



RetroMagazine

World

future days are back

Numero 23 Anno 4 - Maggio 2020 - WWW.RETROMAGAZINE.NET - Pubblicazione gratuita

```

1390 PRINT "PROVA A USARE IL CERVELLO, SE NON E' UN OPTIONAL!":RETURN
1400 IF Y=17 AND OGG(8)=0 THEN OGG(8)=-1:PRINT"OK....":RETURN
1408 IF Y<>2 THEN PRINT "ACCENDITI LA BARBA, IPPOGRIFO!":RETURN
1410 IF LO=4 AND OGG(6) THEN PRINT"Dovrei avere anche una cassetta per farlo!":RETURN
1420 IF LO<>4 THEN PRINT "Dove lo trovo? Su Audio Review?":RETURN
1440 PRINT"OK.. C'e' un messaggio: ";DO#:PRINT"SEI STATO RISVEGLIATO DALL'IBERNAZIONE PER UNA MISSIONE DI EMERGENZA!"
1450 PRINT"L'ASTRONAVE FARMER HA AVUTO UN'AVARIA. DEVI RIPROGRAMMARE IL COMPUTER":PRINT"DI BORDO PER UN ATTERRAGGIO DI FORTUNA."
1460 PRINT"PER FARLO TI SERVE UN FLOPPY DISC CHE CONTIENE I DATI NECESSARI. TROVALO":PRINT"E

```

Listato: ASTRONAVE FARMER (Risveglio in astronave)

- COMAL: la strana storia di un linguaggio dimenticato (seconda parte)
- Grafici in 3D con poche righe di BASIC
- L'angolo dell'oscurità: SEPTERRA CORE
- Giocare a vite infinite sul C64 - "La sfida continua"
- Mister FPGA... Un anno dopo



RetroMath: codici segreti

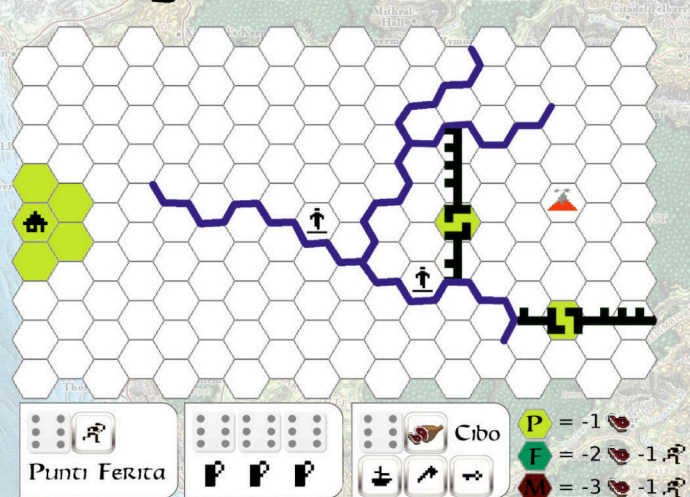
CHUBBY GRISTLE (Recensione C64)



Trespassers will be sat on!

HIScore: 00
SCORE: 00

Cloudy Mountain



CLOUDY MOUNTAIN - Boardgame per un giocatore



GTA: le origini del mito

Primavera 2020: nell'aria c'è profumo di cambiamento

Nonostante l'epoca francamente un po' buia che viviamo, questa primavera ormai inoltrata porta tanti cambiamenti nelle nostre esistenze, forse finalmente verso un ritorno alla normalità e alla vita che conosciamo, dopo l'incubo CoViD-19. Anche in seno alla nostra iniziativa editoriale dedicata al retrocomputing, le novità non mancano, anzi ce ne sono tantissime che si sono già concretizzate o stanno per farlo.

Cominciamo dal nome della vostra (speriamo) amata rivista. Da questo numero il nome della testata muta in **RetroMagazine World**. Siamo stati parsimoniosi e modesti: abbiamo solo aggiunto una piccola parola ("World") al nostro nome storico, soprattutto per segnalare la nostra nuova vocazione a rivolgerci a tutta la comunità internazionale, non più soltanto a quella, benché numerosa, del Bel Paese.

