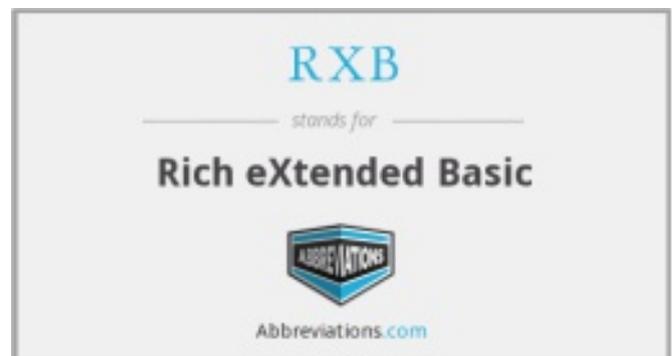


Con il prodotto tedesco Mechatronic Extended BASIC II Plus (cartridge color oro) che era la somma di un normale TI Extended BASIC (clone in cartridge argento) contenente le routine grafiche dell' APESoft Expanded Graphics BASIC negli anni 80 avevamo anche in Italia la possibilità di usare i comandi grafici per creare disegni in alta risoluzione. Uno poteva ottenere gli stessi risultati usando gli originali APESoft disks con il TI Extended BASIC (in Europa) o usando le Amerisoft Expanded Graphics BASIC routines (negli U.S.A.). Per usare le nuove routine grafiche l'utente aveva bisogno come al solito di un sistema completo avente possibilmente il PEB, 32K Ram , unità floppy. Inoltre il prezzo della cartuccia era nel 1985 estremamente caro, parliamo di circa 300.000 lire

(con 400.000 uno ci cominciava a comprare un C64) cosa che ne comportò una scarsa diffusione.



Finalmente finiamo con il Richard Lynn Gilbertson extended Basic. Questo basic ha di particolare che è stato creato usando il linguaggio proprietario della Texas Instruments GPL (Graphics Program Language) come negli anni 80. Lo possiamo definire l'evoluzione del XB3, in pratica ha avuto moltissime migliorie soprattutto nella gestione delle nuove periferiche tra cui la scheda di memoria SAMS da 1MB ed è il solo ad avere l'accesso all'Hard Disk.



Finalmente abbiamo finito questa lunga esposizione dei vari basic che possiamo usare sul TI99, cosa dire al neofita? Quale è il migliore? Indubbiamente verrebbe da dire che il Tony Knerr 2.7 basic e RXB 2015 (ancora in sviluppo) sono la scelta giusta, ma ciò non è detto in quanto molti utenti ancora usano l'originale TI EXT Basic in aggiunta a tool di sviluppo quali TML (The missing link) o XB256 ora inglobato nell' Extended Basic Game Developers Package "ISABELLA" TI-99/4A. Personalmente al neo TI99 User propongo di studiare e usare il RXB, per poi passare all'uso degli altri basic in modo da verificare le migliori ma anche le differenze specialmente per chi vuole creare videogames.

Tutti questi basic XB3,SEB,RXB,XB2.7 ecc. sono fruibili e usabili tramite l'emulatore Classic99, vedi il link seguente per il download dei libri, manuali, programmi ecc.. <https://atariage.com/forums/topic/153704-ti-994a-development-resources/>

Per chi avesse problemi ad usare il Classic99 e non riuscisse a configurare correttamente i vari basic può chiedere un aiuto qui in redazione.





# Introduzione ad HOLLYWOOD - 2a parte

di Gianluca Girelli

Questa serie di articoli sulla programmazione in Hollywood, e quelli che seguiranno sul linguaggio ARexx, sono stati scritti e pubblicati sulla rivista Bitplane tra il 2012 ed il 2013. Nel risistemarli per RetroMagazine ho cercato di riaggiornare date, versioni e link per scaricare eventuali risorse. Hollywood è un framework M.A.L. (Multimedia Application Layer), giunto oggi alla versione 8.0 che sarà a breve trattato su queste pagine.



## 1. INTRODUZIONE

Dopo l'introduzione alla "rinascita del multimedia" sui sistemi Amiga fatta nell'articolo precedente, iniziamo ad entrare nel vivo dell'argomento con un semplice programma che, coniugando grafica animata e suono, può dare meglio l'idea delle potenzialità "MAL" (Multimedia Application Layer) del linguaggio.

Il codice che di seguito analizzeremo potrà essere usato come screen saver anche se, non agganciandosi automaticamente al ciclo "idle" del sistema, dovrà essere lanciato direttamente dall'utente.

Vedremo come con poche righe sia possibile creare un prodotto dall'aspetto professionale, che farà invidia ai vostri amici. Una volta lanciato il programma infatti, una scena gotica composta da statue "viventi" si animerà sul vostro monitor rivelandone la vera natura, mentre la vostra canzone preferita (meglio se black metal!) suonerà in sottofondo.

Per prima cosa iniziamo scaricando la grafica da utilizzare dal sito all'indirizzo [...indirizzo di retromagazine...].

Ricordiamo che tale risorsa, essendo formalmente coperta da diritti d'autore, può essere utilizzata liberamente solo per finalità formative [rif. 1].

Come si può vedere dalla riproduzione della foto in queste pagine, la nostra "texture" è costituita da un fondale statico sul quale andremo a sovrapporre i nostri sprite animati [rif. 2].

Le animazioni si alterneranno in modo temporizzato (non meno di tre secondi tra una e l'altra) e pseudo-casuale, in modo da evitare che la stessa animazione venga utilizzata più volte consecutive rovinando così l'atmosfera della scena.

## 2. ANALISI DEL CODICE

Come tutti i programmi che si rispettino, il nostro codice inizia con una pesante sezione di commento, inclusa nella sequenza `"/* */`, che ha lo scopo di descrivere a grandi linee il programma e le fasi del suo sviluppo. Successivamente, questi commenti vengono poi riassunti ed integrati con l'uso di "pre-processor commands". Introdotti nel tutorial precedente e individuabili dal prefisso `"@"`, questi comandi servono a rendere immediatamente disponibile all'interprete del programma i dati voluti dal programmatore, in questo caso informazioni quali, tra le altre, la versione (`@APPVERSION`) e la descrizione di cosa fa il codice (`@APPDESCRIPTION`). Sotto AmigaOS, le stringhe argomento dei pre-processor commands vengono passate come testo della "commodity" che viene aggiunta ad "Exchange". Su altri sistemi, invece, vengono semplicemente salvate nella applet.

Le tre righe successive, contenenti appunto i comandi `@VERSION`, `@DISPLAY` e `@SCREEN`, dicono che il codice dovrà essere interpretato in un ambiente che sia dotato almeno della versione 4.5 di Hollywood per funzionare (al momento la Hollywood è arrivato alla release 8.0). Esse specificano anche che al lancio del programma questo sarà eseguito all'interno di una finestra (riscalabile e centrata sullo schermo) di 512x240 pixel, ma che un requester ci darà l'opportunità di visualizzare l'applicazione a tutto schermo (`"Mode=Ask"`), con una risoluzione minima di 640x480 punti.

In modalità "full screen" la nostra finestra non avrà bordi e quindi non avrà una barra del titolo.

Il comando successivo, `@MUSIC`, precarica in memoria il file sonoro da noi scelto. Non importa che tipo di file questo sia, ricordate solamente che se si tratta di un MP3, esso avrà bisogno per funzionare sui sistemi Amiga della libreria "mpega.library" che dovrà essere presente in `"System:Libs/"`. La compilazione per ambienti Windows o MacOS non ha richiede invece





plugins aggiuntivi.

Come in ogni linguaggio fortemente strutturato, a questo punto troviamo la dichiarazione di variabili e di procedure. In Hollywood le variabili sono dinamicamente allocate e non hanno bisogno di costrutti particolari che ne definiscano il tipo di dato. Possono essere inizializzate in qualsiasi punto del programma, ma lo stile di programmazione adottato, in cui creiamo una procedura che si occupa di definire e inizializzare variabili e sprite, ci consente di tenere il codice pulito ed ordinato facilitandone così la manutenzione.

La nostra funzione "function p\_InitVars()" fa in effetti proprio questo. Più nello specifico di occupa di costruire le nostre animazioni facciali mediante la combinazione delle istruzioni "BRUSH" e "SPRITE" nonché di definire le coordinate (contenute negli array "X" e "Y") alle quali le singole animazioni verranno visualizzate.

E' interessante notare che il codice poteva essere reso ancora più compatto usando direttamente il pre-processor command "@SPRITE" anziché la combinazione "BRUSH+CreateSprite()". Il punto è che in Hollywood l'animazione di oggetti mediante l'uso degli sprite non è compatibile con altre tecniche grafiche quali i "layers" o il "double buffering" (sia normale che "hardware accelerated"). La tecnica qui usata è quindi una garanzia di scalabilità del codice, in grado di facilitare la transizione al "double buffering" qualora questa dovesse rendersi necessaria.

La funzione successiva ("Function p\_MainLoop()") costituisce il cuore del nostro programma, in quanto si occupa di animare la scena mediante l'uso del comando "DisplaySprite()". Come detto in precedenza, vari test condizionali si occupano di rendere la scena il più credibile possibile evitando che lo stesso oggetto venga animato due volte consecutive o che comunque l'intervallo tra due qualsiasi animazioni successive appaia troppo regolare.

All'interno del MainLoop viene anche effettuato un controllo sull'esecuzione o meno del file sonoro come segue: "If IsMusicPlaying(1)=False Then PlayMusic(1)".

Questo approccio, benché più elaborato del semplice PlayMusic()/StopMusic(), ha due indubbi vantaggi: il primo è che la nostra musica continuerà a suonare automaticamente per tutto il tempo in cui il codice sarà in esecuzione. All'aumentare della complessità del codice non saremo quindi noi a doverla fermare o inizializzare uscendo da una routine ed entrando in un'altra; il secondo è che esso evita la saturazione dei canali audio all'aumentare delle sorgenti sonore.

Definite le subroutine che compongono il nostro programma è ora tempo di lavorare al programma vero e proprio ed è proprio qui che Hollywood dà il meglio di sé, perché in realtà tutto passa attraverso queste tre semplicissime righe di codice:

```
Repeat
  WaitEvent()
Forever
```

che da sole compiono il "miracolo". Come si vede, in questo loop infinito non si fa altro che aspettare il ve-

rificarsi di un evento. Questo evento altro non è che il nostro MainLoop(), il quale viene eseguito alla velocità di 25 fotogrammi al secondo grazie ad un'altra semplicissima ma potentissima istruzione:

```
SetInterval(1, p_MainLoop, 1000/25)
```

La potenza di Hollywood risiede nel fatto che, grazie a questa combinazione, il nostro codice girerà sempre alla stessa identica velocità a prescindere dal sistema su cui esso andrà in esecuzione, sia esso reale ("Classic", OS4.x, Morphos, AROS ..), emulato (Amithlon, WinUAE ..) oppure a prestazioni ben più spinte come i moderni sistemi desktop Win/Mac.

Allo stesso modo lo "scheduler" del programma gestirà il suono, che sarà sempre perfettamente sincronizzato con la parte grafica.

Non c'è virtualmente limite agli intervalli (e quindi agli eventi) che si possono gestire, anche se all'aumentare della complessità del programma sarà meglio utilizzare un approccio diverso e più strutturato, che sarà oggetto di trattazione separata.

### 3. CONCLUSIONI

Con il codice esaminato in questo articolo abbiamo fatto il primo vero passo nel mondo del multimedia nel quale, come sappiamo, il nostro "Amiga" era re incontrastato. Anche se oggi la tecnologia Amiga e le sue applicazioni non sono più "ground breaking" Hollywood è, a parere dello scrivente, un software in grado di fare la differenza tra un approccio semplicemente nostalgico ed uno invece che, seppure di nicchia, possa avere anche risvolti pratici o addirittura commerciali.

Basta infatti fare una ricerca in rete (specialmente su forum esteri quali amigaworld.net) per trovare varie ditte che usano Hollywood (sia la parte framework illustrata in questi primi due articoli, che la parte "Designer") nelle loro attività quotidiane. Sul sito ufficiale del programma, all'indirizzo <https://www.hollywood-mal.com/download.html>, è possibile scaricare il "player" per il proprio computer preferito. Il player non può essere usato per creare progetti (non potete, ad esempio, inserire manualmente il codice sottostante), ma può essere usato per eseguire applets compilate per il target di riferimento. A richiesta del lettore è quindi disponibile il codice compilato per l'architettura voluta.

Come si è visto, programmare con Hollywood è relativamente semplice e veloce. Una volta steso il codice di base, anche con l'aiuto dei numerosi esempi a corredo del software, l'enorme libreria di comandi e funzioni può espandere le funzionalità del programma sino all'infinito.

Nel prossimo tutorial cercheremo di affinare ulteriormente il nostro prodotto, trasformandolo in un visualizzatore di immagini che faranno bella mostra di sé all'interno della nostra cornice "gotica".

Arrivederci sul prossimo numero.



**Rif. 1 - DISCLAIMER!**

Il materiale (grafica e suono) usato per confezionare questo tutorial e' di proprieta' dei rispettivi autori, viene qui riprodotto solo a scopo didattico/formativo e NON PUO' essere usato per fini di lucro. L'autore dell'articolo ritiene che il prodotto si inquadri nell'ottica del "Fair Use" e che quindi nessun copyright siata stato violato. Maggiori informazioni sull'argomento possono essere reperite al seguente indirizzo:

[http://www.bghq.com/copyright\\_info.php](http://www.bghq.com/copyright_info.php)

**Rif. 2 - SPRITE**

In informatica uno sprite e' un oggetto bidimensionale, spesso animato, che viene integrato in uno scenario piu' ampio. Inizialmente usato per descrivere oggetti grafici gestiti separatamente dal bitmap dello sfondo video, il termine ha assunto poi un significato piu' lasco ed e' stato applicato per descrivere diversi tipi di sovrapposizione grafica. A seconda dei sistemi, gli sprite possono essere implementati sia via software che via hardware.

```

/*****
**
** Name:      Gothic Screensaver
** Author:    Gianluca Girelli
** Version:   1.1
** Date:      19.07.2011      Modified: October 2012
** Interpreter: Hollywood 4.5
** Licence:   Hollywood screensaver demo
** Function:  A screensaver-like demo code with gothic
**           background. The images are taken from PS1
**           game "Legacy of Kain: Soul Reaver".
**
** History:
**
** 1.1: (19.07.11)
** -implements a more intelligent use of @DISPLAY preprocessor
**   command featuring Hollywood autoscaling engine.
** It also uses @SCREEN to allow initial user choice of
** windowed or fullscreen display.
**
** 1.0: (14.07.11)
** - initial release
**
*****/

@APPTITLE      "Gothic Screensaver"
@APPAUTHOR    "Gianluca Girelli"
@APPCOPYRIGHT "Freeware - Graphics copyrights remain of original authors."
@APPVERSION   "$VER: 1.1 (19.07.2011)"
@APPDESCRIPTION "A screensaver-like demo done with Hollywood."

@VERSION 4,5
@DISPLAY 1, { Title = "'Gothic Screensaver' - Made on Amiga with Hollywood by g0blin",
             X=#CENTER, Y=#CENTER, width=512, Height=240, HidePointer=True,
             Sizeable=True, ScaleMode=#SCALEMODE_AUTO }
@SCREEN {Mode = "Ask", width = 640, Height = 480}

@MUSIC 1, "your_kick-ass_music_here.mp3"

Function p_InitVars()
;from brush 1 to brush 8 we have facial animations ...
@BRUSH 1, "mainmenu.png", {X=0, Y=241, width=343, Height=48, Transparency = $000080 }
@BRUSH 2, "mainmenu.png", {X=0, Y=289, width=343, Height=48, Transparency = $000080 }
@BRUSH 3, "mainmenu.png", {X=0, Y=338, width=343, Height=48, Transparency = $000080 }
@BRUSH 4, "mainmenu.png", {X=0, Y=387, width=343, Height=48, Transparency = $000080 }
@BRUSH 5, "mainmenu.png", {X=0, Y=437, width=343, Height=48, Transparency = $000080 }
@BRUSH 6, "mainmenu.png", {X=0, Y=486, width=455, Height=64, Transparency = $000080 }
@BRUSH 7, "mainmenu.png", {X=0, Y=551, width=343, Height=48, Transparency = $000080 }
@BRUSH 8, "mainmenu.png", {X=0, Y=600, width=343, Height=48, Transparency = $000080 }
;... while brush 10 is background image
@BRUSH 10, "mainmenu.png", {X=0, Y=0, width=512, Height=240, Transparency = $000080 }

;create all the sprites extracting the animation frames from "mainmenu.png"
CreateSprite(1, #BRUSH, 1, 49, 48, 7, 7)
CreateSprite(2, #BRUSH, 2, 49, 48, 7, 7)
CreateSprite(3, #BRUSH, 3, 49, 48, 7, 7)
CreateSprite(4, #BRUSH, 4, 49, 48, 7, 7)
CreateSprite(5, #BRUSH, 5, 49, 48, 7, 7)
CreateSprite(6, #BRUSH, 6, 65, 64, 7, 7)
CreateSprite(7, #BRUSH, 7, 49, 48, 7, 7)
CreateSprite(8, #BRUSH, 8, 49, 48, 7, 7)

```





```

; starts with animation frame 1
f = 1
; sprites position coordinates over brush 10
X = {235, 265, 273, 325, 340, 409, 382, 399}
Y = { 48, 96, 156, 49, 120, 14, 78, 150}
EndFunction

Function p_MainLoop()
; switch frames every 150ms
If GetTimer(1) > 150
If f > 6
f = 1
wait(50)
DisplaySprite(random, X[random-1], Y[random-1], f)
; the following loop avoids animating the same sprite twice in a row ...
random1=Rnd(8)+1
while random=random1 Do random1=Rnd(8)+1
random=random1
; ... then it wait at least 3 seconds before changing sprite
wait(Rnd(150)+150)
Else
f = f + 1
EndIf
StartTimer(1)
DisplaySprite(random, X[random-1], Y[random-1], f)
EndIf
If IsMusicPlaying(1)=False Then PlayMusic(1)
EndFunction

;===== Begin =====

EscapeQuit(True) ; enable ESCape

p_InitVars()

; display the background ...
DisplayBrush(10, 0, 0)

SetInterval(1, p_MainLoop, 1000/25) ; 25 fps

StartTimer(1)
random=Rnd(8)+1

; endless loop follows
Repeat
waitEvent()
Forever

;===== End =====

```

### ELENCO RISORSE

Sito Ufficiale Linguaggio Hollywood:  
<https://www.hollywood-mal.com>

### Sprites

[http://en.wikipedia.org/wiki/Sprite\\_\(computer\\_graphics\)](http://en.wikipedia.org/wiki/Sprite_(computer_graphics))

### ELENCO FOTOGRAFIE

Fondale statico e fotogrammi di animazione degli sprite.

Fonte: "Legacy of Kain: Soul Reaver", main menu, PSx





## Calcolare in multipla precisione-parte IV

di Alberto Apostolo

Con questo articolo si completa la serie di pubblicazioni sul tema del calcolo in multipla precisione che potete trovare nei precedenti numeri di RM (cfr. n.13 e poi 15,16,17).

Concluderemo in bellezza presentando una calcolatrice non scientifica che effettua calcoli in multipla precisione e un programma per calcolare i logaritmi in base e (il numero di Eulero).

L'idea di preparare una "calcolatrice" in multipla precisione nasce dopo avere riletto un articolo apparso su MC Microcomputer n.96 (Maggio 1990, pp.176-178) nel quale si presentava il software N-Math realizzato da Fabrizio Ermini, Stefano Marapodi, Mario Marini (all'epoca studenti presso la Facoltà di Ingegneria Elettronica dell'Università degli Studi di Firenze).

N-Math consisteva in un insieme di routine per il calcolo in precisione multipla realizzate in Fortran come parte integrante del corso di Calcolo Numerico e Programmazione.

Per compilare i programmi presentati in questo articolo (e nei precedenti) si ricorda che il vostro compilatore C deve essere almeno "C99 compliant" affinché sia possibile dichiarare variabili intere a 64 bit di tipo "long long int".

### IL PROGRAMMA MP\_CALC

Il programma MP\_CALC.C realizza una calcolatrice non scientifica con le quattro operazioni aritmetiche e un registro di memoria (accessibile con le funzioni STO, RCL, M+, M-).

Il main gestisce l'esecuzione delle operazioni gestite tramite i tasti funzione, mentre la routine Calc gestisce la visualizzazione del display, la pressione dei tasti e la digitazione

delle cifre sul display.

In effetti Calc non è altro che una versione modificata della routine Inpx introdotta nel numero 16 di RM.

Un'altra routine implementata per l'occasione è la routine Invx che calcola il reciproco di un numero in multipla precisione con il metodo di Newton-Raphson (cfr. RM 17) ed è utile per costruire la divisione tra due numeri a/b calcolando a\*(1/b).

Tuttavia calcolare a\*(1/b), usando la rappresentazione in virgola fissa e il troncamento delle cifre, espone la calcolatrice ad approssimazioni del risultato anche nel caso di operazioni molto semplici come 12 diviso 3 (vedi dimostrazione d'uso alla fine dell'articolo).

### IL PROGRAMMA MP\_LOGX

Il programma MP\_LOGX.C realizza il calcolo del logaritmo in base e=2.7182818... di un numero in multipla precisione facendo uso della formula di Taylor per il  $\log(1+x)$  in base e.

$$\log(1+x) = \sum_{n=0}^{\infty} \frac{(-1)^{n+1} x^n}{n}$$

Figura 1

Ma la formula di Taylor funziona solo per  $-1 < x \leq 1$ . Per aggirare l'ostacolo e calcolare  $\log(x)$  per qualunque x maggiore di zero, dobbiamo complicare un po' le cose riscrivendo  $\log(x)$  come riportato in figura 2 dove  $0 < \text{abs}(y) < 1/2$  in multipla precisione e u è in formato intero long.

$$\log(x) = \log 2^u (1+y) = u \log 2 + \log(1+y)$$

Figura 2

Con questo artificio,  $\log(1+y)$  converge piuttosto rapidamente. Il valore  $\log(2)$  si calcola tramite la relazione  $\log(2) = 2 \text{ sett tanh } 1/3$  dove la funzione "settor tang.

iperbolica"  $\text{sett tanh}(x)$  è anche chiamata "arcotangente iperbolica di x" (fig.3).

Lo pseudocodice di MP\_LOGX è mostrato in figura 4.

$$\text{sett tanh } x = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1} \quad -1 < x < +1$$

Figura 3

### Conclusione

Ora tocca a voi sbizzarrirvi. Se lo ritenete opportuno potete creare altre routine per arricchire la "libreria" oppure modificare le routine esistenti. Per calcolare altre funzioni (per esempio quelle trigonometriche o quelle iperboliche) si può ricorrere alle formule di Taylor che si trovano in qualunque testo di Analisi Matematica.

### Bibliografia

[Mic90] AA.VV., "N-Math: routine per il calcolo in precisione multipla", MC Microcomputer n. 96, pp.176-178, Maggio 1990. <https://issuu.com/adpware/docs/mc096>

[Mul06] J.M. Muller, "Elementary functions: algorithms and implementation", 2a ed., Springer, 2006.





```

ALGORITMO PER CALCOLARE LN(X) IN MULTIPLA PRECISIONE
NELLA FORMA LN(X)=U*LN(2)+LN(1+Y)

DATI x      ;ARGOMENTO DELLA FUNZIONE      (M.P.)
,h         ;CONTIENE IL VALORE LN(2)       (M.P.)
,b         ;VARIABILE DI LAVORO            (M.P.)
,u         ;COEFFICIENTE DI LN(2)
,y         ;VALORE UTILE PER LA SERIE LOG. (M.P.)
,p         ;VARIABILE DI SALVATAGGIO DI X  (M.P.)
,s         ;CONTIENE LN(X)

h=2*ARCTANH(1/3) ; SERIE ARCO TANG. IPERB. (M.P.)
b=1
p=x
u=0
SE p>=b ALLORA
  b=3/2
  RIPETI MENTRE p > b
    u=u+1
    p=p/2
  FINE-RIPETI
ALTRIMENTI
  b=1/2
  RIPETI MENTRE p < b
    u=u-1
    p=p*2
  FINE-RIPETI
FINE-SE
b=-1
y=p+b
h=u*LN(2)
s=h
s=s+LN(1+y); CALCOLARE CON LA SERIE LOGARITMICA
FINE-ALGORITMO

```

Figura 4

```

/*****
/* Programma: MP_CALC */
/* Funzione : Esempio di calcolatrice non scientifica in m.p. */
/*****
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <math.h>
#include <conio.h> /* tastiera-video PC IBM amb. Windows MS-DOS */
#define LMAX 25L /* ATTENZIONE! IN C RANGE = (0:LMAX- 1) */
#define BASE 1000000000L /* base aritmetica in multipla precisione */
#define NMAX 50000L /* numero max termini di una serie */
/* Dichiarazione dei tipi di variabile strutturati */
typedef struct { long segno; /* segno (-1, 0,+1) */
                long cifra[(LMAX+1)]; /* cifre in base beta */
                } MultPrec;
/* Dichiarazione di funzione e subroutine */
int LeggiTasto(void);
void Mulz(long,long,MultPrec *p_a,long *p_rp);
void Divz(long,long,MultPrec *p_a,long *p_rp);
void Suma(long,long,MultPrec *p_s,MultPrec *p_a,long *p_rp);
void Init(long,long,long,MultPrec *p_a,long *p_rp);
void DisplayMultPrec(long,long,MultPrec *p_a,char *s);
void Inpx(long beta,long l,long lpi,long lv,
          MultPrec *p_x,char *nomevb1);
void Mmpx(long beta,long l,long lpi,MultPrec *p_c,
          MultPrec *p_a,MultPrec *p_b,long *p_rp);
void Invx(long beta,long l,long lpi,MultPrec *p_c,
          MultPrec *p_u,long *p_rp);
void Comp(long beta,long l,MultPrec *p_x,MultPrec *p_y,long *p_cnf);
void Calc(long beta,long l,long lpi,long lv,
          MultPrec *p_x,
          MultPrec *p_m,
          MultPrec *p_y,
          long err,
          int *p_kb);

/* Programma principale */

```

```

int main(void)
{
  long beta = BASE; /* base aritmetica multipla precisione */
  long l = LMAX; /* lunghezza totale vettore mult.prec. */
  long lpi = 5L; /* lunghezza parte intera */
  long lv = l-0; /* numero elementi visualizzabili */
  MultPrec s,p,x,a,e,q,u,y,m; /* variabili multipla precisione */
  MultPrec *p_s =&s; /* puntatore var. mult.prec. */
  MultPrec *p_p =&p; /* puntatore var. mult.prec. */
  MultPrec *p_x =&x; /* puntatore var. mult.prec. */
  MultPrec *p_a =&a; /* puntatore var. mult.prec. */
  MultPrec *p_e =&e; /* puntatore var. mult.prec. */
  MultPrec *p_q =&q; /* puntatore var. mult.prec. */
  MultPrec *p_u =&u; /* puntatore var. mult.prec. */
  MultPrec *p_m =&m; /* puntatore var. mult.prec. */
  MultPrec *p_y =&y; /* puntatore var. mult.prec. */
  char nomevb1[6]=""; /* nome variabile m.p. (per la stampa) */
  long rp, *p_rp=&rp; /* riporto e puntatore riporto */
  long rd, *p_rd=&rd; /* resto e puntatore resto */
  long i; /* contatore (vb1. di lavoro) */
  long ovf, *p_ovf=&ovf; /* indicatore overflow e puntatore */
  int kb=0, *p_kb=&kb; /* tasto premuto */
  long err =0L; /* flag errore calcolatrice */
  long ope =0L; /* indica operazione attivata */

  /*
  _clrscr(); /* pulisci schermo */
  printf("Programma MP_CALC : INIZIO ELABORAZIONE\n");
  /*

  Init(beta,l,lpi,p_x,0L,p_rp);
  Init(beta,l,lpi,p_m,0L,p_rp);
  Init(beta,l,lpi,p_y,0L,p_rp);
  while((kb != 'q') && (kb != 'Q')){ /*q,Q per uscire */
    Calc(beta,l,lpi,lv,p_x,p_m,p_y,err,p_kb);
    if ((kb == 'c') || (kb == 'C')) {
      Init(beta,l,lpi,p_x,0L,p_rp);
      Init(beta,l,lpi,p_y,0L,p_rp);
      err =0;
      ope =0;
    }
    if (err == 0){
      if ((kb == 's') || (kb == 'S')) {m = x;}
      if ((kb == 'r') || (kb == 'R')) {x = m;}
      if ((kb == 'm') || (kb == 'M')) {
        u = m;
        Suma(beta,l,p_u,p_x,p_rp);
        if (rp != 0) {
          err = 1;
        } else {
          m = u;
        }
      }
    }
    if ((kb == 'n') || (kb == 'N')) {
      u = m;
      q = x;
      q.segno = -q.segno;
      Suma(beta,l,p_u,p_q,p_rp);
      if (rp != 0) {
        err = 1;
      } else {
        m = u;
      }
    }
    if ((kb == 'g') || (kb == 'G'))
      {y = x;ope=1;Init(beta,l,lpi,p_x,0L,p_rp);}
    if ((kb == 'h') || (kb == 'H'))
      {y = x;ope=2;Init(beta,l,lpi,p_x,0L,p_rp);}
    if ((kb == 'j') || (kb == 'J'))
      {y = x;ope=3;Init(beta,l,lpi,p_x,0L,p_rp);}
    if ((kb == 'k') || (kb == 'K'))
      {y = x;ope=4;Init(beta,l,lpi,p_x,0L,p_rp);}
    if ((kb == 'l') || (kb == 'L')) {
      if (ope == 1) {
        u=x;

```





```

        Suma(beta,1,p_y,p_u,p_rp);
        if (rp != 0) {err = 1;} else {x = y;ope=0;}
    }
    if (ope == 2) {
        u=x;
        u.segno = -u.segno;
        Suma(beta,1,p_y,p_u,p_rp);
        if (rp != 0) {err = 1;} else {x = y;ope=0;}
    }
    if (ope == 3) {
        ovf=0;
        u=x;
        Mmpx(beta,1,lpi,p_y,p_y,p_u,p_ovf);
        if (ovf != 0) {err = 1;} else {x = y;ope=0;}
    }
    if (ope == 4) {
        if (x.segno == 0) {
            err =1;
        } else {
            ovf=0;
            Invx(beta,1,lpi,p_u,p_x,p_ovf);
            Mmpx(beta,1,lpi,p_y,p_y,p_u,p_ovf);
            if (ovf != 0) {err = 1;} else {x = y;ope=0;}
        }
    }
}
}

/*
printf("\n\nProgramma MP_CALC : FINE ELABORAZIONE\n\n");
return 0;
}
/*-----*/
/* Funzione : visualizza numero in multipla precisione base 10^9 */
/*-----*/
void DisplayMultPrec(long lv,long lpi,MultPrec *p_a,char *nomevb1)
{
    long i,j;
    printf("\n%-6s = ",nomevb1);
    j=lpi%10L;
    if ((*p_a).segno > 0L) {printf("+");}
    if ((*p_a).segno == 0L) {printf(" ");}
    if ((*p_a).segno < 0L) {printf("-");}
    if (j != 0L) {
        for ( i = j ; i < 10L ; i++) {printf(" ");} /*10b1 */
    }
    for ( i = 1L ; i <= lv ; i++) {
        if (i == (lpi+1L)) {printf(".");} else {printf(" ");}
        printf("%9.9ld",(*p_a).cifra[i]);
        if ((i % 10L)==j) {printf("\n ");} /* 10 blank */
    }
}
/*-----*/
/* Funzione: moltiplicazione corta a=a*z in multipla precisione */
/*-----*/
void Mulz(long beta,long l,MultPrec *p_a,long z,long *p_rp)
{
    long i,izer,sz,az;
    long long int ipro,w;
    sz=(z > 0L)-(z < 0L); /* segno di z */
    az=sz*z; /* valore assoluto di z */
    izer=0L;
    *p_rp=0L;
    for (i = 1;i >= 1L;i--) {
        ipro=(*p_a).cifra[i]; /*workaround bug ottim. PellesC*/
        ipro = (ipro * az) + *p_rp;
        *p_rp=ipro/beta;
        w = beta; /*workaround bug ottim. PellesC*/
        (*p_a).cifra[i] = ipro - (*p_rp * w);
        if ((*p_a).cifra[i] > 0L) {izer= 1L;}
    }
    (*p_a).segno=(*p_a).segno * sz * izer;
}
/*-----*/
/* Funzione: divisione corta a=a/z in multipla precisione */

```

```

/*-----*/
void Divz(long beta,long l,MultPrec *p_a,long z,long *p_rd)
{
    long i,izer,sz,az;
    long long int idiv,w;
    sz=(z > 0L)-(z < 0L); /* segno di z */
    az=sz*z; /* valore assoluto di z */
    izer=0L;
    *p_rd=0L;
    for (i = 1L;i <= l;i++) {
        idiv = beta; /*workaround bug ottim. PellesC*/
        idiv = (*p_rd * idiv) + (*p_a).cifra[i];
        (*p_a).cifra[i]=idiv / az;
        w = az;
        *p_rd = idiv - ( (*p_a).cifra[i] * w);
        if ((*p_a).cifra[i] > 0L) {izer= 1L;}
    }
    (*p_a).segno=(*p_a).segno * sz * izer;
}
/*-----*/
/* Funzione: somma algebrica s=s+a in multipla precisione */
/*-----*/
void Suma(long beta,long l,MultPrec *p_s,MultPrec *p_a,long *p_rp)
{
    long i,isom,izer=0L,icnf=0L,az;
    *p_rp=0;
    if ((*p_a).segno != 0L) {
        if ((*p_s).segno == 0L) {
            for (i = 1L;i <= l;i++) {(*p_s).cifra[i]=(*p_a).cifra[i];}
            (*p_s).segno = (*p_a).segno;
        } else {
            az = (*p_s).segno * (*p_a).segno;
            if (az == -1L) {
                for (i = 1L;i <= l;i++) {
                    if ((*p_s).cifra[i] > (*p_a).cifra[i]) {icnf= 1L;}
                    if ((*p_s).cifra[i] < (*p_a).cifra[i]) {icnf=-1L;}
                    if (icnf != 0L) {
                        (*p_s).segno = (*p_s).segno * icnf;
                        break;
                    }
                }
            }
            for (i = 1;i >= 1L;i--) {
                if (az == -1L) {
                    if (icnf >= 0L) {
                        isom = *p_rp + (*p_s).cifra[i] - (*p_a).cifra[i];
                    } else {
                        isom = *p_rp + (*p_a).cifra[i] - (*p_s).cifra[i];
                    }
                } else {
                    isom = *p_rp + (*p_s).cifra[i] + (*p_a).cifra[i];
                }
                *p_rp = (isom >= beta) - (isom < 0L);
                (*p_s).cifra[i] = isom - *p_rp * beta;
                if ((*p_s).cifra[i] > 0L) {izer= 1L;}
            }
            (*p_s).segno=(*p_s).segno * izer;
        }
    }
}
/*-----*/
/* Funzione: inizializzazione con un numero z */
/*-----*/
void Init(long beta,long l,long lpi,MultPrec *p_a,long z,long *p_rp)
{
    long i,irp;
    (*p_a).segno=(z > 0L)-(z < 0L); /* segno di z */
    for (i = 1L;i <= l; i++){ (*p_a).cifra[i] = 0L; }
    *p_rp= (*p_a).segno*z; /* valore assoluto di z */
    for (i = lpi;i >= 1L;i--) {
        irp=(*p_a).cifra[i]+*p_rp;
        *p_rp=irp / beta;
        (*p_a).cifra[i] = irp - (*p_rp * beta);
    }
}
/*-----*/
/* Funzione: digitazione di un numero in m.p. */
/*-----*/

```





```

void Inpx(long beta,long l,long lpi,long lv,MultPrec *p_x,char *nomevbl)
{
    long i;          /* indice (vbl di lavoro) */
    long k;          /* indice (vbl di lavoro) */
    long m;          /* indice (vbl di lavoro) */
    long z=10L;      /* vbl di lavoro */
    long f=1L;       /* cifre in base beta */
    long flag_dec = 0L; /* 1=punto decimale attivo, 0=disatt. */
    long cont_dec = 0L; /* conta cifre dopo il punto decimale */
}
/*
int kbint = 0;      /* memorizza tasto premuto */
long rrp=0L, *p_rrp=&rrp; /* riporto e puntatore riporto */
long rrd=0L, *p_rrd=&rrd; /* resto e puntatore resto */
MultPrec w;        /* variabile multipla precisione */
MultPrec *p_w =&w; /* puntatore var. mult.prec. */

while (z<beta){f++;z=z*10L;}
Init(beta,l,lpi,p_w,0L,p_rrp);
while((kbint != 'q') && (kbint != 'Q')){ /*q,Q per uscire */
    _clrscr();
    for (i = 1;i >= 1L;i--) {
        if ((*p_w).cifra[i] > 0L) {
            if (w.segno == 0L) {w.segno = +1;}
        }
    }
    DisplayMultPrec(lv,lpi,p_w,nomevbl);
    printf("\n\n      (Q,q)= esci, (C,c)= cancella una cifra");
    printf(", (.)= punto decimale, (-) cambia segno\n\n");
    kbint = LeggiTasto();
    if (kbint == '.') {flag_dec = 1;}
    if (kbint == '-') {w.segno = -w.segno;}
    if (flag_dec == 0) {
        if ((kbint >= '0') && (kbint <= '9')
            && (w.cifra[1] < (beta/10L)) ) {
            Mulz(beta,l,p_w,10,p_rrp);
            w.cifra[lpi] = w.cifra[lpi] + kbint - 48;
        }
        if ((kbint == 'c') || (kbint == 'C')) {
            Divz(beta,lpi,p_w,10,p_rrd); /* si parte da lpi */
        }
    } else {
        if ((kbint >= '0') && (kbint <= '9')
            && (cont_dec < ((lv-lpi)*f)) ) {
            cont_dec++;
            z = (cont_dec-1) / f;
            k = lpi + z + 1;
            m = f - (cont_dec-1) + z*f;
            z=1;
            for (i = 1L;i<m;i++) {z=z*10L;}
            w.cifra[k] = w.cifra[k] + (kbint - 48)*z;
        }
        if ((kbint == 'c') || (kbint == 'C')) {
            z = (cont_dec-1) / f;
            k = lpi + z + 1;
            m = f - (cont_dec-1) + z*f;
            z=10;
            for (i = 1L;i<m;i++) {z=z*10L;}
            w.cifra[k] = w.cifra[k] - (w.cifra[k] % z);
            cont_dec--;
            if (cont_dec == 0){flag_dec=0;}
        }
    }
}
}
(*p_x) = w;
}
/*-----*/
/* Funzione realizzata : */
/* restituisce all'esterno il codice ASCII di un tasto premuto */
/*-----*/
int LeggiTasto(void)
{
    int c = 0; /* carattere */
    while (_kbhit()) _getch(); /* finche' ci sono caratteri */
    /* li legge e li ignora */
    c = _getch(); /* legge un tasto */
    return c;
}

```

```

}
/*-----*/
/* Funzione: moltiplicazione tra numeri in m.p. c = a * b */
/*-----*/
void Mmpx(long beta,long l,long lpi,MultPrec *p_c
          ,MultPrec *p_a,MultPrec *p_b,long *p_ovf)
{
    long cifra[(2*l)+1]; /* buffer cifre in base beta */
    long i,j,k;          /* indici (vbl di lavoro) */
    long rr;             /* vbl riporto */
    long izer = 0L;      /* flag = 1 se cifra<0, 0 altrimenti */
    long long int ipro; /* vbl lavoro */

    for (i=(1+1L);i<=(2L*1)+1;i++){cifra[i]=0L;}
    *p_ovf=0L;
    for (i = 1;i >= 1L;i--) {
        rr=0L;
        if ((*p_b).cifra[i]>0L) {
            for (j = 1;j >= 1L;j--) {
                k=i+j;
                ipro=(*p_b).cifra[i]; /*workaround Pellesc*/
                ipro=rr+cifra[k]+((*p_a).cifra[j] * ipro);
                rr=ipro/beta;
                cifra[k]=ipro % beta;
            }
        }
        cifra[i]=rr;
    }
    for (i = 1L;i <= lpi;i++) {
        if (cifra[i]>0L) {
            *p_ovf=1L;
            break;
        }
    }
    for (i = 1L;i <= 1;i++) {
        (*p_c).cifra[i]=cifra[i+lpi];
        if ((*p_c).cifra[i]>0L) { izer=1L; }
    }
    (*p_c).segno=(*p_a).segno * (*p_b).segno * izer;
}
/*-----*/
/* Funzione: confronto due numeri x e y in multipla precisione */
/*-----*/
void Comp(long beta,long l,MultPrec *p_x,MultPrec *p_y,long *p_cnf)
{
    long i;
    *p_cnf = 0L;
    for (i = 1L;i <= 1;i++) {
        if ((*p_x).cifra[i] > (*p_y).cifra[i]) {*p_cnf= 1L;}
        if ((*p_x).cifra[i] < (*p_y).cifra[i]) {*p_cnf=-1L;}
        if (*p_cnf != 0L) { break; }
    }
}
/*-----*/
/* Funzione: digitazione di un numero in m.p. e tasti calcolatrice */
/*-----*/
void Calc(long beta,long l,long lpi,long lv,
          MultPrec *p_x,
          MultPrec *p_m,
          MultPrec *p_y,
          long err,
          int *p_kb)
{
    long i;          /* indice (vbl di lavoro) */
    long k;          /* indice (vbl di lavoro) */
    long m;          /* indice (vbl di lavoro) */
    long z=10L;      /* vbl di lavoro */
    long f=1L;       /* cifre in base beta */
    long flag_dec = 0L; /* 1=punto decimale attivo, 0=disatt. */
    long cont_dec = 0L; /* conta cifre dopo il punto decimale */
}
/*
int kbint = 0;      /* memorizza tasto premuto */
long rrp=0L, *p_rrp=&rrp; /* riporto e puntatore riporto */
long rrd=0L, *p_rrd=&rrd; /* resto e puntatore resto */
MultPrec w;        /* variabile multipla precisione */
MultPrec *p_w =&w; /* puntatore var. mult.prec. */
MultPrec r1;       /* variabile multipla precisione */

```





```

MultPrec *p_r1 =&r1; /* puntatore var. mult.prec. */
MultPrec r2; /* variabile multipla precisione */
MultPrec *p_r2 =&r2; /* puntatore var. mult.prec. */

char vbl[6]=""; /* nome variabile m.p. (per la stampa) */

while (z<beta){f++;z=z*10L;}
w=(*p_x);
while((kbint != 'q') && (kbint != 'Q') &&
(kbint != 's') && (kbint != 'S') &&
(kbint != 'r') && (kbint != 'R') &&
(kbint != 'm') && (kbint != 'M') &&
(kbint != 'n') && (kbint != 'N') &&
(kbint != 'c') && (kbint != 'C') &&
(kbint != 'g') && (kbint != 'G') &&
(kbint != 'h') && (kbint != 'H') &&
(kbint != 'j') && (kbint != 'J') &&
(kbint != 'k') && (kbint != 'K') &&
(kbint != 'l') && (kbint != 'L')
){
_crlrscr();
for (i = 1;i >= 1L;i--) {
if ((*p_w).cifra[i] > 0L) {
if (w.segno == 0L) {w.segno = +1;}
}
}
printf("\n\n CALCOLATRICE IN MULTIPLA PRECISIONE \n");
strcpy(vbl, "Disp.");
DisplayMultPrec(1v,1pi,p_w, vbl);
printf("\n");
r1 = (*p_m);
strcpy(vbl, "Mem.M");
DisplayMultPrec(1v,1pi,p_r1,vbl);
printf("\n");
r2 = (*p_y);
strcpy(vbl, "Reg.y");
DisplayMultPrec(1v,1pi,p_r2,vbl);
printf("\n (Q,q)= esci, (D,d)= cancella una cifra");
printf(", (.) = punto decimale, (X,x)= cambia segno\n");
printf(" (S,s)= STO, (R,r)= RCL, ", "");
printf("(C,c)= CE/C, (M,m)= M+, (N,n)= M- \n");
printf(" (G,g)= +, (H,h)= - ,");
printf(" (J,j)= *, (K,k)= /, (L,l)= uguale \n\n");
if (err != 0) {
printf("\n *** ERRORE ***\n");
}
kbint = LeggiTasto();
if (err != 0){break;}
if (kbint == '.') {flag_dec = 1;}
if ((kbint == 'x') || (kbint == 'X')) {w.segno = -w.segno;}
if (flag_dec == 0) {
if ((kbint >= '0') && (kbint <= '9')
&& (w.cifra[1] < (beta/10L)) ) {
Mulz(beta,1,p_w,10,p_rrp);
w.cifra[1pi] = w.cifra[1pi] + kbint - 48;
}
if ((kbint == 'd') || (kbint == 'D')) {
Divz(beta,1pi,p_w,10,p_rrd); /* si parte da lpi */
}
} else {
if ((kbint >= '0') && (kbint <= '9')
&& (cont_dec < ((1v-1pi)*f)) ) {
cont_dec++;
z = (cont_dec-1) / f;
k = 1pi + z + 1;
m = f - (cont_dec-1) + z*f;
z=1;
for (i = 1L;i<m;i++) {z=z*10L;}
w.cifra[k] = w.cifra[k] + (kbint - 48)*z;
}
if ((kbint == 'd') || (kbint == 'D')) {
z = (cont_dec-1) / f;
k = 1pi + z + 1;
m = f - (cont_dec-1) + z*f;
z=10;

```

```

for (i = 1L;i<m;i++) {z=z*10L;}
w.cifra[k] = w.cifra[k] - (w.cifra[k] % z);
cont_dec--;
if (cont_dec == 0){flag_dec=0;}
}
}
}
(*p_x) = w;
(*p_kb) = kbint;
}
}
/*-----*/
/* Funzione: digitazione di un numero in m.p. c = 1 / u */
/*-----*/
void Invx(long beta,long l,long lpi,MultPrec *p_c
,MultPrec *p_u,long *p_ovf)
{
long i; /* indice (vbl di lavoro) */
long wovf=0L,*p_wovf=&wovf; /* indicatore overflow */
long iz = 0L; /* posizione prima cifra != zero di x */
long ww; /* variabile di lavoro */
long cnf,*p_cnf=&cnf; /* indicatore di confronto */
MultPrec wa; /* variabile multipla precisione */
MultPrec *p_wa=&wa; /* puntatore var. mult.prec. */
MultPrec s,p,x,a,e,q; /* variabili multipla precisione */
MultPrec *p_s=&s; /* puntatore var. mult.prec. */
MultPrec *p_p=&p; /* puntatore var. mult.prec. */
MultPrec *p_x=&x; /* puntatore var. mult.prec. */
MultPrec *p_a=&a; /* puntatore var. mult.prec. */
MultPrec *p_e=&e; /* puntatore var. mult.prec. */
MultPrec *p_q=&q; /* puntatore var. mult.prec. */
long rp, *p_rp=&rp; /* riporto e puntatore riporto */
long rd, *p_rd=&rd; /* resto e puntatore resto */

Init(beta,l,1pi,p_e,0L,p_rp);
e.cifra[1-1pi] = 1L; /* inizializzazione variabile di err. */
wa = (*p_u);
if (wa.segno == 0L) {
wovf = 1L;
} else {
for (i = 1L ;i <= l;i++) {
if (wa.cifra[i] != 0L) {
iz = i;
ww = wa.cifra[i];
break;
}
}
Init(beta,l,1pi,p_x,1L,p_rp);
Divz(beta,1,p_x,ww,p_rd);
if ( iz < 1pi ) {
for (i = iz;i < 1pi;i++) {
Divz(beta,1,p_x,beta,p_rd);
}
} else {
for (i = 1pi;i < iz;i++) {
Mulz(beta,1,p_x,beta,p_rp);
if (rp != 0L) {
wovf = 1L; /* overflow calcolo valore di innesco */
break;
}
}
}
}
if ( wovf != 1L ){
for (i = 1L;i < NMAX;i++){
p=x;
q=x;
Mmpx(beta,1,1pi,p_p,p_p,p_wa,p_wovf);
if (wovf != 0L){break;}
Init(beta,1,1pi,p_s,2L,p_rp);
p.segno = -p.segno;
Suma(beta,1,p_s,p_p,p_rp);
if (rp != 0L){break;}
Mmpx(beta,1,1pi,p_x,p_x,p_s,p_wovf);
if (wovf != 0L){break;}
q.segno = -q.segno;
Suma(beta,1,p_q,p_x,p_rp);
Comp(beta,1,p_q,p_e,p_cnf);
}
}
}
}

```





```

        if (cnf == -1){break;}
    }
}
(*p_ovf) = wovf;
(*p_c) = x;
}

/*****
/* Programma: MP_LOGX
/* Funzione : Calcolo ln(x) in multipla precisione
*****/
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <conio.h> /* tastiera-video PC IBM amb. Windows MS-DOS */
#define LMAX 22L /* ATTENZIONE! IN C RANGE = (0:LMAX- 1) */
#define BASE 1000000000L /* base aritmetica in multipla precisione */
#define NMAX 5000L /* numero max termini di una serie */
/* Dichiarazione dei tipi di variabile strutturati
typedef struct { long segno; /* segno (-1, 0,+1)
long cifra[(LMAX+1)]; /* cifre in base beta

} MultPrec;
/* Dichiarazione di funzione e subroutine
int LeggiTasto(void);
void Mulz(long,long,MultPrec *p_a,long,long *p_rp);
void Divz(long,long,MultPrec *p_a,long,long *p_rp);
void Suma(long,long,MultPrec *p_s,MultPrec *p_a,long *p_rp);
void Init(long,long,long,MultPrec *p_a,long,long *p_rp);
void DisplayMultPrec(long,long,MultPrec *p_a,char *s);
void Inpx(long beta,long l,long lpi,long lv,
MultPrec *p_x,char *nomevb1);
void Mmpx(long beta,long l,long lpi,MultPrec *p_c
,MultPrec *p_a,MultPrec *p_b,long *p_rp);
void Comp(long beta,long l,MultPrec *p_x,MultPrec *p_y,long *p_cnf);

/* Programma principale
int main(void)
{
long beta = BASE; /* base aritmetica multipla precisione */
long l = LMAX; /* lunghezza totale vettore mult.prec. */
long lpi = 3L; /* lunghezza parte intera */
long lv = l-2; /* numero elementi visualizzabili */
MultPrec x,h,b,y,p; /* variabili multipla precisione */
MultPrec s; /* variabili multipla precisione */
MultPrec q; /* variabili multipla precisione */
MultPrec *p_x =&x; /* puntatore var. mult.prec. */
MultPrec *p_h =&h; /* puntatore var. mult.prec. */
MultPrec *p_b =&b; /* puntatore var. mult.prec. */
MultPrec *p_y =&y; /* puntatore var. mult.prec. */
MultPrec *p_p =&p; /* puntatore var. mult.prec. */
MultPrec *p_s =&s; /* puntatore var. mult.prec. */
MultPrec *p_q =&q; /* puntatore var. mult.prec. */
char nomevb1[6]=""; /* nome variabile m.p. (per la stampa) */
long rp, *p_rp=&rp; /* riporto e puntatore riporto */
long rd, *p_rd=&rd; /* resto e puntatore resto */
long i,u; /* contatore (vbl. di lavoro) */
long ovf, *p_ovf=&ovf; /* indicatore overflow e puntatore */
long cnf,*p_cnf=&cnf; /* indicatore di confronto */
long isk; /* scambiatore di segno */
long z; /* scambiatore di segno

/*
_clrscr(); /* pulisci schermo
printf("Programma MP_LOGX : INIZIO ELABORAZIONE\n");

/*

strcpy(nomevb1,"x ");
Inpx(beta,l,lpi,lv,p_x,nomevb1);

if (x.segno <= 0L) {
printf("x <= zero! Impossibile proseguire\n");
} else {
/* calcola ln(2) = 2*arctanh(1/3) */