Come intendiamo farlo? Beh, in realtà, l'abbiamo già fatto lo scorso 2 maggio con l'uscita del numero zero di **RetroMagazine English**, una pubblicazione pilota, interamente in lingua inglese, dedicata a tutti i fan di retrocomputing, retrogaming e retrocoding sparsi sul pianeta, che da tempo ci chiedevano a gran voce di portare in una lingua "franca" (non a vantaggio dei cugini transalpini ma nel senso di lingua ufficiale, comprensibile a tutti) i contenuti e le rubriche che da oltre due anni e mezzo raggiungono i lettori italiani. Il successo di quest'uscita è andato oltre ogni più rosea previsione e ci ha definitivamente convinto a proseguire le pubblicazioni in inglese. Dunque, a partire da questo numero 23, la traduzione completa in inglese sarà pubblicata a distanza di circa una o due settimane da quella in italiano. L'uscita in inglese di RMW includerà l'aggiunta di una selezione degli articoli e delle recensioni più interessanti dei primi 22 numeri italiani.

Anche il nostro logo e la veste grafica generale sono stati leggermente modificati, con l'aggiunta di un piccolo pianeta in grafica 8-bit e la comparsa di un nuovo slogan ("**Future days are back**"), sul quale è doveroso spendere due parole. Sicuramente si tratta di uno slogan un po' nostalgico, ma cerca di coniugare il nostro passato di ragazzi smanettoni e instancabili videogiochiatori con il futuro che allora molti di noi intravedevano, immaginando di far diventare una professione quella che all'inizio era soltanto pura passione. Allora avevamo la sensazione che tutto fosse possibile in un mondo che pian piano si apriva all'era digitale e provavamo uno stupore continuo di fronte a giochi, applicazioni e demo grafiche/sonore generate mirabilmente dai nostri tutto sommato umili computer a 8-bit. In questo nostro presente, per certi versi molto incerto, possiamo ancora ritrovare quelle sensazioni, tornando un po' alle origini e riscoprendo una maniera di concepire l'informatica e di utilizzare le nostre vecchie macchine in chiave moderna. Uno dei modi migliori per attuare questo piano è leggere il numero che avete per le mani. Perché è questo che facciamo noi di RMW: vi riconsegniamo quei giorni di spensieratezza e di divertimento, senza dimenticare i vantaggi tecnologici che ci può fornire il presente.

Le novità per RMW e per tutti voi non finiscono qui. Da pochi giorni abbiamo aperto il nostro profilo Twitter ufficiale **@RetroMagazineW**, che vi invitiamo a seguire per essere sempre aggiornati sulle nostre pubblicazioni, leggere tutti i post con anticipazioni e news sugli ultimi giochi e retro-progetti prodotti dalla comunità internazionale.

Da questo numero pubblichiamo la rivista attraverso la licenza **Creative Commons 4.0 International**, ribadendo il carattere assolutamente non-profit della nostra iniziativa editoriale, priva di pubblicità e totalmente gratuita. Nell'ultima pagina potete trovare il riquadro "Disclaimer" con i dettagli della licenza di distribuzione adottata.

Non ne avete ancora abbastanza? E allora sappiate che stiamo anche lavorando al nuovo sito web di RMW e a molti altri retro-progetti che vi sveleremo nei prossimi numeri.

Il numero che avete scaricato è pieno zeppo di contenuti interessanti e, speriamo, stimolanti. Fra i tanti vi segnaliamo l'articolo su **Cloudy Mountain**, un originale e appassionante board game, creato da Giovanni Galli. Un altro modo per tornare, almeno per qualche ora, ai vostri "giorni del futuro"! Buona lettura a tutti!