```

```

/* */
Init(beta,l,lpi,p_q,1L,p_rp);
Init(beta,l,lpi,p_h,1L,p_rp);
for (i = 3L ; i <= NMAX;i=i+2L) {
Divz(beta,l,p_q,9L,p_rd);
p=q;
Divz(beta,l,p_p,i,p_rd);
if (q.segno == 0L) {break;}
Suma(beta,l,p_h,p_p,p_rp);
}
Divz(beta,l,p_h,3L,p_rd);
Mulz(beta,l,p_h,2L,p_rp);
/* */
/* calcola u */
/* */
p=x;
Init(beta,l,lpi,p_q,1L,p_rp);
u=0L;
Comp(beta,l,p_p,p_q,p_cnf);
if (cnf >= 0){
Init(beta,l,lpi,p_q,3L,p_rp);
Divz(beta,l,p_q,2L,p_rd);
Comp(beta,l,p_p,p_q,p_cnf);
while(cnf > 0){
u++;
Divz(beta,l,p_p,2L,p_rd);
Comp(beta,l,p_p,p_q,p_cnf);
}
} else {
Init(beta,l,lpi,p_q,1L,p_rp);
Divz(beta,l,p_q,2L,p_rd);
Comp(beta,l,p_p,p_q,p_cnf);
while(cnf < 0){
u--;
Mulz(beta,l,p_p,2L,p_rp);
Comp(beta,l,p_p,p_q,p_cnf);
}
}
/* */
/* calcola il logaritmo */
/* */
y=p;
Init(beta,l,lpi,p_b,-1L,p_rp);
Suma(beta,l,p_y,p_b,p_rp);
Mulz(beta,l,p_h,u,p_rp);
s=h;
Init(beta,l,lpi,p_p,1L,p_rp);
isk=1L;
for (i = 1L ; i <= NMAX; i++) {
Mmpx(beta,l,lpi,p_p,p_p,p_y,p_ovf);
if (ovf != 0L){printf("Overflow 1");break;}
q=p;
z=isk*i;
Divz(beta,l,p_q,z,p_rd);
if (q.segno == 0L) {break;}
isk=-isk;
Suma(beta,l,p_s,p_q,p_rp);
if (ovf != 0L){printf("Overflow 2");break;}
}
strcpy(nomevb1,"ln(x)");
DisplayMultPrec(lv,lpi,p_s,nomevb1);
}

printf("\n\nTermini utilizzati = %5d",i);

/*
printf("\n\nProgramma MP_LOGX : FINE ELABORAZIONE\n\n");
return 0;
}

```

Attenzione! Il programma MP\_LOGX.C ha solo il main() perché le routine sono le stesse del programma MP\_CALC.C





In questa fase è stata impostata l'operazione 12 diviso 3 :

CALCOLATRICE IN MULTIPLA PRECISIONE

```
Disp. = +          00000000 00000000 00000000 00000000 00000003
.00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

```
Mem.M =          00000000 00000000 00000000 00000000 00000000
.00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

```
Reg.y = +          00000000 00000000 00000000 00000000 00000012
.00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

(Q,q)= esci, (D,d)= cancella una cifra, (.) = punto decimale , (X,x)= cambia segno  
 (S,s)= STO , (R,r)= RCL , (C,c)= CE/C, (M,m)= M+, (N,n)= M-  
 (G,g)= + , (H,h)= - , (J,j)= \* , (K,k)= / , (L,l)= uguale

Situazione dopo la pressione del tasto funzione "uguale":

CALCOLATRICE IN MULTIPLA PRECISIONE

```
Disp. = +          00000000 00000000 00000000 00000000 00000003
.99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999
99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999
```

```
Mem.M =          00000000 00000000 00000000 00000000 00000000
.00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

```
Reg.y = +          00000000 00000000 00000000 00000000 00000003
.99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999
99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999
```

(Q,q)= esci, (D,d)= cancella una cifra, (.) = punto decimale , (X,x)= cambia segno  
 (S,s)= STO , (R,r)= RCL , (C,c)= CE/C, (M,m)= M+, (N,n)= M-  
 (G,g)= + , (H,h)= - , (J,j)= \* , (K,k)= / , (L,l)= uguale

Programma MP\_CALC : FINE ELABORAZIONE

Dimostrazione d'uso del programma per calcolare il logaritmo in base e di un numero:

C:\Users\Utente\Desktop\C\_Esercizi> MP\_LOGX

```
x = +          00000000 00000000 00000010
.00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

(Q,q)= esci, (C,c)= cancella una cifra, (.)= punto decimale, (-) cambia segno

```
ln(x) = +          00000000 00000000 00000002
.302585092 994045684 017991454 684364207 601101488 628772976 033327900 967572609 677352480 235997205
089598298 341967784 042286248 633409525 465082806 756666287 369098781
```

Termini utilizzati = 280

Programma MP\_LOGX : FINE ELABORAZIONE

C:\Users\Utente\Desktop\C\_Esercizi>

Dimostrazione d'uso dei programmi MP\_CALC.C e MP\_LOGX.C



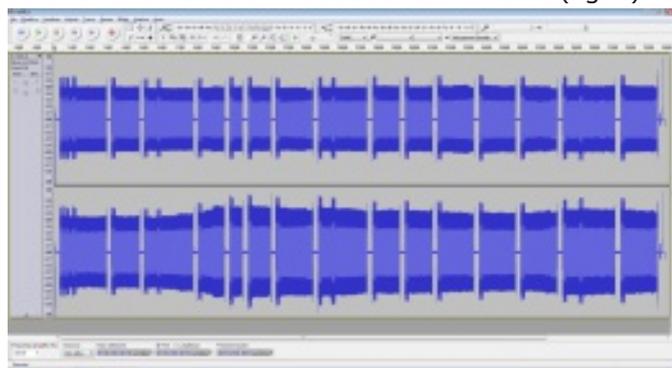


## Il dumping da nastro

di The Dump Club 64 Team

# THE DUMP CLUB 64

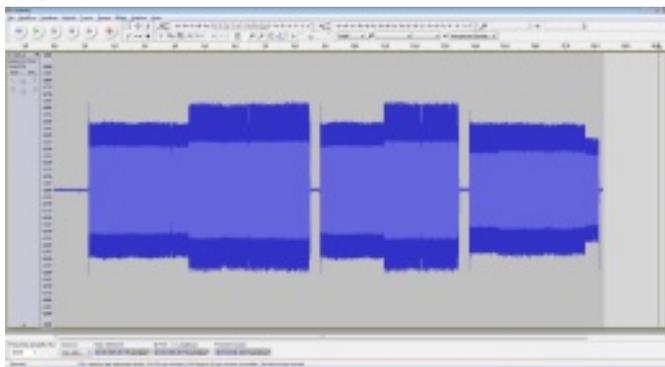
I programmi per C64 venivano memorizzati, oltre che su cartucce e su floppy disk, sui nastri magnetici delle audio cassette (le stesse su cui veniva registrata la musica): questo supporto aveva il vantaggio di poter contenere un buon quantitativo di dati e di essere poco costoso. Anche i dispositivi per riprodurre le cassette erano molto economici, soprattutto se paragonati ai costosissimi disk drive. La diffusione di questi supporti fu agevolata soprattutto dalla facilità di duplicazione, consentita sia da alcuni dispositivi ad hoc (i cosiddetti duplicatori) che dalle normali (ma costose e meno affidabili) doppie piastre di registrazione. Utilizzare le cassette, però, non era sempre facile: trattandosi di dati "binari", la mancata (o non corretta) lettura anche di un solo bit poteva compromettere il caricamento. Bisognava in tal caso procedere alla pulizia della testina del registratore, se non a modificarne l'allineamento... per poi, magari, scoprire che il problema era dovuto al cattivo funzionamento del meccanismo di trascinamento del nastro, che spesso causava la fuoriuscita (e la rottura, in casi estremi ma purtroppo non rarissimi) dello stesso. I nastri, inoltre, con il tempo tendono a rovinarsi - come tutti i supporti magnetici -, soprattutto se non accuratamente conservati: per questo motivo è molto importante preservarli al più presto, digitalizzandone il contenuto tramite la procedura del "dumping". Il modo più artigianale (ma non per questo più semplice) per effettuare un "dump" è il cosiddetto "metodo audio", che consiste nel riversare su computer il contenuto audio della cassetta per poi elaborarlo con appositi programmi. Tutto ciò che occorre è un registratore audio, che va collegato alla scheda sonora del computer nella presa "Line In" (o, in mancanza, in quella per il microfono). Per registrare ed elaborare la traccia audio è sufficiente un programma di editing gratuito come Audacity (<https://www.audacityteam.org/>). Prima di registrare occorre regolare il volume d'ingresso del segnale: se il registratore è dotato di regolazione del volume è bene agire dapprima su di esso, portando il livello d'uscita quasi al massimo; tramite il monitor di Audacity (o del programma di editing che vi è più congeniale) dovrete poi regolare il livello di entrata facendo in modo che non vada in distorsione. (fig. 1)



**Fig. 1 - Regolazione del livello di entrata**

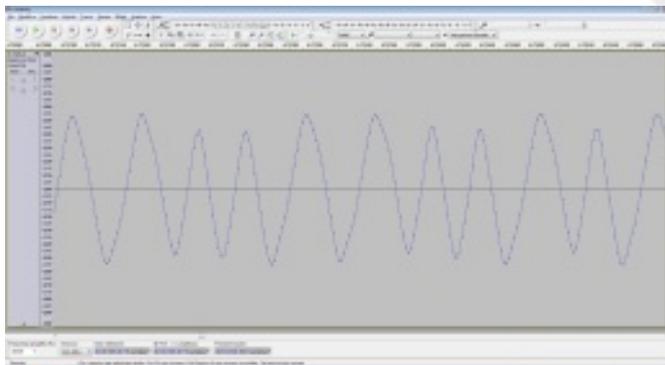
I parametri da impostare per la registrazione sono la frequenza di campionamento (44.100hz o 48.000hz sono più che sufficienti) e la risoluzione (16 bit). Prima di cominciare l'acquisizione è opportuno riavvolgere il nastro un paio di volte, e pulire la testina di lettura con un bastoncino di cotone idrofilo imbevuto di poco alcool.

Il segnale audio è mono, ma se il registratore ha l'uscita stereo (dò per scontato che il cavetto sia stereo da entrambi i lati) otterrete due canali distinti: dopo aver ultimato la registrazione vi conviene salvarla e lavorare su uno dei due canali; se sul nastro sono presenti più programmi, potrete anche copiarli e incollarli uno alla volta in file separati (ricordandovi di prendere un solo canale): solitamente tra un programma e l'altro ci sono alcuni secondi di silenzio (fig 2). I file così ottenuti vanno elaborati tramite il



**Fig. 2 - Secondi di silenzio tra i programmi**

programma Audiotap (<http://wav-prg.sourceforge.net/index.html>), cliccando su "Create a TAP file" e selezionando "from an audio file". Se la vostra scheda audio inverte le forme d'onda dovrete anche barrare l'apposita casella. Le "Advanced Options" andrebbero dapprima lasciate così come sono (a parte la frequenza di campionamento, che deve coincidere con quella della registrazione); se la conversione non dovesse andare a buon fine (e dopo aver escluso che non si tratti delle forme d'onda invertite - fig 3) si può



**Fig. 3 - Escludiamo le forme d'onda invertite**

provare a modificare il livello di "Sensitivity", utilizzando valori intorno ad 80 per nastri non ben conservati; se il nastro è in buone condizioni, è sufficiente un valore tra il 10 ed il 20. Una volta





effettuate le varie scelte, il programma vi dà la possibilità di dare un nome al file .TAP e di scegliere la versione del formato TAP da utilizzare (se state dumpando un nastro per C64 o VIC-20 dovete utilizzare la versione 1, mentre per il C16 andrebbe utilizzata la 2). Infine, non vi resta che provare a caricare il file .TAP con un emulatore, per verificarne il funzionamento.

La conversione può anche essere effettuata "al volo", selezionando l'opzione "Create a TAP file from a sound", ma è preferibile utilizzare il sistema che ho illustrato sopra: il file .WAV, infatti, si rivela prezioso nel caso in cui il dump non sia andato a buon fine: innanzitutto, ascoltando l'audio potrete capire se il nastro girava alla giusta velocità: il tono del segnale iniziale deve corrispondere ad un MI sovracuto. Nel caso in cui il tono sia più basso potrete facilmente modificare velocità ed intonazione dell'intero file tramite l'apposita funzione di Audacity. Se invece il tono non è stabile non avete altra scelta che effettuare nuovamente l'acquisizione, magari dopo aver riavvolto la cassetta... con la speranza che non sia proprio la cassetta ad essere difettosa all'origine. Può capitare, infatti, di fare decine di tentativi e di acquisizioni, per accorgersi magari che il nastro è irrecuperabile ma fortunatamente il software e l'hardware disponibili ai giorni nostri sono di grande aiuto, sia nella manipolazione del segnale audio che nell'acquisizione vera e propria.

Come ogni opera umana anche il dumping ha bisogno di pratica e di qualche insuccesso prima di poter essere padroneggiata con maestria. Indubbiamente è un'attività, soprattutto se fatta con amore e dedizione, in grado di regalare grandi gioie e quell'incomparabile senso di soddisfazione che ripaga di tanto sforzo e lavoro! Nella nostra attività iniziata da pochi mesi abbiamo trovato, tra tantissimo software ampiamente disponibile, qualche piccolo gioiellino che era rimasto ignorato in attesa del definitivo oblio.

A chiusura dei nostri articoli tecnici sul dumping abbiamo così deciso di aggiungere, da ora in avanti, alcuni dei successi più significativi del nostro lavoro per coinvolgere tutti i lettori di RetroMagazine in quella che riteniamo la nobile causa della preservazione dei vecchi supporti magnetici.

Nelle sessioni finora svolte abbiamo trovato alcuni applicativi italiani di cui si erano perse le tracce: Tetraclone v5.4 (una suite di copiatori e utility disco), Recupero Files (un Undelete ante litteram per il c64),



Fig. 4 - Software della tastiera CMK49 della SIEL



Fig. 5 - Basotter: espansione BASIC per plotter

alcuni gestionali della Sirius Elettronica (Gestione Condominio, Gestione Fatture, Agenda telefonica, Gestione Conti Correnti, Analisi Vendite), i dischi e le cassette originali allegate alla tastiera CMK49 della SIEL (fig. 4) gentilmente donate da Gianluca De Angelis, virtuoso musicista e membro del nostro DumpClub64, e poi alcuni titoli della Leoni Informatica Milano (Contabilità Semplice, Gestione Conti, Gestione Hotel) di cui però mancano all'appello moltissimi titoli che non sono stati ancora dumpati e che cerchiamo ardentemente, Il Basotter (un espansione del Basic made in Italy pensata per il Plotter Commodore 1520 e distribuita dalla Commodore Italia) (fig. 05) il cui dump è merito di Alberto Teodoro membro stimatissimo del DumpClub64 e della illustre Associazione 8bitinside (<http://www.8bitinside.com>), e infine moltissimi programmi educativi a cui abbiamo dedicato una serie di speciali sulla didattica e la scuola tra cui spicca il n° 4 della collana CompuTeacher (CTO Software) - Tre grandi della Letteratura Italiana - (fig. 6) che ancora non era mai stato dumpato. Proprio riguardo quest'ultima collana vogliamo fare un appello a chiunque fosse in possesso degli altri numeri pubblicati a contattarci oppure a dumpare e condividere con noi e con tutta la comunità del RetroComputing perché questo materiale possa essere preservato, e come si dice in questi casi, tramandato ai posteri!

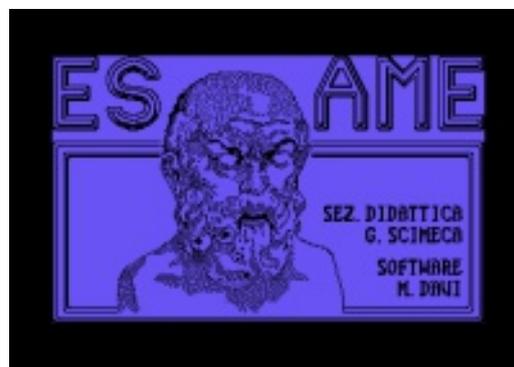


Fig. 6 - Tre grandi della Letteratura Italiana

Ci congediamo infine invitando tutti i lettori di RetroMagazine a visitare la nostra pagina web [www.dumpclub64.it](http://www.dumpclub64.it) in cui troverete uno speciale Halloween dedicato al Commodore 64 visto che la macabra ricorrenza è ormai giunta alle porte.





# RetroMath: Memorie associative e reti di Hopfield

## di Giuseppe Fedele

Una rete neurale (o più formalmente una rete neurale artificiale) è un modello matematico ispirato alla struttura ed agli aspetti funzionali delle reti neurali biologiche. Essa consiste in un gruppo di neuroni artificiali interconnessi tra di loro.

L'uso del termine "rete neurale artificiale" deriva dallo studio del sistema nervoso centrale e dei suoi neuroni, dendriti e sinapsi che costituiscono gli elementi di elaborazione delle reti neurali biologiche. Il primo modello di un neurone fu presentato nel 1943 da W.S. McCulloch e W.H. Pitts [1] e nel 1958 Rosenblatt ideò il perceptrone [2], un semplice classificatore lineare binario e in grado di apprendere efficacemente la regola necessaria per riconoscere due classi di input diverse e linearmente separabili.

Fu John Hopfield nel 1982 che convinse della potenza delle reti neurali presentando un modello di rete capace di creare una memoria associativa in grado di riconoscere configurazioni corrotte e recuperare le informazioni mancanti [3].

### Reti di Hopfield

Una rete di Hopfield consiste in un insieme di neuroni connessi in grado di modificare il loro stato in maniera asincrona (un neurone per volta aggiorna il suo stato interno). Lo stato, ovvero il valore di attivazione è di tipo binario, solitamente  $\{-1,1\}$ . La modifica di tale valore dipende dall'unità stessa e dalle altre unità della rete. Al collegamento tra due unità  $i$  e  $j$  della rete è associato un peso  $w_{ij}$  ed ogni unità ha un valore di soglia che ne determina l'eventuale attivazione.

Il valore di attivazione di un neurone viene calcolato, a tempo discreto, dalla funzione

$$x_i(t+1) = \text{sign} \left( \sum_{j=1}^n w_{ij} x_j(t) - \theta_i \right)$$

dove  $x_i(t)$  è lo stato del neurone  $i$  al tempo  $t$ ,  $\theta_i$  rappresenta il valore di soglia del neurone,

$n$  è il numero di neuroni presenti nella rete (Figura 1).

Figura 1. Struttura di una rete di Hopfield.

Possiamo descrivere l'aggiornamento di tutti i neuroni della rete considerando la precedente equazione in forma matriciale

$$X(t+1) = \text{sign}(WX(t) - T)$$

dove

- $X(t)$  è il valore di attivazione delle  $n$  unità/neuroni all'istante  $t$ :  $X(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}$
- $W$  è la matrice di peso:  $W = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{pmatrix}$  dove  $w_{ij}$  può essere interpretato come l'influenza del neurone  $i$  sul neurone  $j$  (e viceversa)
- $T$  è il vettore contenente i valori soglia per ciascuna unità:  $T = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{pmatrix}$
- la funzione  $\text{sign}(x)$  è definita come 
$$\begin{cases} +1 & \text{se } x > 0 \\ 0 & \text{se } x = 0 \\ -1 & \text{se } x < 0 \end{cases}$$

E' anche possibile rappresentare una rete di Hopfield come un grafo non orientato pesato dove:





- ciascuna unità è un nodo
- l'arco tra due nodi  $i$  e  $j$  è pesato da  $w_{ij}$

**Esempio.**

Nel caso di rete mostrata in Figura 1, si ha

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad W = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix}, \quad T \\ = \begin{pmatrix} -0.5 \\ -0.5 \\ -0.5 \end{pmatrix}$$

Figura 2. Esempio di rete di Hopfield

Solitamente una rete di Hopfield ha la matrice dei pesi  $W$  simmetrica con zeri sulla diagonale principale (non ci sono autoloop, ogni unità non è influenzata da se stessa).

Uno degli utilizzi classici della rete di Hopfield è come CAM (Content Addressable Memory), cioè memorie indirizzabili per contenuto in cui la funzione primaria è di recuperare pattern memorizzati in risposta alla presentazione di un pattern incompleto o rumoroso. Per questa caratteristica questi tipi di rete vengono anche dette error-correcting poiché sono in grado di ricostruire il pattern originale a partire da un pattern che presenta difetti. L'idea dell'utilizzo come CAM consiste nel memorizzare una collezione di patterns chiamati *memorie fondamentali* nella rete scegliendo opportunamente i pesi. Ciascun neurone rappresenta una componente dell'input ed il peso associato al collegamento tra due neuroni misura la correlazione tra le due corrispondenti componenti delle memorie fondamentali. Più il peso è alto più le componenti corrispondenti risulteranno simili tra di loro.

Se indichiamo con  $m$  il numero di patterns  $n$ -dimensionali,  $f_1, f_2, \dots, f_m$  che si vuole memorizzare, è possibile scegliere i pesi della rete come

$$W = \sum_{i=1}^m f_i f_i^T - m \mathbb{I}_n$$

dove  $\mathbb{I}_n$  è la matrice identità  $n \times n$ .

Una volta memorizzati i patterns, i pesi della rete rimangono fissati.

Supponiamo ora di dare in ingresso alla rete un pattern  $g$  con l'obiettivo di individuare il pattern più simile tra quelli memorizzati nella rete. Si tratta di eseguire l'algoritmo descritto qui di seguito:

1. inizializziamo i neuroni della rete come
 
$$x_j(0) = g_j \quad j = 1, \dots, n$$

2. aggiorniamo il vettore di stato della rete  $x(t)$  in maniera asincrona (ovvero un neurone alla volta scelto in maniera casuale) utilizzando la dinamica della rete

$$x_j(t+1) = \text{sign} \left( \sum_{k=1}^n w_{jk} x_k(t) - \theta_j \right) \quad j = 1, \dots, n$$

ripetendo l'aggiornamento fino a quando il vettore  $X$  rimane invariato, ovvero quando  $x_j(t+1) = x_j(t) \quad \forall j = 1, \dots, n$

3. Il vettore  $X$  ottenuto al passo precedente rappresenta allora l'output  $Y$  della rete.



**Esempio.**

Supponiamo che si voglia costruire una rete

per memorizzare i due stati  $f_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$  e  $f_2 =$

$$\begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}.$$

La matrice dei pesi viene allora calcolata come

$$W = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} (1 \ 1 \ 1) + \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} (-1 \ -1 \ -1) - 2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}$$

Questi pesi vengono quindi fissati. Effettuiamo adesso un test utilizzando come input alla rete proprio i patterns che abbiamo utilizzato per calcolare i pesi (ci aspettiamo che l'output della rete coincida con l'input, ovvero che la rete riconosca correttamente il pattern memorizzato). Supponendo nulli i valori soglia delle varie unità, si ha:

$$Y_1 = \text{sign} \left( \begin{pmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$Y_2 = \text{sign} \left( \begin{pmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

Vediamo adesso cosa fornisce in uscita la rete se si da in ingresso un pattern non

memorizzato, ad esempio  $f_3 = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$ ;

notiamo che il pattern è molto simile a  $f_1$  e se non fosse per la prima componente sarebbe proprio uguale; ed infatti l'uscita della rete converge, dopo 3 iterazioni, al pattern memorizzato:

$$\text{sign} \left( \begin{pmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right) = \text{sign} \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\text{sign} \left( \begin{pmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right) = \text{sign} \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

$$\text{sign} \left( \begin{pmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right) = \text{sign} \begin{pmatrix} 4 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Quindi la rete di Hopfield ha corretto l'anomalia sul pattern in ingresso fornendo in uscita il pattern memorizzato più simile.

Si può dimostrare che la capacità di memorizzazione cioè il massimo numero di pattern che la rete può memorizzare è  $\frac{n}{2 \log(n)}$ .

Il codice riportato nel riquadro implementa una semplice applicazione in stile Commodore dove vengono memorizzati due patterns di dimensione 5 ed effettuata una fase di test con un pattern distorto rispetto a quelli memorizzati.

**Bibliografia:**

[1] W.S. McCulloch, W.H. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics, vol. 5 pp. 115–133, 1943.

[2] Rosenblatt, Frank (1957), The Perceptron--a perceiving and recognizing automaton. Report 85-460-1, Cornell Aeronautical Laboratory.

[3] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", Proceedings of the National Academy of Sciences of the USA, vol. 79 no. 8 pp. 2554–2558, April 1982.





### Scheda di approfondimento

#### Convergenza della rete

Il problema che ci si pone è se durante la fase di test, ovvero quando si presenta alla rete un pattern da riconoscere, la rete converge su uno stato memorizzato in un numero finito di iterazioni. Questo è importante per garantire le risposte della memoria associativa in tempi finiti. La dimostrazione della convergenza della rete avviene seguendo i seguenti passi:

- esiste una funzione di energia associata alla rete che diminuisce ogni volta che un neurone cambia stato;
- il numero degli stati che la rete può assumere è finito; ogni neurone può infatti assumere valore  $-1$  o  $+1$ ; la rete è costituita da  $n$  neuroni e quindi  $2^n$  sono i possibili stati della rete;

Questo implica che: l'energia non può decrescere indefinitamente e quindi un neurone non può cambiare stato infinite volte.

Consideriamo come energia associata alla rete la funzione

$$E(t) = -\frac{1}{2}X(t)^T W X(t) - X(t)^T T$$

e supponiamo che ad un certo istante  $t$  solo il  $q$ -mo neurone cambi stato, cioè

$$x_i(t+1) = x_i(t) \quad \forall i \neq q$$

allora è facile dimostrare che

$$\Delta E(t) = E(t+1) - E(t) = -x_q(t+1)^2 + x_q(t)x_q(t+1)$$

Il primo termine  $-x_q(t+1)^2$  è chiaramente negativo; il secondo termine è invece uguale a  $-1$ , infatti poiché il neurone  $q$  ha cambiato stato, potranno verificarsi soltanto i seguenti casi:

1.  $x_q(t) = +1, x_q(t+1) = -1$
2.  $x_q(t) = -1, x_q(t+1) = +1$

In entrambi i casi il prodotto è pari a  $-1$ .

Ne segue che  $\Delta E(t)$  è sempre minore di  $0$ , cioè l'energia della rete è monotona decrescente.