David La Monaca

Email: retromagazine.redazione@gmail.com

Facebook: www.facebook.com/RetroMagazine-2005584959715273/

Twitter: twitter.com/RetroMagazineW

SOMMARIO

◇ MISTER FPGA, un anno dopo	Pag. 3
◇ LM80C Color Computer Parte 3	Pag. 7
◇ Intervista a Francesco Sblendorio	Pag. 9
◇ Fast Basic – compilare Locomotive BASIC in CP/M	Pag. 16
◇ Star Watcher	Pag. 19
◇ Giocare a vite infinite sul C64 – “La sfida continua”	Pag. 23
◇ Retomath: Codici segreti	Pag. 26
◇ ASTRONAVE FARMER (Risveglio in astronave)	Pag. 29
◇ Introduzione ad Arexx – terza parte	Pag. 35
◇ COMAL: la strana storia di un linguaggio dimenticato – seconda parte	Pag. 38
◇ Grafici in 3D con poche righe di BASIC	Pag. 42
◇ Giappone 11 ma puntata: Un nuovo Game & watch?	Pag. 47
◇ Chubby Gristle (C64)	Pag. 51
◇ Nightmare saga (MSX)	Pag. 52
◇ CITY BOMBER 4K (C64)	Pag. 54
◇ CLOUDY MOUNTAIN – Boardgame per un giocatore	Pag. 55
◇ GRAND THEFT AUTO I (MSDOS)	Pag. 58
◇ TRAKERS (AMSTRAD CPC)	Pag. 60
◇ CRYSTAL CASTLES (C64)	Pag. 62
◇ L'angolo delle oscurità SEPTERRA CORE (PC)	Pag. 64
◇ SONIC THE EDGEHOG (Piattaforme varie)	Pag. 66
◇ ASTAL (Sega Saturn)	Pag. 68
◇ POWER SPIKES (Arcade)	Pag. 70

Hanno collaborato alla stesura di questo numero di RetroMagazine:

- Alberto Apostolo
- Andràs Vajda
- Antonio Porcino
- Carlo N. Del Mar Pirazzini
- Daniele Brahimi
- David La Monaca (Cercamon)
- Edoardo Ullo
- Ermanno Betori/Alessandro Tiso
- Federico “Arzak1” Gori
- Flavio Soldani
- Francesco Fiorentini
- Gianluca Girelli
- Giuseppe Fedele
- Leonardo Miliani
- Marco Fiaschi
- Marco Pistorio
- Michele Ugolini
- Roberto Lari
- Starfox Mulder
- **Supporto grafico: Irene G. Valeri**
- **Copertina a cura di Flavio Soldani**





MISTER FPGA, un anno dopo...

di Roberto Lari

Ben ritrovati amici, è passato un anno da quando vi abbiamo parlato dei sistemi FPGA (Mister, Mistica e Mist) su RetroMagazine e in quest'articolo torniamo a parlare del Mister che a mio avviso rappresenta la piattaforma FPGA di riferimento del 2020. Insieme andremo a scoprire ed analizzare tutti gli aggiornamenti e le novità di questi 12 mesi che (come se già non lo fosse) renderanno il Mister ancora più interessante e irresistibile.

Se non conoscete il Mister e/o i sistemi FPGA in generale vi invito a leggere gli articoli scritti nei numeri precedenti (13, 14 e 15), che vi introdurranno a un mondo tutto nuovo di riprodurre i nostri amati sistemi a 8 e 16 bit, siano essi computer o console, nonché un numero sempre crescente di arcade (ossia i videogiochi così come sono stati concepiti nei cabinati da bar da cui poi tipicamente erano tratte le conversioni per i sistemi casalinghi). I sistemi FPGA non hanno assolutamente nulla a che fare con la tipica emulazione software che tutti voi conoscete e a cui siete abituati sui vostri PC e/o Raspberry PI.

Il sistema operativo Linux, operante dal lato ARM, adesso è dotato di una shell vera e propria (richiamabile premendo F9 sulla schermata del menù iniziale), da cui, dopo essere entrati con utente "root" e password (di default) "1", possiamo impartire alcuni dei comandi base del sistema operativo. In realtà si tratta di un ambiente ridotto e non un Linux completo.

Questo "micro Linux" non impatta assolutamente sulla qualità della riproduzione hardware in FPGA delle console o dei computer poiché totalmente indipendente e trasparente

Sempre dalla console Linux è possibile lanciare il programma MC, il classico Midnight Commander, il programma che permette avendo 2 finestre sullo schermo di poter agevolmente creare cartelle, copiare o spostare file, rinominarli, cancellarli ecc. E' molto comodo da usare poiché supporta anche l'uso di un mouse USB e l'uso di memorie di massa esterne (chiavette e/o hard disk) per copiare i file da queste alla MicroSD del Mister e viceversa. Ma fate attenzione, le memorie esterne non sono viste all'interno dei core, quindi non potete utilizzarle per caricare i giochi direttamente da esse.