```

10 rem-----
20 rem Rete di Hopfield
30 rem 2019 - Giuseppe Fedele
40 rem-----
50 :
100 rem numero di neuroni e pattern
110 n=5 : m=2 : dim P(n,m)
120 P(1,1)=1 : P(2,1)=1 : P(3,1)=1 : P(4,1)=1 : P(5,1)=1
130 P(1,2)=-1 : P(2,2)=-1 : P(3,2)=-1 : P(4,2)=-1 : P(5,2)=-1
140 :
150 :
160 rem Training
170 dim W(n,n)
180 for i=1 to n
190 : for j=1 to n
200 :   for k=1 to m
210 :     W(i,j)=W(i,j)+P(i,k)*P(j,k)
220 :   next k
230 : next j
240 : W(i,i)=W(i,i)-m
250 next i
260 :
300 rem Test
310 dim X1(n)
320 X1(1)=-1 : X1(2)=1 : X1(3)=-1 : X1(4)=1 : X1(5)=-1
330 for i=1 to n
335 : print X1(i)
338 next i
339 print "-----"
340 dim X2(n)
345 u=1
350 for i=1 to n
360 : X2(i)=0
370 : for j=1 to n
380 :   X2(i)=X2(i)+W(i,j)*X1(j)
390 : next j
400 : if X2(i)>0 then X2(i)=1
410 : if X2(i)<0 then X2(i)=-1
420 : if abs(X2(i)-X1(i))>0 then u=0
430 : X1(i)=X2(i)
440 : print X2(i)
445 next i
450 print "-----"
460 if u=0 then goto 345

```





## Intervista a Vincenzo Scarpa

di David La Monaca (Cercamon)

Fin dall'inizio delle nostre pubblicazioni, abbiamo cercato di esplorare il panorama del retrocomputing italiano che porta avanti attivamente eventi, iniziative e progetti reali. Abbiamo incontrato programmatori, appassionati di retrohardware, fondatori di club retro e vecchie glorie del passato che gentilmente hanno risposto alle nostre domande. Il nostro tentativo di dare corpo alla "scena" italiana prosegue con un'intervista a Vincenzo Scarpa, un vero e proprio punto di riferimento per tutti coloro che praticano con profitto l'installazione di emulatori di computer e console sui loro sistemi Mac, Windows, Linux, ecc. In questa prima parte dell'intervista parleremo soprattutto di EmuWiki, il sito web che raccoglie e archivia centinaia di emulatori software e non solo. Nella seconda parte, Vincenzo ci parlerà dell'altra sua attività sempre legata al mondo dei retrogames: le avventure testuali ed i metodi per svilupparle attraverso delle applicazioni specifiche come Inform e Glulx.

**DLM:** Ciao Vincenzo e grazie per aver accettato di rispondere alle nostre domande. Nell'ambito del retrocomputing e retrogaming italiano godi di una certa notorietà grazie soprattutto alle iniziative che nel corso degli ultimi anni hai portato avanti. Iniziative in particolar modo legate a due tue grandi passioni: gli emulatori di sistemi retro e le avventure testuali. A beneficio dei nostri lettori ti chiedo di tracciare un tuo profilo personale e professionale.

*VS:* Grazie a voi per l'invito. Che dire? Sì, è vero, le avventure testuali e gli emulatori di sistemi retro sono due tra le mie grandi passioni (ce ne sarebbero anche delle altre in realtà, ma il tempo per seguirle tutte purtroppo scarseggia) e sono entrambe inevitabilmente legate alla mia infanzia. Ho iniziato come tanti ragazzini dell'epoca ad arrembiare con il BASIC e, dopo aver usato per un breve periodo il Commodore 64, gli preferii l'MSX, pur sapendo che avrei dovuto rinunciare al sonoro del SID e alla grafica del VIC-II, in favore di un BASIC più avanzato e decisamente più usabile. In quello stesso periodo le avventure testuali andavano piuttosto di moda e io stesso ebbi modo di giocare qualcuna soprattutto sul Commodore 64, grazie alle formidabili storie raccontate dal bravissimo Bonaventura Di Bello. Ma la vera rivoluzione fu per me l'arrivo del Commodore Amiga; era una macchina straordinaria e soprattutto completa. E fu proprio su questo sistema che usai per la prima volta un emulatore, il Transformer, che consentiva all'Amiga di emulare l'odiata piattaforma PC; era il 1990, e in quel periodo stavo frequentando un corso privato di C. L'Amiga era considerato né più né meno che un banale giocattolino, ma quando gli portavo gli esercizi qualcuno si rese conto che qualcosa non quadrava. Erano programmi scritti per il PC ma io il PC non lo possedevo... com'era possibile? Il tempo passava e il fallimento della Commodore mi aveva costretto (purtroppo) a passare proprio al PC. Nel frattempo mi ero diplomato nel 1998 come perito infor-

*matico, ma ben presto mi resi conto che il mio sogno di lavorare nell'informatica era destinato a cadere in frantumi. Il multimedia era ancora agli albori, così come il web design e, viste le mie difficoltà con la matematica, decisi anche di non frequentare l'università d'informatica puntando invece su quella triennale di fisioterapia. Nel frattempo però, il germe della programmazione in me era tutt'altro che estinto e un bel giorno provai a digitare su Google la frase "avventure testuali"... davanti ai miei occhi si spalancò un mondo nuovo, e scoprii anche che esisteva un linguaggio di programmazione chiamato Inform che consentiva addirittura di scriverle (da qui l'idea di scrivere un manuale su Inform e Glulx). Ripresi intanto a giocare alle avventure testuali vecchie e nuove e mi vidi così costretto (si fa per dire) a usare diversi emulatori. Decisi allora di scrivere alcuni manuali d'utilizzo per gli emulatori dello Spectrum, del Commodore 64, dell'MSX e dell'Apple II ma, quando provai a pubblicarli su Wikibooks, come avevo fatto qualche anno prima per il mio manuale su Inform; incontrai diversi ostacoli: la presunta illegalità degli emulatori in sé, i copyright degli screenshot dei giochi e tanto altro ancora. Ma non mi arresi. Il mio sito esisteva già, ma era in html e avevo bisogno di uno strumento più evoluto. Adoravo il wiki di per sé ma non sapevo proprio da dove iniziare; quando stavo ormai per perdere le speranze, mi arrivò in aiuto Roberto Grassi: oltre che a essere un caro amico era anche il webmaster di IF Italia, un sito basato su un motore wiki (chiamato PmWiki) che si rivelò ben presto essere il mio strumento ideale. Nacque così EmuWiki: un wiki, cioè, dedicato agli emulatori (e non solo).*

**DLM:** Tanto per fornire subito un'idea della tua passione e delle motivazioni dietro ai tuoi progetti per il retrocomputing come ti definiresti e come definiresti la tua attività?

*VS:* Mi definisco né più né meno che un hobbista (o un forte appassionato, se volete) e la mia attività è quindi un hobby che porto avanti come posso nei miei ritagli di tempo. Non è quindi (e lo sottolineo) un lavoro ed ecco perché mi rattristo tutte le volte che qualcuno mi chiede come mai non cerco di guadagnare qualcosa da tutto questo: un hobby è forse un lavoro? Ne ho già uno da fisioterapista (grazie al cielo, visti i tempi che corrono); perché mai dovrei per forza cercarmene un altro privandomi così del divertimento?

**DLM:** Sei da sempre un utente "avanzato" di computer per fini ludici o professionali? Qual è stata la scintilla che inizialmente ti ha fatto scoprire il mondo dei computer?

*VS:* Diciamo che sono un utente di computer ben informato, ma non su tutto. Uso il computer per gli scopi più disparati: la programmazione, l'emulazione, l'ascolto della musica su Spotify, la visione di film e serie TV (sia lodato Netflix) e naturalmente i giochi... principalmente





quelli vecchi, però, un po' perché sono un maledetto nostalgico e un po' perché i nuovi giochi richiedono un mucchio di tempo per giocarli. Come afferma scherzosamente qualcuno: "Ho un lavoro, dunque ho i soldi per i videogame... ma non posso giocarci perché ho un lavoro". Per

quanto riguarda invece la scintilla iniziale, sicuramente è stato l'interesse per il BASIC ed i suoi listati (sì, lo ammetto, ero uno di quelli sfigati che passava ore ed ore a digitare listati davanti al computer) e naturalmente anche i giochi in sé: H.E.R.O., Hunchback, Pitstop 2 e molti altri.

**DLM: Ricordi qual è stato il tuo primo contatto con un computer?**

VS: A casa di un amico che aveva il Commodore 64. Era un mio vicino di casa, nonché compagno di classe delle scuole medie, che andavo di tanto in tanto a trovare per poter fare qualche partita a Pitstop 2 e ad Hunchback. Inutile parlare del divertimento che provavo e che provo (grazie agli emulatori) tuttora.

**DLM: Quando hai pensato di occuparti attivamente di emulazione e di sviluppo di un progetto divulgativo e informativo come EmuWiki, divenuto ormai un punto di riferimento in Italia e all'estero per tutti coloro che praticano l'emulazione di vecchi sistemi?**

VS: Non appena ho iniziato a pubblicare su Wikibooks il mio manuale su InfoDLM: Lì ho capito fin da subito le potenzialità che uno strumento come wiki può offrire, una su tutte la possibilità di creare delle pagine web senza doversi più di tanto preoccupare del codice. Se devo essere onesto, all'inizio non pensavo che questo progetto potesse diventare un giorno così esteso, ma si sa, l'appetito vien mangiando. Ad ogni modo, quello che dovevo gestire sono riuscito a gestirlo, e questo per adesso mi basta.

**DLM: Com'è articolato il tuo sito e come funziona la sua gestione? Riesci a fare tutto da solo o ti avvali di qualche collaboratore fidato?**

VS: Il mio sito usa PmWiki, un motore wiki scritto in PHP e utilizzabile anche senza database. Con questo strumento sono nel tempo riuscito a gestire un numero di pagine piuttosto elevato senza doverle per forza editare

tutte, grazie a delle pagine speciali chiamate "header" che applicano le modifiche a tutte le pagine ad esse collegate, cosa questa che mi ha permesso di creare numerose sezioni; EmuWiki infatti comprende, oltre agli emulatori, una biblioteca, degli articoli vari sui vecchi computer e sui vecchi giochi, una sezione dedicata al retrogaming online e molto altro ancora. In genere riesco quasi sempre a fare tutto da solo in quanto, se ben progettato, il wiki stesso ti consente di farlo; sebbene infatti le sezioni siano tante, sono principalmente quelle degli emulatori ad essere costantemente aggiornate (una volta al mese in concomitanza con l'uscita della nuova versione del MAME). In realtà però, il grosso dell'aggiornamento avviene nelle news, in quanto trovo sostanzialmente inutile farlo direttamente nelle pagine tranne nel caso in cui il file lo si debba direttamente scaricare dal sito. A differenza di altri siti d'emulazione, EmuWiki dà la possibilità di scaricare gli emulatori solo se è difficile reperirli altrove; in caso contrario, l'utente può scaricarli accedendo, tramite le pagine stesse del sito, ai rispettivi siti web ufficiali; trovo questo metodo meno dispendioso (in termini di gestione) e soprattutto più onesto nei confronti degli autori degli emulatori stessi. Per quanto riguarda i collaboratori fidati, me ne avvalgo quando in sostanza ho bisogno di qualche consiglio su come gestire un determinato contenuto; sembra tutto banale e scontato, ma quando non si ha la possibilità di lavorare in un certo ambito (mi riferisco ovviamente al web design) molte cose sfuggono o sembrano comunque scontate. Non è affatto così e non finirò mai di ringraziarli per tutto l'aiuto che mi hanno dato finora.

**DLM: Quali sono le fonti per tenere aggiornato un progetto così grande e dove trovi il tempo per occuparti anche della parte divulgativa sui social network principali?**

VS: Beh... in genere ricevo delle segnalazioni varie sulle uscite o meno di certi emulatori, mentre in altri casi vado a verificare io stesso andando direttamente a controllare nei siti web di origine dei diversi software d'emulazione (soprattutto per Android). Quasi sempre però, confronto poi il tutto con le news di EmuCR ed Emu-France, altri due ottimi siti (in particolare il secondo) sull'emulazione. Per quanto riguarda invece la parte divulgativa sui social network mi concentro principalmente su Facebook, in particolar modo sul gruppo del sito "Quelli degli emulatori e del retrò" e sulla pagina; il resto si traduce in una mera condivisione delle news su altri gruppi Facebook (con l'ovvio consenso dei rispettivi admin). A tal proposito, sono stupito dal fatto che queste condivisioni possano dare fastidio a qualcuno: i social network hanno come punto di forza proprio la condivisione dei post e il problema quindi per me non si pone.

**DLM: Nell'ambito dell'emulazione, ti limiti (si fa per dire) a collezionare sul tuo sito solo gli emulatori o anche i simulatori e i programmi di virtualizzazione?**

VS: Li colleziono tutti e tre in quanto, pur essendo diversi tra loro, fanno secondo me sempre parte della stessa





"famiglia". Un emulatore, infatti, è un programma che in sostanza replica il funzionamento di un sistema, permettendo così anche l'esecuzione di tutto (o quasi) il software ad esso correlato. Un simulatore invece, è un programma in grado di replicare il comportamento di un sistema (e quindi solo una sua parte) e si usa in genere per tutte quelle architetture non emulabili, come ad esempio gli scacciapensieri ed alcune calcolatrici scientifiche. I programmi di virtualizzazione consentono infine di creare ed eseguire delle macchine virtuali, ovvero degli ambienti al cui interno è possibile eseguire un sistema operativo diverso (o uguale) da quello installato sulla macchina host vera e propria.

**DLM: Ti sei mai imbattuto in delle problematiche legali nella gestione di EmuWiki? E se sì, quali?**

VS: No, mai, almeno fino ad ora e toccando ferro. La cosa non mi stupisce comunque più di tanto poiché, a differenza di quanto molti ancora pensano, gli emulatori di per sé sono assolutamente legali. Ho affrontato questo tema in un divertentissimo podcast con i Bit-elloni dove, in sostanza, ho specificato che se un emulatore viene distribuito così com'è, senza le rom di sistema coperte da copyright, non viola proprio un bel nulla. Per quanto riguarda la biblioteca, ho messo nella sezione un apposito banner ad hoc sul copyright dei testi e dei loro riferimenti. Per quanto riguarda invece la sezione dei Retrogames Online ho specificato, ogni volta che si clicca sul nome di un gioco, che il gioco in sé risiede su un sito esterno (nel mio caso Vizzed.com) anche se in futuro, non so dire ancora tra quanto, userò invece Internet Archive.

**DLM: Parlando di sistemi hardware reali (non emulatori), quali sono stati i tuoi sistemi preferiti fra quelli di un tempo e perché?**

VS: Beh, tra i sistemi a 8 bit, i miei preferiti in assoluto sono stati lo Spectrum e l'MSX. Il primo, nonostante fosse privo delle porte joystick di serie e degli sprite ed avesse come audio solo un semplice e limitato buzzer, aveva però dalla sua un ottimo BASIC, un prezzo relativamente basso e una buona velocità di calcolo. Il secondo invece era un computer che si basava, primo nel suo genere, sul concetto di standard; in effetti, la sigla MSX (Machines with Software eXchangeability) era uno standard informatico a tutti gli effetti, al quale poi diverse grandi aziende giapponesi come Canon, Sony Toshiba e l'olandese Philips, si erano adattate per introdurre sul mercato le loro macchine. Tra le sue notevoli caratteristiche, vi erano quindi un ottimo BASIC (l'MSX BASIC sviluppato da Microsoft), un discreto chip sonoro (il General Instrument AY-3-8910) che non reggeva però al confronto, questo va detto, con il ben più dotato e flessibile SID del Commodore 64, e una risoluzione grafica per i tempi più che accettabile (256x192 pixel e 64x48 pixel a 16 colori scelti da una tavolozza di 16, con la limitazione però che ogni area orizzontale di 8x1 pixel poteva avere solo 2 colori). Tra i sistemi a 16/32 bit invece, il mio preferito è stato senza ombra di dubbio il Commodore Amiga. Nella sua categoria di computer casalingo, aveva per l'epoca delle capacità multimediali

straordinarie che lo rendevano unico in quanto utilizzava, oltre al processore della serie Motorola 68000, diversi coprocessori dedicati alla gestione della grafica, del suono e delle risorse del sistema. Purtroppo però, sempre storicamente parlando, tutti sembrarono aver compreso le sue straordinarie capacità tranne il management della Commodore stessa, che con le sue scelte suicide portò l'azienda al fallimento nell'aprile del 1994. Le console invece non mi hanno mai attirato più di tanto; ne possiedo solo una, la Nintendo Wii, che utilizzo quasi esclusivamente con il software/gioco Wii Fit.

**DLM: E invece fra gli emulatori, quali sono quelli che preferisci e che usi più spesso?**

VS: Tra i vari emulatori che utilizzo, quelli che in assoluto preferisco sono: 1) MAME (acronimo di Multiple Arcade Machine Emulator) che è stato da sempre il principale emulatore dei giochi arcade che si trovavano ai tempi nei bar e nelle sale giochi. Nel tempo, il MAME ha assorbito il MESS (acronimo di Multi Emulator Super System) diventando così una sorta di vero e proprio "emulatore universale" in grado di emulare, oltre ai già citati giochi arcade, anche i computer e le console dell'epoca; 2) ScummVM, il principale emulatore delle avventure grafiche della Lucas Arts e non solo (indimenticabili Maniac Mansion e Zak McKracken) visto che non riproduce solo il funzionamento dello SCUMM, ma anche quello di altri motori grafici, come ad esempio il Virtual Theatre della Revolution Software o l'AGI della Sierra On-Line; 3) WinUAE, un emulatore in grado di emulare su Windows tutti i modelli di Amiga usciti sul mercato, compresi il Commodore CDTV e il Commodore Amiga CD32. Come se non bastasse, è anche possibile utilizzare, sempre in emulazione, gli Amiga con le schede acceleratrici (provare per credere).

**DLM: Quanto è importante secondo te preservare la memoria di un tempo e di un mondo come quello degli anni '80 e '90 creatosi attorno all'avvento di console e**





**home computer sul mercato nazionale ed internazionale? Gli emulatori riescono a contribuire sensibilmente in questa direzione?**

*VS: Preservare la memoria storica di un sistema che non è più sul mercato da molto tempo è fondamentale proprio per evitarne l'oblio. Gli emulatori contribuiscono sensibilmente in questa direzione poiché riescono ad assolvere appieno a questo nobile (così mi piace definirlo) compito. Va però anche detto che da un paio d'anni a questa parte esistono diversi emulatori che riescono a emulare delle macchine molto recenti, di cui alcune di esse sono addirittura ancora in commercio. Partendo dal presupposto che sono perfettamente legali in quanto in origine non vengono, lo ripeto, distribuiti con le ROM di sistema, non ha molto senso parlare in questo caso di preservazione della memoria storica, ma se non altro si ha comunque la possibilità di provare un dato sistema senza doverlo per forza possedere fisicamente, contribuendo così ad evitarne l'oblio anche se in tempi decisamente più lunghi e di utilizzare i giochi con delle caratteristiche che nelle macchine originali non sono presenti come ad esempio, col Dolphin, i giochi del Nintendo GameCube e della Nintendo Wii in Full HD.*

**DLM: Cosa pensi dell'emulazione hardware fornita dalla tecnologia FPGA? Credi che rappresenti, come molti affermano, il futuro del retrocomputing/retrogaming?**

*VS: Quando si parla di questa tecnologia, più che di "emulazione hardware" userei piuttosto il termine "riproduzione hardware" in quanto un dispositivo FPGA (acronimo di Field Programmable Gate Array) è formato da un circuito integrato le cui funzionalità sono programmabili tramite un linguaggio di descrizione hardware. Andando più nel dettaglio, Roberto Lari, in un suo articolo pubblicato sul numero 13 della vostra stessa rivista, afferma che all'interno del MIST e del MISTICA, due dei vari dispositivi basati su questa tecnologia che si possono trovare sul mercato, l'FPGA è un circuito integrato che all'accensione è sostanzialmente "vuoto", ma se opportunamente programmato può contenere le informazioni hardware e software relative alla struttura di vari computer e console. I nuclei elaborativi che "pilotano" l'FPGA vengono definiti "core" il cui caricamento avviene attraverso una normalissima scheda SD o MicroSD con adattatore e la loro preparazione è opera di esperti tecnici e ingegneri che sfruttano le potenzialità di due diversi linguaggi di programmazione appositamente creati per le schede FPGA: Verilog e VHDL. Quello che in sostanza occorre capire è che, mentre un emulatore fa in modo che un processore esegua a turno le istruzioni di ogni singolo componente di una macchina (a rotazione e a velocità molto elevata), l'FPGA fa invece sì che tutti i sistemi emulati lavorino simultaneamente, ed è per questo motivo che risulta essere più fedele all'originale rispetto a un emulatore. C'è anche però chi sostiene che, a mano a mano che la potenza delle CPU dei PC usati come host per eseguire gli emulatori aumenterà, le differenze si noteranno sempre di meno e il lag diminuirà a tal punto da non essere più "avvertito" dall'utente e ci si accorgerà di essere su un'emulazione*

*software solo in caso di desync audio/video. Il tempo ce lo dirà; ad ogni modo, ora come ora la tecnologia FPGA è quella che, se ben programmata, più in assoluto si avvicina al comportamento dell'hardware originale al punto da diventare indistinguibile. Una testimonianza di quanto sto dicendo potete trovarla sul sito Retroacademy. Concludendo, i sistemi FPGA rappresentano sicuramente il futuro del retrocomputing/retrogaming, ma solo per gli appassionati più esigenti; gli altri dovranno continuare ad accontentarsi (si fa per dire) degli emulatori.*

**DLM: Di recente anche i sistemi di emulazione software basati sui Raspberry Pi in versione Bare Metal hanno riscosso un qualche successo. Ritieni che avranno un futuro a lunga gittata e che molti dei sistemi a 8/16-bit seguiranno i pionieri BMC64/128/VIC-20 e BMZX?**

*VS: Gli emulatori Bare Metal programmati da Randy Rossi hanno la particolarità di accedere direttamente alle risorse dei Raspberry Pi 2 e 3 senza appoggiarsi al kernel Linux. In pratica, come afferma Roberto Lari in un articolo pubblicato sul numero 15 della vostra rivista, sono gli emulatori stessi che fungono da sistema tra l'utente e l'hardware (grazie anche ad un set di librerie chiamate Circle), andando così a risparmiare delle preziose risorse come la RAM e i cicli della CPU e facendo sì che perfino un non troppo potente Raspberry 2 (con cpu da 900mhz dual core) sia più che adeguato per garantire un'emulazione accurata dei vari sistemi supportati. Tuttavia, essendo il Bare Metal un sistema senza Linux, non è in grado di riconoscere le periferiche USB di nessun tipo se collegate a sistema già avviato, e occorre quindi averle già connesse prima di accendere il Raspberry oppure si è costretti a spegnere e riaccendere per averle attive e funzionanti. A parte questo, sembra proprio che non avranno un futuro a lunga gittata; la situazione attuale vede infatti il solo Randy Rossi portare avanti i suoi progetti e, una volta che sarà esaurito il parco macchine a 8 bit del VICE, non uscirà probabilmente più nulla di nuovo. Spero comunque di sbagliarmi.*

**DLM: Cosa pensi dell'attuale panorama dei sistemi retro? Il relativo ritorno di popolarità di questi ultimi anni porterà, secondo la tua opinione, ad un incremento dell'offerta a tutti i livelli (nuovo hardware, sistemi FPGA, emulatori software classici e BM, nuove produzioni software e di videogame)?**

*VS: Beh, il ritorno di popolarità di questi ultimi anni dei sistemi retro sta già portando ad un incremento dell'offerta a tutti i livelli: basti pensare agli emulatori vari, ai sistemi FPGA ma anche ai Nintendo Classic Mini e alle nuove produzioni di videogame sul Commodore 64, sullo Spectrum e non solo. Siamo sempre però in un contesto di nicchia (piuttosto grossa devo dire, ma sempre ristretta ad un certo numero di appassionati) che, almeno per quanto riguarda il retrogaming, è probabilmente destinata negli anni a perdurare, in quanto in un modo o nell'altro i vecchi giochi vengono sempre riproposti (basti pensare ai telefonini), mentre invece sul re-*





*trocomputing credo che questo grande interesse durerà solo finché ci sarà qualcuno che quelle macchine le ricorderà avendole in passato già possedute. Dopodiché verranno secondo me un po' messe da parte, anche se non scompariranno mai del tutto grazie alla presenza degli emulatori e dei vari musei e/o collezioni varie sparse in giro per il pianeta.*

**DLM: Ci sarà ancora un futuro secondo te per l'emulazione, alla luce della sempre maggiore importanza dello streaming e della presenza sul mercato di piattaforme di cloud gaming come Google Stadia?**

*VS: Lo streaming è necessario principalmente per due motivi: la riduzione dei costi di produzione e soprattutto la riduzione dei tempi di uscita dei nuovi giochi e/o aggiornamenti vari. Ora, se da una parte Google punta solo ed esclusivamente al cloud gaming, non è detto che debbano per forza farlo anche la Nintendo, la Sony e la Microsoft. Ovviamente anche loro sono giocoforza costrette a utilizzare lo streaming e il cloud gaming, ma non credo proprio che i videogiochi scompariranno del tutto dagli scaffali dei negozi. Di conseguenza, anche l'emulazione continuerà secondo me ad avere un suo futuro, anche se con un parco software decisamente più ridotto.*

Ringraziamo Vincenzo per aver risposto ai nostri quesiti e rimandiamo i lettori alla seconda parte dell'intervista, che sarà pubblicata in uno dei prossimi numeri di RM, in cui tratteremo il sempre affascinante mondo delle avventure testuali dal punto di vista dello sviluppatore e dello "sceneggiatore". Come detto in precedenza, Vincenzo ha scritto un manuale d'uso pratico di Inform, un vero e proprio ambiente di sviluppo per le AT, e da tempo è un esperto di Glulx, una macchina virtuale portatile che permette di scrivere, produrre e giocare le IF (Interactive Fiction).

## RIFERIMENTI WEB

**EmuWiki**  
<https://www.vincenzoscarpa.it/emuwiki>  
<https://www.facebook.com/EmuWiki>

**Biblioteca EmuWiki**  
<https://www.vincenzoscarpa.it/emuwiki/pmwiki/pmwiki.php?n=Biblioteca.Biblioteca>

**EmuWiki RetroGame online**  
<https://www.vincenzoscarpa.it/emuwiki/pmwiki/pmwiki.php?n=OnlineGames.OnlineGames>

**PmWiki**  
<https://www.pmwiki.org/>

**EmuCR**  
<http://www.emucr.com/>

**Emu-France**  
<http://www.emu-france.com/>

**Quelli degli emulatori e del retrò**  
<https://www.facebook.com/groups/quellidegliemulatori/>

**Emuparadisi fiscali**  
<https://ibitelloni.com/2018/11/06/una-trasmissione-di-nerda-ep-8-emuparadisi-fiscali/>

**Vizzed.com**  
<https://www.vizzed.com/>

**Internet Archive**  
<https://archive.org/>

**MAME**  
<https://www.mamedev.org/>

**ScummVM**  
<https://www.scummvm.org/>

**WinUAE**  
<http://www.winuae.net/>

**Articolo di Retroacademy sull'FPGA**  
<https://www.retroacademy.it/2016/06/16/retrocomputers/amiga-mist-esperienza-utilizzo-opinioni/2578/>





## M4 Interface per Amstrad CPC

di Andrea Pastore

Che cos'è la M4? Molti di voi se lo staranno sicuramente chiedendo, soprattutto dal momento che si sono avvicinati da poco (vuoi per passione o per una semplice curiosità), al fantastico mondo a 8-bit offerto dalla linea di computer Amstrad CPC, home computer che uscirono nella nostra bella Italia a cavallo tra il 1985 e il 1986. Per chi non conosce questi computer, provo a riassumere in breve la loro storia.



### Storia breve di Amstrad e del CPC

Amstrad (acronimo di Alan Michael Sugar Trading) fu fondata da Alan Michael Sugar nel 1968, ed è una società tutt'ora esistente con sede a Brentwood nel Regno Unito. Nella visione iniziale di Sugar, Amstrad doveva essere un'azienda specializzata nel commercio di piccoli prodotti e nulla di più, ma fin dai primissimi mesi di vita (e visti i guadagni nettamente superiori), la politica aziendale cambiò rapidamente direzione, focalizzandosi verso l'elettronica di consumo. Inizialmente il compito era quello di comprare stock di prodotti a bassissimo costo (come ad esempio radio, televisori, impianti stereo e amplificatori), rimarchiarli come Amstrad e rivenderli ad un prezzo maggiorato, ma che fosse sempre vantaggioso per il pubblico. I guadagni da questo tipo di commercio furono tali, che in brevissimo tempo Amstrad ebbe la possibilità di creare i suoi prodotti direttamente in fabbrica.

Nel 1980 Amstrad divenne un marchio affermato; nel Regno Unito non esisteva famiglia senza un prodotto Amstrad in casa e la società venne perfino quotata alla Borsa di Londra. Nello stesso anno il valore societario raddoppiò e nel corso dei primi anni '80 l'utile e il valore di mercato della società salì vertiginosamente ogni anno. Nulla poteva fermare la sua ascesa ormai.

Nel 1984, riconoscendo l'opportunità offerta dai computer casalinghi, la casa britannica lanciò una sua macchina a 8-bit, basata sul processore Zilog Z80, l'Amstrad CPC 464. Anche se la gamma dei CPC (acronimo di Colour Personal Computer) presentava delle macchine interessanti (avevano il supporto per il CP/M ed erano dotate di un buon interprete BASIC), si trovò a competere con rivali molto agguerriti come il tecnicamente e più potente Commodore 64, il popolare Sin-

clair ZX Spectrum, per non parlare poi del massiccio e didattico BBC Micro. Nonostante questo, tre milioni di unità vennero vendute in tutto il mondo, con una vita produttiva piuttosto lunga. Ben otto gli anni di permanenza sul mercato e il lancio di altri due modelli di CPC che costituirono l'evoluzione diretta del primo leggendario CPC 464. I modelli in questione furono il CPC 664 (che aggiungeva un disk drive interno), il CPC 6128 (che aggiungeva 64 kb in più oltre al disk drive interno), senza contare l'edizione speciale per il mercato spagnolo, il CPC 472, che altro non era che lo stesso modello base, con una RAM aggiuntiva per aggirare una legge locale riguardante la quantità di memoria presente nei computer di quel periodo.



### La linea Plus

L'onda sulla cui cresta Amstrad si posò per così tanto tempo ebbe una brusca battuta di arresto agli inizi degli anni '90. Pensando che il mercato fosse ormai maturo per una naturale evoluzione della linea di computer CPC, vennero create tre nuove macchine dalle caratteristiche (secondo la casa madre) sbalorditive. In realtà la nuova linea Plus proponeva di nuovo al pubblico delle macchine a 8-bit (basate sempre sul processore Zilog Z80), con caratteristiche avanzate come più colori su schermo, sprite e scrolling hardware. Le macchine prodotte furono il 464 Plus, il 6128 Plus e la console GX4000. Il risultato delle vendite di questi prodotti fu a dir poco catastrofico. Nel 1990 una politica basata sui prodotti al risparmio e tecnicamente inferiori alla media di quel periodo non raccolse nessun consenso di pubblico e i Plus vennero ritirati dal mercato già alla fine dello stesso anno. Gli ultimi modelli rimasero nei negozi (ad esaurimento scorte) fino ai primi mesi del 1991.



Dopo solo due anni, nel 1993, Amstrad abbandonò definitivamente l'informatica ed iniziò la sua nuova avventura in un settore emergente, quello delle telecomunicazioni, producendo PDA, dispositivi portatili innovativi con telefonia e posta

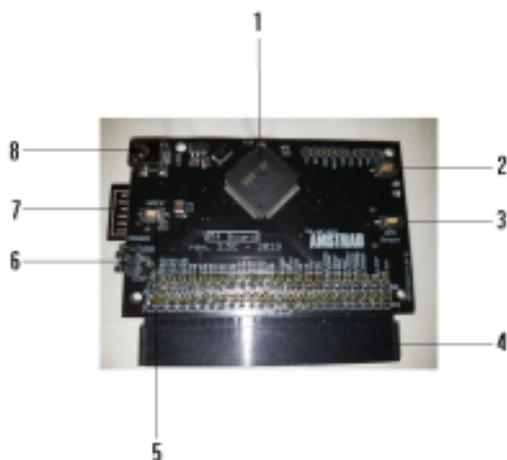




elettronica integrata, che ottennero un buon successo di pubblico grazie alla crescita del nuovo settore di mercato, restando in produzione fino agli inizi del decennio successivo.

Oggi il marchio Amstrad è legato principalmente ai decoder per la ricezione del digitale terrestre e quelli satellitari. Sul sito ufficiale sono illustrati i prodotti attuali, tra cui il noto DRX700i HD, prodotto per Sky Italia e vi si trova riassunta anche la storia della casa in cui sono citati gli anni d'oro del CPC464, ma nulla su esperimenti fallimentari come la linea Plus o la console GX4000. Di fatto si enfatizza di più il periodo delle telecomunicazioni, che del resto è il suo business attuale.

### La scheda M4



Quello che rimane a noi come eredità è una macchina che sicuramente non ha avuto la stessa diffusione dei blasonati Commodore 64 o Sinclair ZX Spectrum, ma che è riuscita ugualmente a ritagliarsi un posticino d'onore nella storia degli home computer a 8-bit. Possedere oggi una macchina del genere, ma completamente spoglia del software che la contraddistinse all'epoca, non ci permette di poterla apprezzare completamente ed è un peccato poichè appassionati da tutto il mondo (soprattutto francesi, spagnoli e tedeschi) producono costantemente programmi e giochi, ancora oggi. Ed è proprio su questo fronte che arriva in nostro soccorso la scheda d'interfaccia M4, un prodotto costruito in Germania dal 2016, che si propone come la scheda "flash" definitiva per Amstrad CPC, ma procediamo con ordine.

Ovviamente il termine "definitivo" è decisamente azzardato, in quanto al momento ci sono schede più performanti adattabili al CPC, come ad esempio il Gotek, ma di sicuro la M4 rappresenta la soluzione più pratica. In primis, per il Gotek abbiamo bisogno necessariamente di un CPC 6128 (o CPC 664) con un disk drive interno, mentre la M4 è adattabile a tutti i modelli di CPC e Schneider tramite opportuno adattatore. Il Gotek necessita dello smontaggio del drive interno per poterlo collegare al suo posto, mentre la M4 si collega direttamente alla porta di espansione che si trova dietro la tastiera. Ma vediamo nel dettaglio cosa offre la M4, attraverso lo schema seguente:

- 1) Un chip ARM, il cuore della scheda
- 2) Hard Reset (il reset della scheda M4)
- 3) Soft Reset (il reset del CPC, mantiene in memoria l'ultima directory visitata)
- 4) Connettore Edge per il collegamento alla porta di espansione
- 5) Tasto Hack (serve per salvare lo stato del sistema e riprendere il programma o il gioco dove lo avevate lasciato)
- 6) Uscita di corrente (impostata a 5 volt di default ed in realtà non ho capito bene a cosa possa servire)
- 7) Il modulo per il collegamento ad una rete Wi-Fi
- 8) La presa per l'alimentazione esterna
- 9) Uno slot per le schede microSD

Non scenderò in dettagli tecnici, in quanto me ne intendo decisamente poco e non vi parlerò di quello che effettivamente potrete fare con la M4 collegandola in rete, sul web ci sono diversi filmati esplicativi su Youtube e tutorial che illustrano minuziosamente come utilizzarla in tal senso, ma mi limiterò a spiegarvi come la utilizzo io, ossia in maniera tale che, anche se non si dispone di una connessione ad Internet, avrete la vostra bella scheda M4 già pronta da posizionare dietro la porta di espansione del CPC, accenderlo e iniziare a giocare.

Iniziamo. Per prima cosa dovete avere una M4 e per poterne acquistare una basta inviare una email a Duke (al momento ha prodotto circa 700 unità ed è ben disponibile a produrne altre) attraverso il sito <http://www.spinpoint.org>, che tra le altre cose contiene anche tutte le informazioni tecniche di cui avrete bisogno ed i tutorial per collegarla in rete. Il secondo acquisto necessario da fare è una microSD, da collocare nell'apposito alloggiamento sulla M4 (vedi slot 9 nella foto precedente) ed un PC Windows. Bene, ora siete pronti per ballare! Collegate la M4 nella porta di espansione, accendete il CPC e la scheda creerà in automatico i file di boot che faranno apparire la schermata iniziale simile a quella



della pagina seguente.

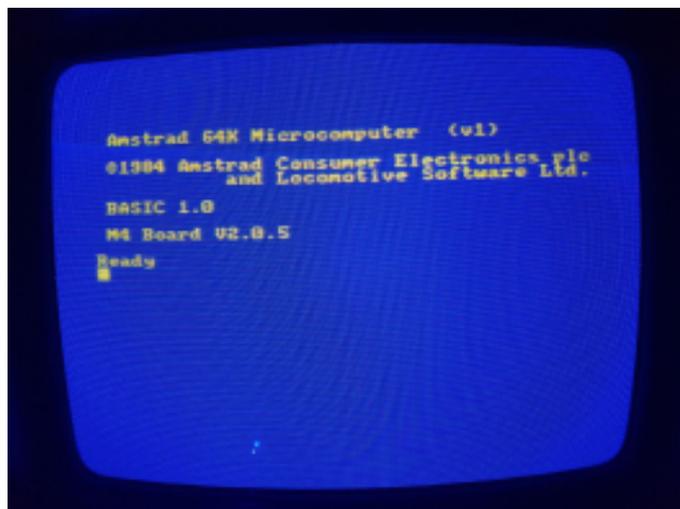
La scritta sotto al BASIC 1.0 significa che la scheda M4 utilizza la release del firmware V2.0.5. Ora spegnete tutto e andiamo a procurarci l'agognato frontend. Quello che utilizzo io è una vecchia versione di un frontend (l'M4 Explorer) che andrebbe aggiornato, in quanto la release è rimasta ferma alla versione 1.0 e purtroppo ad ogni rilascio di un nuovo firmware M4, lo stesso ri-





sulta sempre meno veloce. Ad ogni modo il sito per scaricare il frontend ed i file da inserire nella root della SD card li trovate agli indirizzi indicati nel riquadro dei riferimenti a fine articolo.

Dopo aver inserito questi tre file nella root, create un'ulteriore cartellina e rinominatela Games (o come preferite) ed inserite al suo interno i file .dsk dei vostri giochi preferiti. Staccate la sd dal PC, inseritela nella M4 e riaccendete il CPC. Ora battete RUN"M4 e il frontend si caricherà, mostrandovi una schermata come questa:



I tasti freccia servono per navigare, ENTER per entrare nella cartella (e all'interno dei file dsk), con il tasto U ritornerete nella directory precedente e con il tasto R caricherete i vostri giochi, niente di più semplice, buon divertimento!

Il mio consiglio è quello di provare più versioni dump di ogni gioco, in quanto uno dei difetti della M4 è proprio la compatibilità con il software stesso e non tutto quello che inserite nella scheda in formato .dsk purtroppo viene riconosciuto e caricato. Dopo qualche mese di lavoro sono riuscito a creare una mia scheda personale con circa 2000 giochi funzionanti, ma fuori ne sono rimasti alcuni che non ne vogliono proprio sapere di funzionare. Se volete la piena compatibilità, allora i vostri occhi devono posarsi necessariamente sul Gotek, siete avvisati!

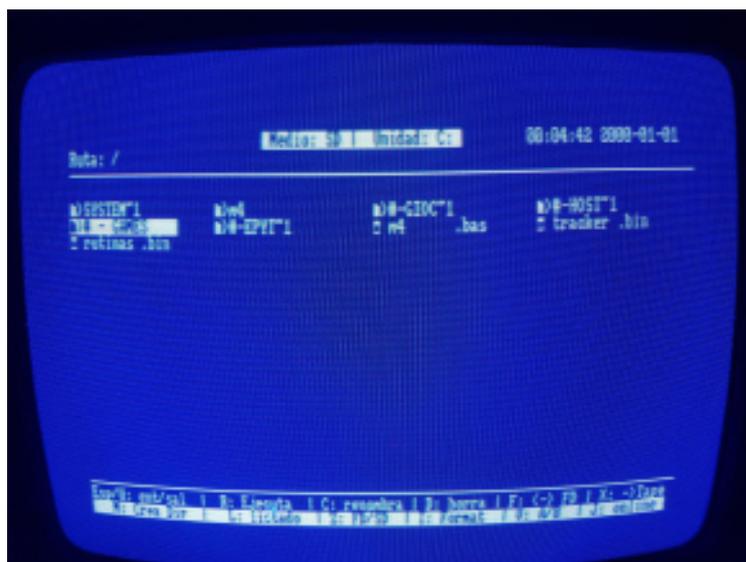
## RIFERIMENTI WEB

**Dukem produttore della M4:**  
<http://www.spinpoint.org>

**M4 Explorer:**  
<http://rafa32gr.westeurope.cloudapp.azure.com/wordpress/m4-explorer/>

**Integrazione di un Gotek per CPC 6128:**  
<https://www.youtube.com/watch?v=6HCQcTmm64>

**Giochi per Amstrad CPC:**  
<https://cpcrulez.fr/games.htm>





# Sedai Linux

di Giorgio Balestrieri

In questo articolo di metà autunno vogliamo parlarvi ancora una volta di un progetto basato su Raspberry PI, capostipite e portabandiera di tutti i sistemi "SoC per le masse" nati ad inizio decennio e animati dal sistema operativo Linux. La versatilità e le caratteristiche di questo microcomputer hanno stimolato lo sviluppo di innumerevoli progetti, dai più semplici fino a distribuzioni complete, dedicate a compiti ben determinati come la creazione di server domestici o da ufficio, monitoraggio meteo, videosorveglianza, centralini telefonici VoIP o l'emulazione di console e home computer retro. Tra questi ultimi troviamo distribuzioni altamente specializzate, come Amiberry e BMC64, che trasformano il Raspberry PI rispettivamente in un'Amiga ed in un Commodore 64, o più generaliste come Lakka, Batocera e naturalmente la notissima RetroPie, per cui abbiamo presentato un pacchetto di utility nel numero 11 di RetroMagazine, che hanno l'obiettivo di includere ed uniformare la gestione di quanti più emulatori possibili, trasformando il sistema in una vera piattaforma per retrogaming, in grado di riprodurre l'esperienza di decine di console e microcomputer e di far girare porting nativi per Linux di giochi come Doom, Quake o Duke Nukem. In quest'ultima categoria si inserisce, o almeno così sembrerebbe a prima vista, Sedai Linux, una distribuzione creata da Maurizio Cammalleri e specializzata nell'esecuzione degli home computer 8bit, tanto popolari negli anni '80 e '90, su cui moltissimi di noi all'epoca adolescenti hanno passato ore felici, imparando a programmare ed a giocare. In realtà Sedai Linux è molto di più di questo, come scoprirete nell'intervista fatta all'autore che ci racconterà come è nata questa distribuzione e quali obiettivi si è prefissato per il futuro di questo progetto.

**GB: Come da rito, iniziamo dalle presentazioni: cosa fai in questa fase della vita e come sei approdato alla passione per la retroinformatica?**

*MC: Sono uno sviluppatore di software indipendente. Opero prevalentemente in Veneto, ove mi sono trasferito da qualche mese dalla Sicilia. Mi occupo principalmente di controlli di processo, ma ogni tanto non disdegno incursioni nel settore aerospaziale. Sto sviluppando un EFIS (il cruscotto "virtuale" che mostra la strumentazione di bordo su un monitor NdR) per l'aviazione civile.*



*Proprio da questo progetto ho derivato il progetto Sedai Linux, un omaggio all'home computer, sul quale è cresciuta la mia passione per la programmazione (il Commodore 64 ndr).*

**GB: Che sistemi (console, microcomputer) hai usato nel periodo '80-'90? Quale hai apprezzato di più?**

*MC: Il primo computer su cui ho messo mano è stato il VIC-20. Mi ha esaltato perché è stato il primo, ma il vero amore è stato per il Commodore 64. Ho avuto modo di conoscere anche l'Amiga 2000, ma non ne vedevo sbocchi professionali, quindi ho deciso di puntare sui sistemi MS-DOS. Console, ad eccezione di un clone di Pong, non ne ho mai usate né possedute. Ho sempre preferito i coin-op.*

**GB: Sistemi SoC: quali hai utilizzato e perché hai scelto i microcomputer della serie Raspberry PI?**

*MC: Ne ho provati tanti, tra i quali anche l'Olinuxino A64 sul quale si sta tentando il porting di Sedai. Il punto è che certe architetture sono poco documentate, in particolare le GPU (Mali400 nel caso dell'Olinuxino), non sono disponibili driver in formato sorgente oppure, se disponibili, gli stessi sono forniti in formato binario, il che richiede di applicare patch al kernel mainline. Proprio per questo è stato scelto il progetto Raspberry Pi, perché è il più aperto tra le varie architetture chiuse. Ma il sogno è quello di un'architettura totalmente aperta.*

**GB: Sedai Linux: quando e come ti è venuta questa folle idea, in cosa differisce dalle altre distribuzioni orientate alla retroemulazione ed a quali obiettivi hai puntato?**

*MC: Credo che, raggiunta una certa età, le nostre nostalgie prendano il sopravvento. Il nome Sedai non è stato scelto a caso: in giapponese significa "generazione". Una sorta di passaggio di testimone. Qualcuno lo chiama "scimmie". Ebbene, questa "scimmia" si è svegliata mentre lavoravo al disegno dell'EFIS. Mi sembrava di essere tornato ai tempi del Simons' Basic. Il passo verso Sedai è stato breve: mi è bastato togliere la patch per il realtime ed applicare altre piccole modifiche, compilare ed installare i vari emulatori, et voilà il sistema*





era servito. Ovviamente il primo degli emulatori è stato Vice. Nel creare Sedai non ho badato a cosa facessero le altre distribuzioni. Ho solo pensato che un sistema snello avrebbe aiutato ad eseguire meglio gli emulatori che, come è noto, possono essere molto avidi di risorse. Per esempio, Vice gira su un unico processo a singolo thread, per rispettare in maniera stringente i timing delle componenti hardware emulate. In particolare, la libreria ReSID mette sotto torchio la CPU che neanche tutto il resto dell'hardware emulato. Molti fanno overclock per raggiungere prestazioni decenti, io preferisco rinunciare all'estrema accuratezza e puntare ad un sistema snello. Infine, Sedai non vuole essere solo una piattaforma di emulazione, ma un sistema per riscoprire lo spirito che ha caratterizzato l'epoca dell'home computer.

**GB: Come hai creato il sistema? Già dal primo avvio è evidente che non hai utilizzato una distribuzione Linux standard, che tool hai utilizzato per creare la tua distribuzione?**

*MC: Il tutto è partito da Buildroot, un insieme di script e di toolchain per la cross-compilazione mirante alla creazione di ambienti Linux per sistemi embedded. Che poi era quello che mi serviva per il mio EFIS. Al Buildroot originale sono state aggiunti altri script, modificati alcuni file di configurazione e perfino aggiunti pacchetti. Ma non si vuole creare un fork di Buildroot, quindi si sta cercando di rientrare nell'alveo del progetto originale, rilasciando script, patch e pacchetti vari come opzioni da aggiungere.*

**GB: Quanti sistemi sono inclusi attualmente in Sedai? Quanti conti di includerne prima di ritenerti soddisfatto?**

*MC: In stretto ordine alfabetico, Sedai include: Amiberry, Atari800, BasiliskII, DOSBox, Fuse, Hatari, openMSX, Vice, ZEsarUX. Quindi include quasi tutti i sistemi Commodore, Atari, Sinclair e Amstrad e, ultimi ma non meno importanti, tantissimi modelli MSX e MSX2. Ciliegina sulla torta, sono emulati i Macintosh fino ai modelli Quadra e generici sistemi MS-DOS. Su quest'ultimi girano perfettamente dalla classica accoppiata MS-DOS e Windows 3.11 fino a Windows 95. In verità mi sarei voluto fermare ai sistemi 8-bit, ma ho avuto pressanti richieste per sistemi più evoluti. Devo dire che girano tutti abbastanza bene, ma mantenere tutti questi pacchetti non è semplice. Quindi non credo che ne aggiungerò altri. Anzi, se non riuscirò ad affidare a qualcuno il mantenimen-*

*to degli emulatori dei sistemi 16/32 bit, è possibile che deciderò di rimuoverli. Su richiesta di qualcuno, ma anche per mia curiosità, ho provato qualche emulatore di console. Devo dire che anche in questo caso i risultati sono promettenti. Ma il punto è sempre lo stesso: problemi di licenze a parte, non è possibile sprecare risorse su mille cose. Finisci per non farne bene alcuna.*

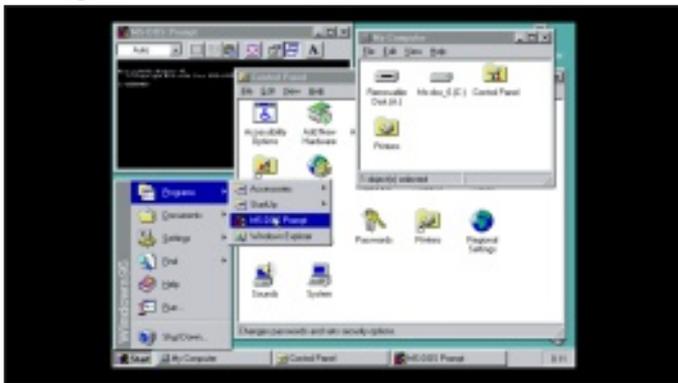
**GB: Nella creazione della tua distribuzione "custom made" si nota che hai sacrificato molte funzionalità standard a favore della leggerezza e della brevità del tempo di avvio (Sedai è pronta all'uso in pochi secondi, letteralmente). A cosa hai rinunciato e perché?**

*MC: Ve la immaginate un'avionica (l'insieme dei dispositivi elettronici installati su un aeromobile e preposti al volo NdR) che va in crash mentre si è in volo e richiede preziosissimo tempo per essere ripristinata? Ecco i veri obiettivi: il sistema doveva essere stabile e, in caso di problemi, doveva ritornare operativo nel più breve tempo possibile. In Sedai sono state rimosse tutte quelle patch pensate per utilizzi mission critical e real time, perché sono semplicemente inutili se non controproducenti per gli emulatori. Sedai non è un sistema Linux come Debian e similari. Non ambisce ad alcun ruolo come server. Ci sarà OpenSSH, ma solo per permetterne la gestione remota. Non vuole essere un sistema desktop, infatti non è presente né XOrg né Wayland. La shell di default è BusyBox, ma è presente il set minimale per Bash. A parte quanto basta per gestire i filesystem Ext, FAT e NTFS, non ci sono altre utilità di rilievo. Allo stato attuale ci sono alcune librerie extra che stiamo utilizzando per alcuni test, ma molte di esse non saranno presenti nella prima versione stabile che verrà rilasciata. Anche il kernel sarà ridotto rispetto a quello attuale, perché verranno eliminati tutti i moduli ritenuti non necessari, mentre quelli fondamentali saranno compilati nel kernel stesso, per evitare inutili overhead all'avvio. Infine, non è nemmeno previsto una qualche forma di gestione dei pacchetti software aggiuntivi. Il sistema è volutamente minimale, una sorta di home computer in chiave moderna.*

**GB: Sedai è dunque una distribuzione orientata principalmente al retrocomputing che mira a risvegliare l'originale spirito degli anni '80 e '90. Un progetto meritevole ma anche ambizioso e di non piccole dimensioni; quanto tempo riesci a dedicargli e quanti ti aiutano in questo compito?**

*MC: Meno di quanto vorrei, perché l'attività professiona-*





le assorbe la maggior parte del tempo. Intorno a Sedai ci sono diverse persone che collaborano in un modo o in un altro, ma non mi dispiacerebbe avere qualcuno che mi desse man forte, in particolare nell'ottimizzazione degli emulatori e nello sviluppo di Sedai Shell (la GUI di Sedai, che non richiede uso di mouse) e Sedai Basic.

**GB:** Come sviluppatore trovo Sedai Basic un argomento particolarmente interessante e sono ragionevolmente sicuro che stuzzicherà anche la curiosità di molti lettori. Puoi dirci di più?

**MC:** Questo è un progetto abbastanza ambizioso. Sedai Basic è in piena fase di sviluppo. L'interprete funziona, ma la shell interattiva è ancora acerba. Il progetto prevedeva l'aderenza agli standard ANSI X3.60 e X3.113, la cui documentazione purtroppo è a pagamento. Poiché open è sicuramente più bello, i due standard saranno sostituiti rispettivamente dagli ECMA-55 e ECMA-116. In più, il Sedai Basic sarà polimorfico, ci saranno dei plugin per rendere l'interprete compatibile con i vari dialetti dei vecchi home computer. Arriveremo anche ad emulare molte delle locazioni di memoria originali. Sarà bello veder funzionare le istruzioni PEEK e POKE, ma è chiaro che sarà difficile se non impossibile rispettare i timing delle macchine originali. Ma le istruzioni basilari, anche per il disegno in modalità hires, saranno sicuramente disponibili. L'ultimo tassello del progetto Sedai Basic sarà un traduttore da Basic a Free Pascal, così da poter poi compilare ed ottenere binari più performanti.

**GB:** In futuro credi possa essere interessante includere anche software per lo sviluppo sui sistemi emulati, sia eseguibili sulle piattaforme originali sia in formato "cross-development"?

**MC:** Qualcosa c'è già nella corrente versione di Sedai: il cc65 è un compilatore C per architetture 65xx, che può compilare per molti sistemi 8 bit. Con esso è persino possibile scrivere applicazioni GEOS per C64, C128 ed Apple //e. Per sistemi 16, 32 e 64 bit verrà incluso il Free Pascal, che ritengo essere ancora oggi un linguaggio valido sia sul piano didattico sia produttivo. In più, Free Pascal compila anche per architetture Motorola 68k, quindi Amiga e Macintosh. Per i sistemi Z80 è in corso la valutazione di diverse soluzioni.

**GB:** Cosa ne pensi dei sistemi baremetal? Negli ultimi tempi essi hanno guadagnato popolarità grazie ai vantaggi derivati dall'utilizzo in esclusiva dell'hardware, ma Sedai Linux è in grado di offrire ottime pre-

stazioni, dal boot completato in pochi secondi fino ad una interferenza minima del sistema operativo nei compiti di emulazione. Credi ci sia un reale vantaggio nell'usare un sistema baremetal e privarsi del tutto del supporto di un sistema operativo?

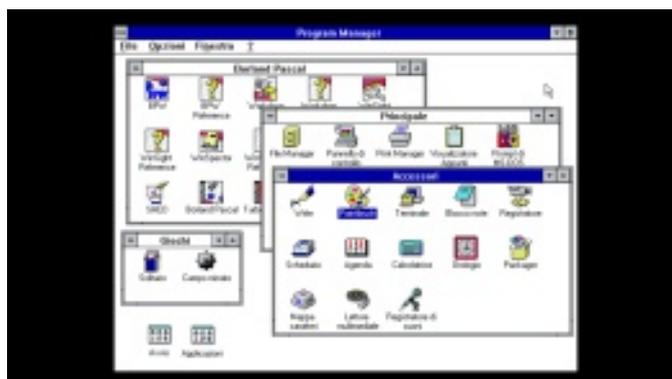
**MC:** Il baremetal è un bell'esercizio di stile, ma non migliora in maniera significativa le prestazioni. Per di più, comporta notevoli limitazioni d'uso, ben oltre quelle ammissibili perfino per un sistema minimale. Un kernel Linux perfettamente bilanciato e la giusta priorità ai processi garantiscono eguali prestazioni, ma maggiore flessibilità e la possibilità, come nel caso di Sedai, di eseguire numerosi altri sistemi.

**GB:** Qual è la tua opinione sui sistemi FPGA? Ti sembrano un efficace strumento per la preservazione della memoria storica informatico/ludica? Hai mai pensato di sviluppare un Sedai FPGA?

**MC:** Sì, i sistemi FPGA sono utili alla preservazione storica. Ritengo che l'emulazione software sia un buon viatico per sviluppare in FPGA, ma non ho pensato di svilupparci Sedai. Comunque, ho accolto il suggerimento di tanti utenti di prevederne il supporto.

**GB:** Il futuro di Sedai Linux, obiettivi a medio e lungo termine. Cosa prevede il calendario dei lavori?

**MC:** A breve e medio termine il calendario prevede il rilascio di una versione di aggiornamento del sistema e degli emulatori, poi seguirà il rilascio del Sedai Basic e, forse, della versione per Olinuxino A64. Quest'ultimo è un ottimo candidato per una versione portatile di Sedai, essendo dotato di circuiteria per l'alimentazione a batteria e la relativa ricarica. A lungo termine è prevista la creazione di una piattaforma hardware dedicata. Questa è la parte più ardua ed affascinante del progetto Sedai, forse quella per cui sono state partorite le idee più copiose. La piattaforma prediletta è ARM, ma si guarda anche ad altre. MIPS e RISC-V. Sono open source e molto interessanti, ma manca una GPU altrettanto aperta da affiancare. Il più aperto dei sistemi ARM è Raspberry Pi, non per niente Sedai è stata sviluppata intorno al modulo industriale Raspberry Pi CM3. Il tutto dovrebbe essere affiancato da altri sistemi di supporto. Per esempio un microcontroller tipo Cortex-M per la sintesi audio ed un altro per gestire task indipendenti come il download di file. Non dovrebbe mancare almeno un'interfaccia per unità SATA. Sono previste altre funzionalità opzionali, da aggiungere con apposite schede di espan-





sione, quali per esempio linee di I/O e controller per motori passo-passo. Quest'ultima funzionalità dovrebbe permettere di collegare stampanti 3D ed altre macchine CNC che sfrutterebbero un'unica logica centralizzata. Ed ancora unità esterne per la domotica ed altre per la creatività musicale. *Dulcis in fundo*, come richiesto da molti utenti, una scheda di espansione aggiungerà un FPGA per emulare vecchio hardware e sperimentarne ancora. Ovviamente il tutto dovrà essere incluso in un case dotato della sua brava tastiera, possibilmente meccanica. Il design del case sarà forse una delle attività più lunghe. Ce ne vorrà per ottenere un disegno soddisfacente, perché io e gli amici di Sedai non amiamo affatto il gusto per l'orrido!

Non sbagliavamo dunque nel dire che Sedai Linux solo apparentemente rientra nella categoria delle distribuzioni dedicate all'emulazione retro. Come avete potuto apprendere dalle parole del suo papà, questo è un progetto ambizioso e molto interessante, che va ben oltre quello che si può intuire testando le versioni attualmente disponibili, che pure offrono un "fattore di conservazione della memoria storica" novevolmente alto, grazie alla capacità di emulare non solo sistemi ad 8bit, target specifico della distribuzione ma, con prestazioni dignitose, anche macchine Mac, Windows 3.11 e Windows 95, come testimoniano le immagini delle schermate mostrate in questo articolo. Una chiara dimostrazione del fatto che, lavorando con perizia sul software, anche un hardware con performance non particolarmente brillanti come il Raspberry PI può dare molto più di quanto si creda. Ringraziamo dunque Maurizio Cammalleri per questo dettagliato tour esplorativo di Sedai Linux e approfittiamo di queste pagine per lanciare un appello: chi volesse unirsi al progetto, può contattare Maurizio via Facebook ed offrire il suo aiuto. Lo sviluppo di questo prodotto, distribuito a titolo completamente gratuito, richiede notevoli risorse e le più eterogenee competenze; se qualcuno di voi lettori si fosse appassionato a questa singolare ma affascinante distribuzione e desiderasse dare il proprio contributo, si faccia pure avanti!

#### RIFERIMENTI WEB

**Sedai Linux:**  
<https://www.sedai.it/v1/>

**FreePascal:**  
<https://www.freepascal.org/>

**RetroPie:**  
<https://retropie.org.uk/>

**Batocera:**  
<https://batocera-linux.xorhub.com/>

**Lakka:**  
<https://www.lakka.tv/>

**Olinuxino:**  
<https://it.wikipedia.org/wiki/Olinuxino>

**RetroPie Utilities:**  
<http://www.retromagazine.net/getrm.php?id=11>





# DRACULA THE UNDEAD

“Salve piccoli Zombie” com’era solito esordire Zio Tibia durante la presentazione di Notte Horror sull’Italia 1 dei primi anni 90. Siccome è di nuovo Halloween sono tornato ad indossare i panni di Pennywise Mulder e condurvi per mano all’interno di un orrorifica esperienza che, guarda caso, fece la sua comparsa nel mondo dei videogames proprio nel periodo di massima attività del cadaverico Zio sopraccitato.



Dracula The Undead è l’unica avventura grafica mai creata per il piccolo portatile Atari e ci riporta ai fasti del romanzo scritto da Bram Stoker un secolo prima. Per non farci mancare nulla sarà proprio il celebre scrittore che durante l’introduzione al gioco presenterà la storia al giocatore, informandolo che vestirà i panni di Jonathan Harker e rivivrà parte della storia narrata nel libro, differendo però da essa in base alle sue azioni. La realtà è meno romantica di quel che un giocatore moderno potrebbe immaginarsi: no, non esistono finali multipli, a meno che non consideriate tali le tante possibili morti in cui

incapperete se farete la mossa sbagliata al momento sbagliato.

La sinopsi serve a calarci dentro il castello nei carpazi ed il primo impatto non sarà dei più facili. A tutti gli effetti il gioco è un punta e clicca classico ma dal momento che la piccola Lince non possiede un mouse, il gameplay è stato adattato alla croce direzionale (con la quale ci muoveremo all’interno degli ambienti di gioco) ed ai tasti funzione.

Con A potremo selezionare l’azione da svolgere tra una lista comprendente i vari “Prendi”, “Apri”, “Lascia” e così via, mentre con il tasto B confermeremo l’azione scelta.

Inizialmente non sarà affatto intuitivo e vi parrà lento e macchinoso, ma una volta preso il via vedrete che il gioco scorrerà tranquillamente, lasciando le vostre rotelle mentali occupate coi tanti enigmi di cui è pieno il gioco. Oggetti da combinare, personaggi con cui parlare, stanze da esplorare, sì insomma: tutto il classico delle avventure grafiche.



Developer:  
Handmade Soft.  
Anno: 1992  
Piattaforma: Atari  
Lynx  
Genere: Avventura  
grafica

## Copertina



## Creepy Soundtrack



Ogni passo uno scricchiolio, ogni stanza un nuovo inquietante rumore a 16 bit pronto a farci gelare il sangue.





In tutto ciò dove sta l'orrore? Oltre al tema trattato ed all'altissima mortalità del gioco, elemento di per se già strano per il genere di cui fa parte, l'intero comparto tecnico è finalizzato a calarvi in un'atmosfera da racconto gotico.



Infine va citata l'impossibilità di salvare il gioco: o lo finite tutto di fila o siete fregati, perché ogni partita ricomincerà dall'inizio.

La grafica del gioco, molto curata, è completamente disegnata in toni di seppia mentre il sonoro è composto di continui rumori



Prima di disperarvi sappiate però che l'avventura in se si porta a termine piuttosto in fretta una volta imparata la risoluzione di tutti gli enigmi (50-60 minuti) ma per capire cosa sia giusto fare state pur certi che ricomincerete il gioco più e più volte, concludendo la maggior parte delle vostre partite non a causa di un enigma dalla soluzione ignota, ma per mano della cara e vecchia mietitrice!

inquietanti, scricchiolii ed ululati capaci di creare quell'atmosfera vintage che magari non riuscirà a farvi saltare sulla sedia ma di certo saprà porvi nel contesto in cui i primi lettori di Stoker si sarebbero terrorizzati a morte. La sensazione che si ha, nel corso di tutto il gioco, è che si stia davvero vivendo all'interno di un romanzo di fine '800, pur con i mezzi messi a nostra disposizione da un secolo di progressi.

di Pennywise Mulder



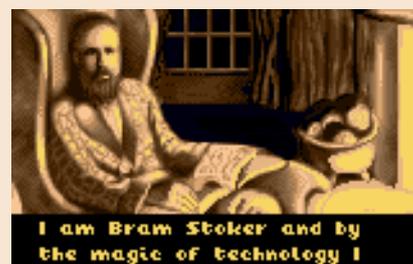
## GIUDIZIO FINALE

### » Giocabilità 75%

L'impatto iniziale potrebbe spaventarvi ma una volta preso il via agirete senza pensare.

### » Longevità 70%

Il gioco si finisce in meno di un'ora ma prima di arrivare al The End dovrete penare sette camicie.





# GANBARE GOEMON YUKIHIME KYUUSHUTSU EMAKI

Anno: 1991  
Producer: Konami

Piattaforma: SNES  
Genere: Avventura/  
Platform



La saga di Ganbare Goemon su Snes è qualcosa di davvero particolare da giocare.

Un mondo pieno di "giapponesità" e di gag nipponiche, ambientato in periodo Edo (periodo che va da metà del 1600 ai primi del 1800) popolato di creature mitologiche, robot e ninja folli.

Il protagonista è Goemon, coloratissimo personaggio che ricorda un attore Kabuki, ispirato alla figura di Goemon Ishikawa, leggendario guerriero ninja e bandito.

Il nostro eroe nasce dalla mente di Etsunobu Ebisu, producer di Konami nel 1986, ma durante il periodo aureo del Super Famicom (Super Nes) si farà notare in una serie di videogiochi incredibili.

Qui vorrei parlare dei primi due capitoli, Ganbare Goemon: Yuki-hime Kyuushutsu Emaki del 1991 e Ganbare Goemon 2: Kiteretsu Shōgun Magginesu del 1993.

Negli anni successivi sono usciti altri due giochi della saga sulla

console a 16 bit nintendo, ma i primi due furono ai tempi innovativi, univano action, elementi di gioco di avventura e una marea di minigiochi. Insomma erano per i tempi due gioconi davvero eterni. Ma partiamo dal primo.

Come sono cambiati i tempi e i modi di videogiocare vero? In un periodo in cui sembra essersi dimenticati delle avventure a tono leggero ecco che risalta fuori questo vecchio titolo per Snes, dove alla guida di Goemon siamo incaricati da una setta di gatti Ninja ad investigare sul rapimento della loro principessa.. Insomma in un mondo moderno fatto di giochi di zombie, spappolamenti o avventure iper realistiche, questo ottimo diversivo di avventura aiuta ad affrontare con leggerezza quasi tutto.. anche i videogames.

Nel gioco è semplice.. la storia ancora di più.. Voi siete Goemon, giovane Ninja ladro che assieme al suo assistente e amico Ebisumaru (paffuto e sessualmente ambiguo) avrai il compito di investigare su misteriosi eventi legati al





rapimento della principessa. Da questo momento in poi verrai catapultato in un'avventura bizzarra in uno strano Giappone feudale, pieno di mostriciattoli e creature assurde.

La storia, come tradizione vuole, è narrata con alcuni dialoghi che spezzano i vari livelli, che a loro volta si dividono in due sezioni principali: nella prima il protagonista esplora un villaggio zeppo di nemici e in cui può comprare equipaggiamento, armi, giocare d'azzardo o qualche minigame (anzi direi tantissimi minigame e alcune chicche come Gradius)

In questi livelli è possibile spostarsi in ogni direzione e la visuale è dall'alto, in stile Zelda. Nella seconda il punto di vista è laterale e il gioco diventa un platform abbastanza impegnativo, simile a Ghost N'Goblins, con tanto di immancabile boss finale, che vi coinvolgerà in uno scontro dalle meccaniche sempre differenti.



Potrà sembrare strano ma le sezioni più complesse sono proprio quelle nel villaggio, perché via via che il gioco progredisce vi troverete di fronte a nemici con schemi d'attacco sempre diversi e non è sempre facile anticiparne le mosse, soprattutto nelle situazioni più affollate. Ogni nemico ucciso potrebbe far cadere una moneta o altri oggetti utili, che vi permetteranno di migliorare le vostre armi o gli attacchi speciali. Le monete sono fondamentali per proseguire perché vi permetteranno di comprare oggetti utilissimi come armature, sandali per correre più veloce e

cibo per recuperare energia e, fidatevi, ne avrete un dannato bisogno. Le sessioni platform hanno il sapore tipico del retrogaming, quello della bile che sputerete per non morire mai.

Goemon ci tiene infatti a farvi capire fin da subito che proviene da un tempo in cui i giochi erano difficili, punto e basta. Non ci sono tutorial, istruzioni o vite extra dietro l'angolo. Ogni salto dev'essere calibrato, ogni movimento va pensato, altrimenti vi troverete di fronte ad una scritta che ormai non si vede quasi più: Game Over.

Fortunatamente, una volta morti potrete ricominciare dall'inizio del livello o dell'ultimo capitolo, anche se questa possibilità è resa rapidamente inutile dalla funzionalità della Virtual Console di salvare il gioco in qualunque momento. Una funzione che potrebbe far storcere il naso ai giocatori più hardcore ma largamente consigliata ai novizi. Il tutto può essere poi reso ancora più divertente dalla modalità cooperativa, che darà il meglio di sé negli scontri con i boss.

Gambare Goemon è in tutto e per tutto un titolo in stile Nintendo: è divertente e spensierato ma da affrontare con rispetto, difficile ma in grado di regalare soddisfazioni. Sarà un discorso da vecchi ma giochi così in giro non se ne vedono più in giro, e se il retrogaming è ciò che amate o se siete curiosi per una volta di capire a cosa si giocava un tempo, compratelo subito. Senza dubbio è uno di quei titoli che rendono speciale la e che speriamo sia presto affiancato da altri grandi classici.

di **Carlo Nithaiah Del Mar Pirazzini**



## GIUDIZIO FINALE

### » Giocabilità 90%

Sia la versione in lingua originale sia la versione in lingua inglese risultano piuttosto intuitive fin da subito e vi catapulteranno in un universo fatto di platform e avventura in una ambientazione follemente nipponica. Amerete ogni singolo momento passato con Goemon.

### » Longevità 90%

Il titolo è spensierato, scanzonato e... infinito! Decine e decine di mini giochi sparsi per il mondo di Goemon. Potete giocare dal semplice tictactoe alla corsa dei cavalli, da Gradius al gioco a quiz sulla storia dei videogames Konami a Pong, ad acchiappa la talpa... E molto altro... Mooolto altro... Infinito. Unico appunto il livello di difficoltà, bilanciato ma alto e che non perdona le imprudenze.





# GANBARE GOEMON 2

## KITERETSU SHŌGUN

Anno: 1993  
Producer: Konami

Piattaforma: SNES  
Genere: Avventura/  
Platform



Eccoci ancora qui.. e se pensavate che il primo capitolo fosse folle... questo secondo dimostra livelli incredibili di demenza videoludica nipponica!

Partiamo dal principio.. Nel 1854, il commodoro americano Matthew Perry firmò con i rappresentanti dello shogunato la convenzione di Nakagawa, grazie alla quale, per la prima volta, si stabilivano intensi rapporti commerciali tra il Giappone ed una potenza straniera, gli Stati Uniti. Del quale, il secondo capitolo della saga di Gambare Goemon uscito su Super Famicom, svela finalmente tutti i retroscena. Lo "straniero" non era un accigliato ed austero ufficiale di Marina, ma un biondo psicopatico con kimono chiamato Magginesu. I suoi uomini, al posto della divisa d'ordinanza, indossavano ridicoli costumi da conigli bianchi giganti. Ed il vero scopo della spedizione non era stabilire un accordo mercantile ma stravolgere ed occidentalizzare lo stile di vita della popolazione nipponica. Ma

questo non accadrà!!! Ed ecco di nuovo in viaggio il prode Goemon, assieme a Ebisumaru e al nuovo arriato Sasuke (un robot costruito dal loro mentore, il vecchio inventore maniaco chiamato Wiser Man) assoldati dallo Shogun per risolvere la questione e liberare il paese.

Se il primo gioco era di per se un piccolo gioiello di gameplay, questo seguito rappresenta l'esempio di "seguito perfetto". Prende tutto ciò che c'era di buono e lo espande, lo migliora e vi aggiunge alcune nuove caratteristiche sviluppando le potenzialità nascoste che forse nel primo mancavano. Il risultato...??? Maestoso! Dopo aver scelto il personaggio preferito fra i tre disponibili, ognuno con punti di forza e punti deboli, il giocare si troverà ad affrontare 40 livelli di puro esempio di "platform old school". Come in Super Mario World, ci si sposta su una mappa con visuale dall'alto che rappresenta i mondi in cui è divisa





l'avventura. In ogni mondo ci sono delle città da esplorare dove interagire con una nutrita varietà di personaggi non giocanti, acquistare armature e cibo per ripristinare l'energia persa, dormire (e salvare la partita) nelle locande e, soprattutto, partecipare ad una varietà sconosciuta di mini-giochi ai quali vincere soldi ed oggetti utili. Ce ne sono talmente tanti che Konami ha deciso di premiare i giocatori più scaltri con un'intero villaggio segreto deputato esclusivamente ad essi dove, tra le altre cose, è possibile giocare una riproduzione del secondo livello di Xexex o imbattersi in caricature di altri famosi personaggi-simbolo della casa nipponica (Sparkster da Rocket Knight Adventures, Simon Belmont da Akumajo Dracula e Gillian Seed da The Snatcher).

Il vero punto di forza di Goemon 2 è rappresentato dai livelli a scorrimento multidirezionale, esempi di maestria strutturale e concettuale. Ogni livello è un microuniverso di complessità, genio inventivo e equilibrio. Non troverete un livello uguale all'altro, anzi troverete numerosi passaggi segreti e uscite seconderie che vi stimoleranno ancora di più. La risposta dei comandi è perfetta e il calcolo delle collisioni è ad un livello tale da soddisfare il più esigente retrogamer. Ma Goemon 2 è di per se un'opera affascinante

e innovativa. Capace di regalarvi ore di divertimento. Innovazione superlativa??? L'introduzione di un diversivo eccezionale come gli scontri con i guardiani finali a bordo dell'Impact, un gigantesco robot dall'espressione beata (in un'orgia di effetti pseudo tridimensionali in Mode 7) aggiunge poi ulteriore sale ad un'esperienza già di per sublime.

Concludo parlandovi della realizzazione tecnica di primissimo ordine, tanto da risultare un adelle migliore mai viste sul NINTENDO a 16 bit, con gran sfoggi di dettagli e colori vivissimi che vanno ad animare un mondo incredibile, vario e... folle!! Unico appunto? La lingua.. non tanto per la difficoltà degli enigmi ma per l'impossibilità per chi non mastica il giapponese di capire i dialoghi che il titolo sa regalare. La Konami non penso mai di esportare il gioco al di fuori del Sol Levante.

Un'ingiustizia alla quale, oggi, fortunatamente possiamo porre rimedio togliendo una patina di oblio che smorzava la brillantezza di una delle gemme più preziose del Super Famicom... grazie all'emulazione, al retrogaming.. e, come nel mio caso al Super Famicom Mini.

di **Carlo Nithaiah Del Mar Pirazzini**



## GIUDIZIO FINALE

### » Giocabilità 93%

Solo in lingua originale ma con una marcia in più. Più intuitivo, più veloce, più dinamico e più platform game. Con la possibilità di giocare utilizzando i tre protagonisti e le loro caratteristiche differenti. Lode assoluta per la sessione sul mega robot gigante che omaggia i vecchi cartoni animati anni 80 e regala momenti di gioco totali.

### » Longevità 95%

Il primo Goemon ha mille mini giochi? Bene.. questo ha un mondo intero pieno.. con alcune chicche assolute.. Infinito.

L'utilizzo del mega robot, dei gadget durante i livelli e i mondi con le uscite multiple... Ore e ore ed ore... ed ore... ed ore... ed ore... di gioco! Yatta!!!





# SUPER MARIO BROS

Anno: 2019  
Coder: ZeroPaige

Piattaforma: C64  
Genere: Platform



Era il 1985, Lido di Valsalva una strana località che ricordava i "bagni della riviera romagnola" ma in realtà si trovava sulle colline di Bologna in una splendida località fluviale. Avevo 8 anni e, tra un cornetto, una coca cola, la ragazzina che mi piaceva e altro... rimasi fulminato sulla via di Shigheru.

Vidi il cabinato di VS Super Mario Bros! Che figata! Grafica e gioco incredibili, ore e gettoni spesi... alla faccia della ragazzina che mi piaceva, del gelato o degli amici. Adoravo passare un po' del mio tempo con Mario.

Scopri poco dopo che questa versione di SMB era una conversione della più celebre versione per Nintendo NES... Cavolo dovevo averlo!! Io avevo il C64, non c'era Mario... Mi aveva detto il mio "spacciatore di cassette" che non era previsto.

C'era un gioco con delle tipe... The Great Gina Sisters... Però io volevo Mario e lo avrei voluto sul Commodore.

Nel 1987 riuscii (dopo un'estate di lavoro devastante) a comprare il Nes con Mario... ma... ma mi rimase un po' di amaro in bocca. Chissà come sarebbe stato vederlo su C64?

Sono passati 34 anni da allora ed ecco che salta fuori il porting perfetto! Non è una conversione, un adattamento... No di certo... E' un porting da Nes sul Commodore.

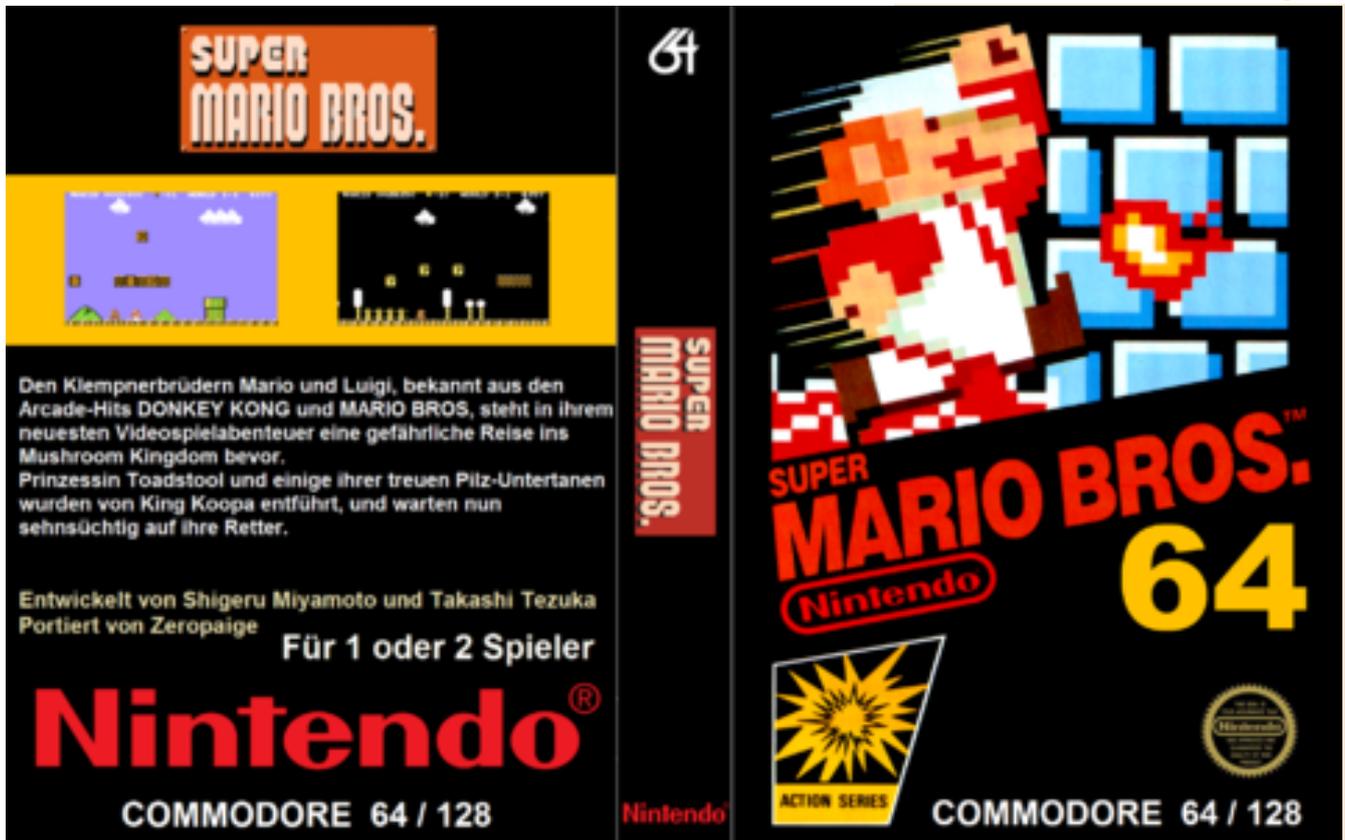
Non vi tedierò con la storia, se state leggendo e se nella vostra vita avete anche solo aperto le finestre di casa sapete già chi è Mario e la sua storia. Vi parlerò di quello che ho visto quando ho acceso il Commodore.

Intanto parliamo di chi ci ha fatto questo dono.

Si tratta dello sviluppatore indipendente ZeroPaige, molto attivo sulla scena del retro gaming. Appena acceso il C64 o l'emulatore c'è da farsi cadere la mascella.

C'è tutto! I livelli, i segreti, i warps.. tutto! Ed è fluido! Davvero fluido.





Ovviamente è un prodotto con qualche limite. Certamente, il solo pulsante del vostro joystick vi complicherà non poco la vita (per saltare dovrete spingere in su la leva), in alcuni momenti il calo di frame rate è evidente...

Ma è SUPER MARIO BROS su C64. Ad 8 anni avrei venduto mio fratello pur di averlo. Il titolo ha avuto non pochi problemi. Distribuito gratuitamente in formato d64 facilmente utilizzabile da emulatore o caricato su c64 MINI, con un po di lavoro si caricherà anche su un biscottone originale o sul vostro C64C.



La Nintendo ovviamente è partita per vie legali.

Consiglio, se lo trovate... scaricatelo oppure cercate lo sviluppatore e fatevene inviare una copia. Ne vale la pena. Ne vale davvero la pena!

di Carlo Nithaiah Del Mar Pirazzini

## GIUDIZIO FINALE

### » Giocabilità 90%

Devo davvero scrivere una banalità? E' Super Mario... e voi avete il C64... La voglia di giocare e il gioco è il perfetto porting del originale.

### » Longevità 95%

Bello come la prima volta. Come la prima "morosa" che avete conosciuto al mare o in vacanza. Duraturo, pieno di warp zone, segreti. E' figlio di NINTENDO.





# SPECIALE RETROHALLOWEEN



Non poteva mancare uno speciale su Halloween e sui retrogames adatti alla festività delle streghe, delle zucche e di tutto quello in grado di risultare lugubre... ma con moderazione. Halloween è sempre una ricorrenza gioiosa a modo suo e quali sono i giochi, anzi, i retrogames più adatti al periodo?

Già l'anno scorso avevamo fatto una cosa simile, ovviamente con altri titoli. Chi vi scrive aveva anche solamente accennato ad un paio di classici, ma ora vi spiegheremo il perché. Soprattutto perché ne troverete qualcuno che a primo acchito sembrerebbe non centrare nulla con questa raccolta.

Bando alle ciance ed andiamo ad illustrare i 13 giochi con cui divertirsi ad Halloween... più qualche menzione nel finale.

## GHOST 'N GOBLINS

Non poteva che partire da lui questa nostra carrellata. Sir Arthur, protagonista di questo classico di ogni tempo firmato Capcom ed uscito in sala giochi nel lontano 1985, che deve salvare la sua bella rapita da un demonio. Non è un tipo fortunato, Sir Arthur: è sì un cavaliere ma senza cavallo... e deve correre con un'armatura pesante ma al tempo stesso quasi inutile. Basterà un colpo per farla cadere e far rimenare il nostro eroe in mutande. Il nostro Arthur dovrà attraversare diverse ambientazioni per compiere l'impresa e potrà lanciare lance, coltelli, asce, torce contro i demoni, piante carnivore, zombie ed altri mostri che gli si porranno davanti.

Il tutto in un'atmosfera lugubre: si parte da un cimitero, meglio di così. Ma anche la colonna sonora con motivi e sonorità (ricorderete certamente la musica di quando iniziava il gioco o quella del primo stage, ma soprattutto quella ugualmente iconica degli high score, ndr) decisamente horror ed azzeccate. Ghost 'N Goblins è così amato da aver dato vita ad un filone che di recente è stato animato dall'ottimo Battle Princess Madelyn.

## GHOST 'N GHOULS

Tre anni dopo, Capcom tornò ancora in sala giochi col suo sequel dal titolo Ghost 'N Ghouls. Ancora Sir

Arthur protagonista e sempre più o meno per lo stesso motivo. Le caratteristiche sono simili ma un po' ampliate con armi leggermente diverse come il giavellotto, grande scure, acqua infuocata, grande scure, dico rotante ed il celebre psycho cannon. Le ambientazioni, ancora più lugubri con nemici sempre più prepotenti. Grafica più rifinita (tre anni del resto sono tanti), ma ritmi sempre incessanti accompagnati da una difficoltà sempre elevata. Anche Ghost 'N Ghouls fu un successo grazie a quello che riusciva a trasudare. Ed indimenticabili anche le ottime musiche.

## VAMPIRE

Dopo due giochi molto conosciuti uno decisamente sconosciuto. Il suo titolo è Vampire ed è un action adventure di Infogrames pubblicato a metà anni '80 su Olivetti Prodest PC 128. Computer che chi vi scrive ha avuto la fortuna (mai capita fino ai 30 e passa anni, ovviamente) di avere.

Un gioco semplice nelle sue dinamiche ma piuttosto complicato in alcuni passaggi. Si vaga all'interno di un cimitero veramente ampio, soprattutto per un titolo dell'epoca, ovviamente infesto di creature infernali. Anche in questo caso il nostro eroe (questa volta improvvisato) deve salvare la sua amata rapita da queste creature per riti non meglio identificati.

Bisognerà quindi combattere mostri e vampiri, pipistrelli ed altre creature andando di location in location, raccogliendo oggetti che servono per sbloccare passaggi o chiavi per aprire porte e cancelli oltre che le armi specifiche per avere la meglio di questo o quel nemico. Stando attenti ad evitare di toccare acqua, fuoco e le creature stesse. Su YouTube dovrebbero esserci ancora dei videogameplay che ne testimoniano la lunghezza e difficoltà. Oltre che atmosfera lugubre. Del resto... siamo di notte in un cimitero e la prima scena mostra anche una vecchia Dyane rossa (auto cult di un notissimo marchio francese).

## RAINBOW ISLAND

Vi chiederete come mai il coloratissimo Rainbow Island sia presente in questo speciale. In effetti è proprio una





nota di “colore” in questo nostro racconto a tinte fosche. Il terzo mondo di gioco del sequel di Bubble Bobble firmato da Taito nel lontano 1987, ci porta in Monster Land, un’isola cimitero oscura infestata da scheletri, mostri, vampiri e licantropi. Non manca nulla. Gli stage ambientati in questo mondo sono decisamente a tema Halloween e sono la nota grigia di un bellissimo capolavoro griffato Taito convertito magnificamente su Amiga ma anche su C64. Il cielo qui diventa nero, le nuvole grigie ed i mostri non sono così tanto rotondi come negli altri mondi.

### THE LAST NINJA

Anche questo titolo lo abbiamo voluto inserire perché include un livello ad ambientazione horror con tanto di scheletri che si animano e cominciano a menare quando si va nei sotterranei del Palazzo del livello quattro. Parliamo di The Last Ninja, capolavoro senza tempo di System 3 (Mark Cale, boss del publisher britannico, ma quando ci farai il quarto? Ndr). Il nostro Armakuni, dunque, nelle Segrete del Palazzo, dovrà affrontare nemici sovranaturali come scheletri e fantasmi ma anche ratti e ragni. L’ambientazione più horror della amatissima serie con un capolavoro di brano incastonato in una colonna sonora incredibile composta dal compianto Ben Daglish e da Anthony Lees.

### AGONY

Il titolo di questo shoot'em up a scorrimento orizzontale per Amiga dice già tutto e non è da confondere con un action adventure in prima persona uscito lo scorso anno per Pc e console moderne. Sviluppato dai belgi di Art e Magic per Psygnosis, Agony è uscito nel lontano 1991. Caratteristica peculiare di questo sparattutto era l’ambientazione ed il protagonista trasformato in una civetta che doveva farsi largo attraverso alcuni stage infestati di creature mostruose di ogni genere che fanno parte del bestiario fantasy.

Cosa ci ricordiamo del titolo e perché sta in questo speciale? Le ambientazioni stupende raccontate in una grafica di primissimo ordine, ma soprattutto alcuni brani della colonna sonora composta da Jeroen Tel, Tim Wright, Martin Wall, Robert Ling, Matthew Simmonds, Martin Iveson ed Allister Brimble. Quella dei titoli per molti fu quasi “sprecata” per un videogioco... quando all’epoca i videogiochi non avevano ancora grossa considerazione come forma d’arte.

### GHOSTBUSTERS

Forse il titolo più “allegro” di questa rassegna: Ghostbusters di Activision per Commodore 64 uscì nel

1984 ispirato – naturalmente – dalla pellicola di successo che uscì sempre in quell’anno.

Il lavoro di Activision fu molto buono soprattutto per l’epoca ed il gameplay tutto sommato variegato ed abbastanza profondo. Bisognava controllare la squadra degli Acchiappafantasmisti e stare attenti anche al portafoglio per quanto riguarda spese e ricavi. Tenere a bada le apparizioni ectoplasmatiche nella città di New York. Non mancano le sezioni di guida (caratterizzate da una visuale dall’alto) dove è possibile anche catturare al volo i fantasmi per strada. Una volta arrivati sul posto, invece, bisognerà catturare lo slimer stando attenti alla sua posizione nella facciata dei vari palazzi. Riuscita l’operazione si dovrà tornare alla sede degli Acchiappafantasmisti, svuotare le trappole e riscuotere le ricompense per comprare equipaggiamenti nuovi e migliori così come auto più performanti. Gradevole graficamente con la colonna sonora digitalizzata.

### DRAGON’S LAIR

Un’ambientazione disneyana, medieval fatansy, una grafica da cartone animato (Don Bluth la realizzò con estrema cura) ed un gameplay da incubo, un vero e proprio inferno soprattutto in sala giochi. Dragon’s Lair uscì originariamente nel 1983 in Laser Disc in sala giochi. Per molti fu la folgorazione: l’aspetto era sorprendente, un vero e proprio cartone animato in cui Dirk The Daring era il protagonista che noi muovevamo per andare a salvare la principessa Daphne rapita da uno stregone malvagio e difesa da un drago sputafuoco all’interno di un castello dalle mille insidie.

Capolavoro d’animazione con una grande ambientazione. Un sonoro all’altezza ma un gameplay davvero difficile in cui il tempismo assoluto ed al millimetro era fondamentale per andare avanti. Non c’era margine di errore: pena le morti più assurde e violente (sempre in tono cartoonesco). Tutte le mostruosità incluse in questo castello oscuro la cui immagine iniziale è iconica e riconoscibile soprattutto da chi ha vissuto il gioco che uscì qualche anno dopo anche su C64 (impossibile una conversione che ricordasse il capolavoro arcade) e su Amiga (ottima realizzazione nonostante i “tagli” ed i tanti dischi che era necessario cambiare per poter giocare sul 16 bit di casa Commodore.

### ALONE IN THE DARK

Ideato da Frederick Raynal e diffuso da Infogrames, Alone in the Dark fu uno dei titoli più memorabili del 1992, nonché molto importante perché sostanzialmente su uno dei primi giochi del genere survival horror. Le atmosfere di Alone in the Dark (che è diventata una serie di successo sia a livello





videoludico che per quanto riguarda i film), sono ispirate chiaramente alle opere del maestro H. P. Lovecraft che viene anche citato nei ringraziamenti. In *Alone in the Dark*, il giocatore può scegliere tra due personaggi collegati in modo differente alla storia. Si possono vestire i panni dell'investigatore privato Edward Carnby, o Emily Hartwood, nipote di Jeremy che non crede al suo suicidio che dà inizio alla storia. Inutile star qui a parlare della trama ma i tanti riferimenti a Lovecraft ed una realizzazione tecnica di assoluto rilievo per l'epoca parlano da soli. Visuale in terza persona con sfondi pre-renderizzati mentre i personaggi e gli oggetti sono poligonali. Un mix che riuscì. Uno di quei giochi che riusciva a mettere ansia acuita dalla colonna sonora originale firmata da Philippe Vachey.

## DOOM

Viene quasi di fermarsi al titolo. *Doom*: sparattutto in prima persona che sostanzialmente ha ridefinito il genere. Demoni, gore, azione dai ritmi alti, ansia a manetta ed ambientazione opprimente, infernale. Chi vi scrive lo gioca solo in compagnia. Ci sarebbe da scrivere moltissimo ma di *Doom* si sa moltissimo.

## HEXEN

È un vero e proprio pezzo di storia. *Hexen* (che in tedesco vuol dire streghe), è un fps del 1995 ad ambientazione dark fantasy firmato da Raven Software (quelli di *Black Crypt*, *Heretic*, e più recentemente di alcuni *Call of Duty* e dei reboot di *Wolfenstein* nel 2009, solo per citare alcuni loro lavori, ndr).

Il gioco ci catapulta in una fortezza dove, una volta scelto il proprio eroe tra quattro, si dovranno superare le orde di mostri mandate da Korax che nel frattempo ha schiavizzato il mondo di Chronos. Forza brutta ma anche cervello fine per superare alcuni enigmi. Potremmo quasi definirlo un antenato di *Doom*. Anche questo è stato fruito tassativamente in compagnia.

## SILENT HILL

Qui vale lo stesso discorso di alcuni classici nominati in precedenza ed un'ansia che non permette al sottoscritto di giocarlo se non alla presenza di amici che sappiano sdrammatizzare. Capolavoro firmato Konami e datato 1999 è sicuramente uno di quei giochi che stanno perfettamente in uno speciale di questo genere. Guidare Harry Marrison alla ricerca della figlia sperduta in questa città fantasma è anche angosciante in alcuni tratti. Frutto di un lavoro di fino del team. Le parti action sono quasi geniali in alcuni tratti: se si fa uno sprint mirare con le armi da fuoco sarà complicato perché la stanchezza ne minerà la mira. Di contro affrontare i nemici con armi bianche sarà altrettanto

rischioso perché il protagonista non è per niente abituato. E questo è solo un aspetto della grandiosità di questo titolo.

## PROJECT ZERO

Action adventure firmato da Tecmo, *Project Zero* uscì nel 2001 su PS2 e poi su Xbox One. Il gioco ci porta nel settembre del 1986. In una notte oscura la giovane Miku entra nel Palazzo Himuro per trovare il fratello che si è perso. Mafuyu, questo il nome dello sfortunato ragazzo, a sua volta era andato a cercare notizie del suo amico Junsei Takamine, uno scrittore di romanzi horror. Una catena di scomparse misteriose.

Miku, che il giocatore dovrà guidare, entra in questo palazzo famoso per i suoi rituali che gli avevano fatto guadagnare la poco invidiabile fama di luogo maledetto. La nostra protagonista trova una macchina fotografica che il fratello aveva portato con sé. Questa macchina fotografica ha la capacità di catturare ed esorcizzare i fantasmi quando viene scattata loro una foto.

Chi l'ha detto che scattare foto sia una passeggiata di salute? La regia e l'ottima ambientazione, coadiuvata da un sonoro a tema soprattutto per quanto riguarda gli ottimi effetti sonori, lo rendono un titolo abbastanza interessante da giocare ad Halloween.

## MENZIONI VARIE

La rivista fatta di carta virtuale (PDF) ha dei limiti e non potevamo certo descrivere anche se sommariamente tutti i giochi che ci venivano in mente. Ma almeno una menzione a qualche titolo bisogna farla. Partiamo dai primi due **Diablo** (atmosfera fosca e soprattutto una colonna sonora letteralmente da brividi. *Tristram* è famosa per il suo arpeggio di chitarra quanto per essere un luogo martoriato).

Non poteva mancare **Medieval** che tra l'altro è uscito anche qualche giorno fa in versione remaster. Lo scheletrico Ser Daniel Fortesque e le buie ambientazioni del regno di Gallowmere sono protagoniste assieme all'ironia di questo action molto divertente.

Chiudiamo con due picchiaduro: il primo è **Mortal Kombat** (e tutta la relativa serie) grazie alla sua seriosità e violenza, il secondo è **Darkstalkers** molto più colorato ma con personaggi adatti al tema.

di Edoardo Ullo





# NOSFERATU THE VAMPYRE

Anno: 1986  
Sviluppatore: Crystal Computing  
Piattaforma: Multi  
Genere: Azione

## Nosferatu - The Vampire



## GIUDIZIO FINALE



### » Giocabilità 75%

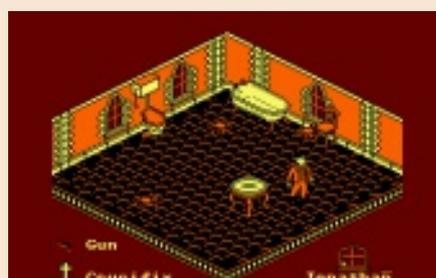
Abbastanza fluido, impegnativo ma di facile utilizzo.

### » Longevità 80%

Ci vorrà un'intera lunazione per imparare il tragitto. E per trovare gli oggetti nascosti.



## Titolo esplorativo



Anche in questo numero di Halloween non ho potuto fare a meno di ripescare un gioco un po' horror dei bei tempi che furono e forse anche un pochino dimenticato.

Dal titolo si può già intuire di che gioco sto parlando e sicuramente a molti di voi suonerà familiare il nome, grazie al film, ai romanzi e naturalmente al videogioco! Esso si presenta come un'avventura isometrica suddivisa in numerose stanze che cambiano colore. Ogni stanza ha diverse ambientazioni, di cui le prime in vari locali della castello in cui è ambientato il gioco... Eh sì, l'avventura inizia proprio all'interno di un enorme maniero in cui il protagonista, un agente immobiliare, sta concludendo una trattativa fino a quando scopre che si trova proprio nella residenza del vampiro! Così ci troviamo presto nel bel mezzo di un labirintico e dispersivo luogo da cui fuggire ma non prima di aver recuperato gli oggetti necessari, tra cui il contratto.

Queste avventure potrebbero disorientare il giocatore soprattutto nella ricerca degli oggetti chiave e sulle strade da prendere per arrivare in fondo al gioco ma non disperate, ci prenderete gusto e memorizzerete le stanze fino a sognarvele di notte.

Per quanto riguarda gli oggetti, cercate cercate e ancora cercate in tutti i posti che vedete, compresa la vasca da bagno non solo per proseguire il gioco ma anche per difendervi (armi) e recuperare l'energia che i nemici vi hanno

tolto anche se non sono fortissimi da evitare e sconfiggere.

Più avanti nel gioco non saremo soli e ci verranno in aiuto due preziosi alleati di cui una sarà essenziale per la sconfitta del vampiro, giunto ormai ad aver vampirizzato un'intera città.

La musica è parecchio carina e perfettamente in tema per il genere e state tranquilli che non vi farà fare incubi, mentre per quanto riguarda i movimenti, sono parecchio fluidi e non dovrebbero subire rallentamenti.

Anche la versione per ZX Spectrum l'ho trovata meritevole e divertente dal punto di vista giocabile. Sarà forse perché alcuni sottovalutavano le avventure isometriche, oppure le menti erano concentrate su ben altri capolavori che questo titolo passò un po' inosservato e solo ora sta avendo le attenzioni che avrebbe meritato all'epoca.

Se cercate un gioco da rigiocare la notte di Halloween, questo non può assolutamente mancarvi e se potete, procuratevi anche il film e qualche romanzo per fare il pacchetto completo Halloween!

Con questo, vi auguro un vampiresco Halloween e al prossimo numero!

di **Daniele Brahimi**





# RALLY BIKE

Publisher: Taito  
Anno: 1988  
Piattaforma: Arcade  
Genere: Racing



Se vi chiedessi di dirmi il primo titolo arcade che vi viene in mente pensando a quelli dedicati al mondo delle motociclette, sono sicuro che la maggior parte di voi mi risponderebbe Hang On di casa Sega, famoso nelle sale giochi anche per il suo originale cabinato. Io invece penserei immediatamente a Rally Bike, prodotto dalla Toaplan e lanciato dalla Taito nel 1988.

Questo gioco, che ha la classica visuale a volo d'uccello tanto in voga anche per altri titoli di corsa di quel periodo, ha come protagonista un simpatico motociclista che deve attraversare vari circuiti delle strade americane che vanno da San Francisco fino a New York.

Nelle sei gare che dobbiamo sostenere lo scopo è sempre lo stesso e cioè cercare di tagliare il traguardo cercando di rientrare in un determinato range di qualificazione sempre più ristretto che ci permetterà di affrontare la gara successiva.

Questo semplice schema di gioco di base è arricchito da tanti elementi che rendono il gameplay di Rally Bike davvero piacevole.

Innanzitutto i nostri avversari, sotto forma di automobilisti e motociclisti, cercheranno di ostacolarci e superarci in tutti i modi per tagliare prima di noi il

traguardo.

Poi i percorsi presentano tante varianti e scorciatoie. Così possiamo trovarci a bordo di un autocarro e spazzare via i nostri inseguitori o possiamo sfruttare trampolini e lanciai in strade alternative ma molto pericolose.

Durante la nostra corsa potremmo aumentare poi il nostro score prendendo punti bonus seminati lungo il percorso ma soprattutto potremmo raccogliere item lanciati da un elicottero che di volta in volta ci permetteranno di attivare un turbo, avere scudi protettivi o prendere taniche di benzina aggiuntive.

Proprio il carburante è un fattore importante nel gioco perchè con il pieno iniziale non riusciremo mai ad arrivare in fondo al circuito. Per fortuna possiamo incontrare stazioni di servizio per effettuare soste che dobbiamo però gestire con attenzione perchè permetteranno ai nostri avversari di superarci.

Infine la grafica colorata, la variabilità dei percorsi e un ottimo mix di musiche ed effetti sonori contribuiscono a rendere Rally Bike un titolo molto vario e mai noioso.

di **Querino Ialongo**

## GIUDIZIO FINALE



### » Giocabilità 90%

Anche se privo di un cabinato dotato di volante o acceleratore, Rally Bike resta un titolo di corse piacevolissimo da giocare anche solo con l'uso dell' arcade stick.

### » Longevità 80%

la bellezza dei fondali e degli sprites unita alle meccaniche di gioco rendono questo titolo divertente da giocare anche a distanza di più di 30 anni.





## Giappone 666^puntata: Il Giapostroano caso di Obachan button

di Michele conte Ugolino

Bentornati alla lettura di Retromagazine! Celebriamo insieme, a distanza di un anno, il secondo numero speciale dedicato ad Halloween.

Cosa pensereste se, per una volta, si parlasse non di retrogaming, ma di retrogamers?

Non parleremo di titoli datati giocati da persone giovani, bensì parleremo di titoli odierni giocati da persone "datate". Perché ho scelto un titolo assonante con quello del film "Il curioso caso di Benjamin Button" ?

Proviamo ad immaginare: gente di una certa età che, seguendo la logica a ritroso del titolo in questo articolo, schiaccia i pulsanti del joypad freneticamente come se fossero dei ragazzini. Tutto normale fin qui. Cosa pensereste però, se il joypad fosse tenuto in mano da una simpatica nonnina?

Il discorso sarebbe più complicato!

Una deliziosa obachan (nonnina) giapponese di 89 anni che, quotidianamente, trascorre la propria pensione videogiocando, sarebbe forse il preambolo per una follia? Forse no. Però se vi raccontassi che, durante lo streaming del gameplay di Grand Theft Auto V, la telecamera che inquadra il viso dell'adorabile nonnina non ruba neppure una smorfia mentre spara in testa al nemico?

Ecco perché ho scelto quel titolo per l'articolo.

Eccovi l'identikit della simpatica nonnina (cfr. figura 2 , 3).

Nome: Hamako Mori, 89 anni.

Città di provenienza: Tokyo, quartiere di Asakusa.

Sport praticato: nuoto, sospeso

all'età di 80 anni.

Giocatrice da: più di quaranta anni.

Giochi preferiti passati : Dragon Quest, Zelda.

Gioco preferito attuale: Days Gone.

Prima console: Epoch Cassette Vision (1981).

Modalità di gioco preferito: Single player.

Segni particolari: Youtuber, gestisce un canale tutto suo chiamato "Gamer Grandma" con più di 40.000 iscritti.

Ogni tanto visiono i suoi video e devo ammettere che mi riempiono il cuore di gioia. Gustare la sua compostezza e le sue adorabili esclamazioni "Tadaaa" mentre elimina i nemici, è veramente fantastico. Notate come si destreggia nel traffico a velocità sostenuta, oppure come seleziona le armi adeguate per proseguire nelle missioni. E' ben intuibile quanto io sia amante delle produzioni 2K e Rockstar Games. Sapere che anche la nostra nonnina, appunto obachan Hamako, gioca a GTA 5, beh, mi illumina di immenso!

Il suo canale Youtube nasce nel 2014, dopo esser rimasta sorpresa dal fenomeno di "Let's Play".

Le voglio augurare di proseguire nel suo intento altri cento anni.

Nell'arco della sua vita la nostra obachan ha potuto vedere la nascita dell'elettronica, la nascita dei videogiochi e l'evoluzione della grafica. Poche persone al mondo come lei hanno potuto mordere in maniera così profonda la mela del mondo videoludico, assaporandone il gusto così efficacemente.

"Mi tiene allenata la mente", ha risposto ai giornalisti.

"Giocare ai videogame mi tiene lontana dalla demenza senile" sostiene Hamako!

Come darle torto? Sarebbe realmente fantastico mantenere una mente così reattiva a quell'età.

Quanto vorrei impartire, a novanta anni, la giusta direzione al mio destriero, mentre un branco di lupi mi sta inseguendo nelle fredde nevi del West Elizabeth ed io, con solo arco e frecce riesco ad ottenere più pelli perfette, dopo averli uccisi e scuoiati tutti. Amici animalisti, state tranquilli, sto parlando del mio adorato "Red Dead Redemption 2".

"Se sei appassionato di moda o fai sport, arriva un momento in cui non puoi continuare a coltivare queste attività". Tuttavia, Hamako, sostiene e rinforza l'idea che invecchiare significa maturare per godere dei frutti attraverso la conoscenza acquisita.

Quali titoli odierni sono giocati dalla nostra nonnina giapponese?

La lista è di tutto rispetto: The Elders Scrolls V: Skyrim, Days Gone, Grand Theft Auto 5, Borderlands 3, Call of Duty: WWII.

Sono sicuro che si aggiungeranno presto altri titoli poiché ha confessato di aspettare con impazienza GTA6 e The Elder Scrolls VI.

Incredibile ma vero, anche io!

"Anche quando invecchi, è meraviglioso continuare a giocare."

sostiene Hamako. Dopo il





Cassette Vision, in Giappone, era dilagata la potentissima console: Nintendo Famicom. La nostra nonnina l'ha veramente amata. Incredibile, anche io! Il famoso Family Computer in Occidente era chiamata Nintendo NES. Nintendo è sempre stata abile nel fidelizzare i propri proseliti e tra essi possiamo annoverare anche Hamako.

"Inevitabilmente, se sei sul campo di battaglia con giocatori più giovani, li rallenterai ... Ma, penso che all'aumentare del numero di giocatori anziani, ci saranno server dedicati in cui ciò non sarà un problema". Intuizione limpida, obiettiva e attinente al rinforzo di un futuro adattamento per sviluppatori next-gen che lavorano nel mondo videoludico: inserire la modalità: "nonni" !

"La grafica dei giochi recenti è davvero sorprendente", ha detto. "Penso che sia davvero meraviglioso vivere così a lungo."

E' incredibile l'arguzia e l'innocenza nel constatare di cantare fuori dal coro, ed è altrettanto energizzante ascoltare parole così proiettate nel futuro.



Figura 1

Volete leggere l'intervista integrale che GameSpark le ha recentemente fatto?

<https://www.gamespark.jp/article/2019/09/23/93264.html>

L'articolo è in lingua giapponese, però vorrei condividere con voi alcuni dettagli dell'intervista datata 23/09/19. Hamako in tale data comunicava di avere 20.000 iscritti. Andiamo a visionare oggi il numero di iscritti al suo canale Youtube, il counter, in "informazioni", ne mostra 40.100 (cfr. figura 1)

Incredibile! Ma quanto sono avanti i giapponesi... io che ho meno della metà dei suoi anni, neppure conosco tutte le funzioni di Youtube, non ho neppure un canale mio, quindi sono alla misera quota di zero iscritti!

Ancor più incredibile è la quantità di video che ha caricato : ben 81 !

Alcuni dati interessanti del suo canale Youtube: in poco più di 5 ore i suoi video collezionano più di mille visualizzazioni.

Un video di GTA5 trasmesso un anno fa ha collezionato mezzo milione di visualizzazioni.

Playlist create: GTA5 (cfr. figura 4), Dark Souls 3, Battlefield 4, Skyrim.

La data di iscrizione del canale risale al 19 Dicembre 2014, da tale data ad oggi, ha collezionato 1.111.670 visualizzazioni.

Considerando che la nostra talentuosa Chiara Ferragni si è iscritta il 23 Agosto 2013 e che vanta 4,5 milioni di visualizzazioni con 137.000 iscritti, temo che a breve il confronto sarà inevitabile.

Scherzi a parte, nel sito di Kotaku.com potrete leggere un breve sunto dell'intervista di



Figura 2





Figura 3

GameSpark:

<https://kotaku.com/89-year-old-grandma-says-video-games-keep-her-mind-shar-1838388356>

Se volete iscrivervi al suo canale Youtube ecco il link:

<https://www.youtube.com/channel/UCc7Ygo3HC1zWJOrtNiwuT3w>

Non sono sicuro che esista un supervisore o traduttore che la stia aiutando nei caricamenti o nel gestire comunicazioni in altre lingue, comunque mi è sembrato un sito molto genuino e pratico, l'unica lingua che ho potuto visualizzare è il giapponese, senza alcun sottotitolo in inglese.

Nonostante ciò vi invito ad iscrivervi, non sarà una streamer all'ultimo grido ma vi assicuro che è una ottima player.

E' tutto ragazzi. Un articolo particolare per un numero di RM particolare. Nel prossimo numero prometto altre "nippe follie" inerenti al mondo del retrogaming.

Nel frattempo, buon Halloween a tutti!



Figura 4





## Firenze Vintage Bit 2019 - Conferenza

*a cura di Associazione Firenze Vintage Bit Onlus  
(Leonardo Vettori e Federico Gori)*

Per Giacomo Leopardi la Luna è una presenza misteriosa ed enigmatica che segue il cammino dell'uomo.

Per il Filosofo Soren Kierkegaard la Luna è la coscienza della terra.

Per il gruppo musicale Pink Floyd, in Dark Side of the Moon, la Luna è una metafora dell'alienazione.

Per i complottisti, la conquista della Luna è una grande rappresentazione cinematografica realizzata grazie all'aiuto di Stanley Kubrick.

Per Jim Storer, sconosciuto studente della Lexington High School, la Luna rappresentò invece una magnifica occasione per utilizzare un costosissimo PDP-8 della scuola, dove programmò in linguaggio FOCAL uno dei primi videogiochi della storia.

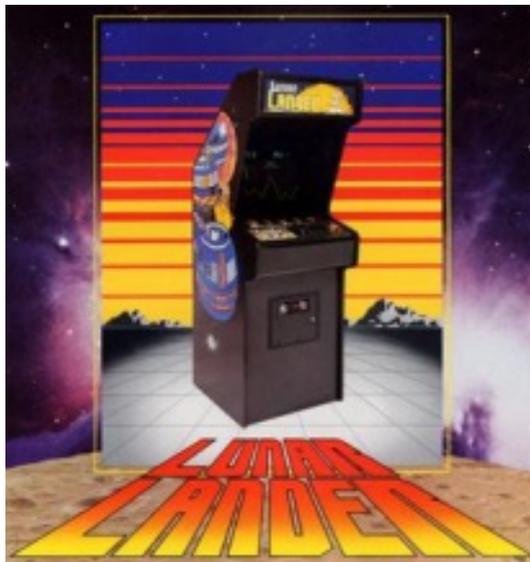


Cinquanta anni sono passati dal quello storico momento del 1969, dove l'uomo camminò per la prima volta sulla superficie lunare, grazie all'uso di innovative tecnologie, che oggi sono diventate preistoriche. Eppure, all'epoca, 152 Kbyte complessivi di memoria ed una frequenza di calcolo da 2 MHz permisero di coordinare il flusso di dati per il sistema di navigazione dell'Apollo 11.

La missione Apollo rese possibile non soltanto approdare fisicamente sulla Luna, ma contribuì a cambiare, attraverso il messaggio che portava, la mente dell'umanità.

Nel 1971 due giovani, Bushnell e Dabney, realizzarono, presso la Stanford University su hardware PDP-11, la loro prima esperienza commerciale. Nacque così Computer Space, oggi considerato il primo videogioco arcade costruito su larga scala, che funzionava previo inserimento di una moneta. Questa intuizione aprì l'epoca dell'oro dei videogiochi.

Nel 1969 la missione Apollo 11 scese sulla Luna; nel 1979 Atari permise con il gioco Lunar Lander di scendere sulla Luna a tutte le persone che avevano una monetina in tasca.



Per una strana coincidenza con il numero della navetta Apollo del 1969, arriva l'undicesima edizione del Firenze Vintage Bit, manifestazione nata nel 2009 da una idea di Walter Pugi e fondata insieme a Maurizio Morandi, che anche quest'anno si terrà nel comune di Lastra a Signa (FI).

Domenica 24 Novembre apriremo nuovamente le porte dello storico "Spedale di Sant'Antonio", per riunire gli appassionati di retrocomputing, retrogaming e di storia dell'informatica.

Per gli espositori, la manifestazione inizia, come di consueto, il sabato mattina. Oltre ad allestire le postazioni, la giornata è essenziale come occasione di ritrovo, sia per scambiare pareri ed opinioni, sia per approfondire la passione che ci unisce.

E' una bella occasione per conoscere anche settori diversi da quello di riferimento di ogni singolo espositore, in un clima tranquillo, scherzoso ed informale. La sera poi si va tutti a cena fuori insieme. La domenica mattina, il Firenze Vintage Bit ha ufficialmente inizio.

Alle ore 09:00 si aprono al pubblico le porte del Sant'Antonio, che rimarranno aperte fino alle ore 13:00. Si riprenderà alle 14:30 dopo la pausa per il pranzo, ed il tutto si concluderà alle ore 18:00.

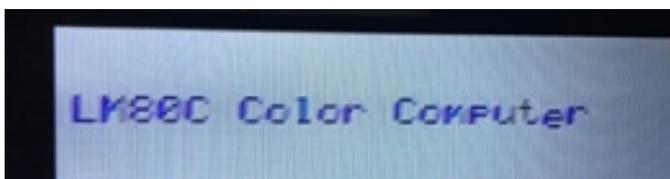
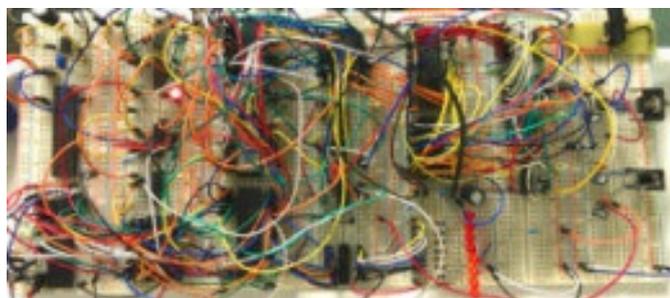
Alle ore 12:00 celebriamo insieme il cinquantesimo anniversario dell'allunaggio, grazie alla conferenza "50 anni di giochi lunari", tenuta dal nostro Federico Gori, che racconterà l'evoluzione delle scorribande video ludiche sul nostro satellite naturale.





Alle ore 15:00 Leonardo Miliani ci parlerà della costruzione del suo computer personale, ovvero un sistema ad 8 bit totalmente progettato da lui, chiamato LM80C (Leonardo Miliani 80 Color).

Se volete saperne di più, potete dare un'occhiata al suo ottimo canale Youtube, dove troverete video ottimamente realizzati e ben spiegati che raccontano l'evoluzione del progetto.



Alle ore 16:30 ci sarà la terza conferenza, curata dal professor Giovanni Cignoni, che racconterà la storia dei primi cinquant'anni della facoltà di informatica di Pisa.



Che cos'è quindi il Firenze Vintage Bit? E' una grande festa, organizzata da Associazione Firenze Vintage Bit Onlus, per gli appassionati ed i curiosi che desiderano rivivere la storia dei computer e delle console che ci accompagnano fin dall'infanzia.



Vi aspettiamo Domenica 24 Novembre 2019 presso lo "Spedale di Sant'Antonio" nel centro storico di Lastra a Signa, alle porte di Firenze.

Per chi volesse invece partecipare in qualità di espositore lo potrà fare iscrivendosi dai primi di novembre tramite il sito ufficiale dell'associazione [www.retrocomputer.org](http://www.retrocomputer.org).

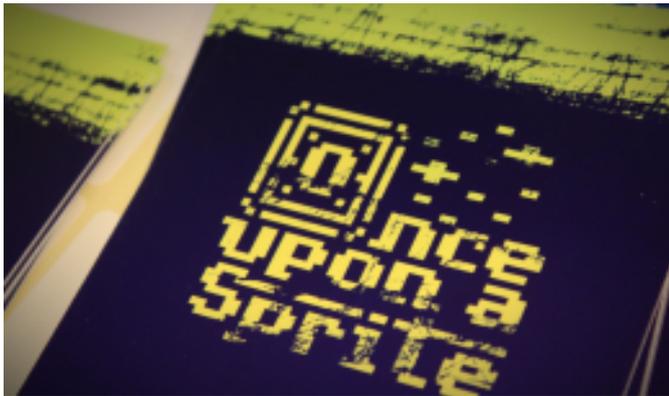
Non mancate!!!





## Once Upon A Sprite 2019 – resoconto breve

Cercamon/David La Monaca



In una Milano bagnata ma soleggiata è andato in scena lo scorso 26 ottobre l'evento "**Once Upon A Sprite 2019**", manifestazione promossa e organizzata da **Andrea Ferlito** (CodeMotion). Per soddisfare le molte richieste, Andrea ha portato OUAS 2019 nella città meneghina (le prime tre edizioni si sono tenute a Roma) nella splendida location della Copernico Blend Tower di Piazza IV Novembre e per l'occasione ha chiesto ed ottenuto l'appoggio degli amici milanesi **Francesco Sblendorio**, **Carlo Santagostino** e **Fabrizio Lodi**.



La giornata ha visto il susseguirsi di presentazioni, dibattiti e talk sui tanti aspetti che hanno contraddistinto lo sviluppo dell'informatica nel nostro paese dagli anni '70 in avanti. Fra i temi affrontati: l'evoluzione dell'editoria professionale, la storia del (mancato) mercato interno dei videogiochi, gli aneddoti sui programmatori solitari degli anni '80, una panoramica sui vasti territori del phreaking, dell'hacking e del cracking durante il ventennio 1980-2000 e un excursus sulla pirateria della scena C64 fino ai giorni nostri. Non sono mancati i progetti che sfruttano strumenti e reti moderne per coniugare vecchie e nuove tecnologie: un framework di programmazione per implementare rapidamente una BBS per PETSCII/C64, la *reverse-engineering* di Space Invaders e sua emulazione su Arduino Nano e un *how-to* sulla programmazione di un emulatore per C64.

Fra i tanti ospiti intervenuti ricordiamo, oltre ai già citati Carlo Santagostino e Fabrizio Lodi, che hanno curato una tavola rotonda sul passato e presente dell'editoria videoludica, **Alessandro De Simone** (Commodore Computer Club), i giornalisti e tecnici

**Claudio Todeschini**, **Paolo Besser**, **Davide Corrado** e **Bonaventura Di Bello** (via Skype), **Andrea Pachetti** (QuattroBit blog), **Stefania Calcagno** (ESoCoP), **Antonio Mazzanti** (Razor 1911), **Silvio Savarino** (HF), **Valerio Lupi**, **Giuliano Peritore**, **Vittorio Signorelli**, **Maurizio Damiani Chersoni** (Developer).

A chiudere l'evento la premiazione della game Coding Challenge sul tema "Milano", organizzata da **OUAS** in collaborazione con **RetroCampus - SYS64738**. Il vincitore della sfida vedrà il suo gioco presentato e recensito nel prossimo numero di RetroMagazine.



Ed è stata proprio la presentazione della nostra rivista **RetroMagazine** ad aprire la manifestazione. **Cercamon** (David La Monaca), accompagnato per l'occasione dalla fotografa **Vovo** (Virginia La Monaca), ha illustrato al folto pubblico di appassionati e addetti ai lavori, la storia, le motivazioni e gli obiettivi del progetto editoriale ideato da **Francesco Fiorentini** nell'ottobre 2017 e portato avanti con entusiasmo e risultati concreti dalla numerosa schiera di redattori ed esperti di retrocomputing e retrogaming.

Nel prossimo numero troverete un più ampio resoconto del riuscitissimo meeting, con tante immagini e dettagli sui protagonisti della tavola rotonda e dei talk che si sono succeduti durante la giornata.

La redazione di RetroMagazine tiene in particolare a ringraziare **Andrea Ferlito** e lo staff di **Once Upon A Sprite** per l'invito, l'ospitalità e la visibilità fornita alla nostra rivista all'interno di un evento senza alcun dubbio fra i migliori in Europa in quanto a livello dei contenuti, qualità degli interventi ed organizzazione generale. La prossima edizione che probabilmente tornerà nell'autunno 2020 a Roma, si preannuncia già come un evento da non perdere per tutti i fan del retrocomputing.



# Gia' 2 anni di RetroMagazine?

Il tempo passa in fretta. Mi verrebbe di dire che è passato poco tempo da quando uscì il fatidico numero 1 di questa fanzine, numero che realizzò **Francesco Fiorentini** con le sue sole forze, facendo diventare realtà la vostra fanzine che più retro non si può :)

Successivamente all'uscita di quel primo numero, la fanzine divenne il frutto dell'impegno di tutti i redattori che pian piano si unirono alla "ciurma" di RetroMagazine.

L'avventura era appena iniziata.

Nacque poco più tardi il sito [www.retromagazine.net](http://www.retromagazine.net) e gli obiettivi della fanzine diventavano sempre più ambiziosi.

Uno degli obiettivi che ci siamo proposti da sempre è stato quello di realizzare articoli di qualità che possano soddisfare non soltanto gli appassionati commodoristi, ma una platea più ampia possibile.

Al fine di raggiungere tale scopo abbiamo cercato da subito di stringere collaborazioni con diverse realtà nell'ambito del retrocomputing, a 360 gradi.

Abbiamo cercato di dare voce, all'interno delle nostre pagine, a tutti coloro che intendessero promuovere eventi, incontri e/o iniziative varie.

Tutto ciò per divertirci, coltivando le nostre passioni, cercando nel contempo di diventare un punto di riferimento per vecchi e nuovi appassionati del retrocomputing, retrogaming, retroprogramming etc.

Al di là della qualità degli articoli (sempre di alto livello grazie alla competenza ed

alla passione di tutti i redattori), abbiamo cercato di migliorare anche la veste grafica della fanzine, adoperando strumenti opensource di impaginazione, e lo sforzo nel cercare di migliorarci è continuo.

Il feedback dei nostri lettori, che diventano sempre più numerosi, ci è prezioso sia per comprendere quali nuove tematiche affrontare, quali argomenti approfondire, sia per comprendere se stiamo procedendo nella giusta direzione.

Come già ribadito anche in passato, la fanzine è del tutto gratuita, ed è il risultato degli sforzi di tutti noi redattori che proviamo a trasmettere ed a condividere con voi lettori la nostra comune passione.

E' indubbio che, nonostante i risultati fin qui raggiunti siano stati estremamente incoraggianti, la strada da percorrere sia ancora lunga. Ci auguriamo di percorrerla a lungo, insieme a tutti voi, amici lettori di RetroMagazine.

Chi volesse darci una mano, come sempre, è il benvenuto.

Per proposte di collaborazione scrivete pure all'indirizzo:

[retromagazine.redazione@gmail.com](mailto:retromagazine.redazione@gmail.com)  
Il da fare non manca, credetemi sulla parola :)

E' il momento di salutarvi, ma non temete. La vostra fanzine preferita tornerà presto, come sempre.

Un saluto da parte mia e di tutta la redazione di RetroMagazine. Buona lettura e...a presto!

**Marco Pistorio**

## Disclaimers

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

RetroMagazine viene concesso con licenza: Attribuzione - Non commerciale - Condividi allo stesso modo 3.0 Italia (CC BY-NC-SA 3.0 IT)  
<https://creativecommons.org/licenses/by-nc-sa/3.0/it/>

In pratica sei libero di: condividere, riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato, modificare, remixare, trasformare il materiale e basarti su di esso per le tue opere, alle seguenti condizioni:

### Attribuzione

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

### NonCommerciale

Non puoi utilizzare il materiale per scopi commerciali.

### StessaLicenza

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

### Divieto di restrizioni aggiuntive

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



RetroMagazine  
Anno 3 - Numero 18

Direttore Responsabile  
Francesco Fiorentini

Vice Direttore  
Marco Pistorio

Ottobre 2019