A proposito di shell Linux, se si ha il Mister connesso a Internet (cosa che consiglio caldamente), vi potete accedere anche dal PC tramite SSH, stabilendo una connessione tramite un programma come Putty, da cui sarà possibile effettuare le stesse identiche operazioni, come lanciare gli aggiornamenti, modificare la configurazione se necessario ecc., mentre da Windows (se il router di casa lo permette) è possibile riuscire ad accedere alla MicroSD del Mister (volendo anche mappandola come unità di rete) per poter agevolmente aggiungere, spostare, rimuovere e riorganizzare al meglio i file presenti all'interno della stessa, perfino montare un file VHD (hard disk virtuale usato da alcuni core come Amiga, MSX, 486 per citarne alcuni) per fare le stesse cose al suo interno.

Sempre grazie alla connessione Internet, si otterranno l'ora e la data sempre aggiornate, questo a beneficio di alcuni core come 486 e Amiga.

Tranquilli, nel caso il vostro router vi creasse impedimento, vi sarà sempre possibile effettuare le stesse operazioni tramite FTP e un client come il buon vecchio FileZilla.

Altra cosa interessante che permette di fare il Mister è il boot da una memoria di massa esterna, funziona ma non l'ho testato troppo a fondo, quello che ho potuto notare è che da questa, non si possono solamente lanciare gli script ma per il resto mi sembra essere in tutto e per tutto analogo all'uso tradizionale tramite la MicroSD (immaginate poter usare un HDD esterno da qualche Tera).

I core per impostazione predefinita cercano i giochi all'interno di cartelle con un nome specifico, se si copieranno all'interno di queste, sarà più semplice ritrovarli perché il core in uso non mostrerà la root della MicroSD (obbligando noi a navigare all'interno delle sottocartelle per andare in quella dove li abbiamo copiati), poiché punterà direttamente alla cartella con il nome corrispondente.

Ad esempio Minimig Aga cerca la cartella "Amiga", il core C64 cerca la cartella "C64", attenzione che non tutti i core sono così semplici, ad esempio quello del PC Engine / Turbo Grafx cerca la cartella "TGFX16" e non qualcosa

```

@ mc [root@Vodafone-Retrogaming]~
/root# update.sh
Do you want a
faster update?
Please try
PARALLEL_UPDATE="true"
in your
update.ini

Checking SD-Installer-Win64
URL: https://github.com/MiSTer-devel/SD-Installer-Win64_MiSTer
Nothing to update

Checking Main
URL: https://github.com/MiSTer-devel/Main_MiSTer
Nothing to update

Checking Menu
URL: https://github.com/MiSTer-devel/Menu_MiSTer
Nothing to update

Checking Archie
URL: https://github.com/MiSTer-devel/Archie_MiSTer
Nothing to update

Checking Altair8800_MiSTer
URL: https://github.com/MiSTer-devel/Altair8800_MiSTer
Nothing to update

Checking Minimig-AGA
URL: https://github.com/MiSTer-devel/Minimig-AGA_MiSTer
Nothing to update

Checking Amstrad
URL: https://github.com/MiSTer-devel/Amstrad_MiSTer
Nothing to update

Checking ao486
URL: https://github.com/MiSTer-devel/ao486_MiSTer
Nothing to update

Checking Apogee
URL: https://github.com/MiSTer-devel/Apogee_MiSTer
Nothing to update

Checking Apple-I
URL: https://github.com/MiSTer-devel/Apple-I_MiSTer
Nothing to update

Checking Apple-II
URL: https://github.com/MiSTer-devel/Apple-II_MiSTer
Nothing to update

Checking MacPlus
URL: https://github.com/MiSTer-devel/MacPlus_MiSTer
Nothing to update

Checking Aquarius_MiSTer
URL: https://github.com/MiSTer-devel/Aquarius_MiSTer

```

Figura 1 - Aggiornamenti via SSH





Figura 2 - Cavetto switch ON-OFF

tipo "PCENGINE".

Questo è l'elenco di alcune delle cartelle predefinite ricercate dai core:

AMIGA, AMSTRAD, APPLE-II, AO486, AQUARIUS, ARCHIE, ASTROCADE, ATARI800, ATARI2600, ATARI5200, ATARIST, BBCMICRO, C16, C64, COLECOVISION, GAMEBOY, GENESIS, MACPLUS, NES, ODISSEY2, PET2001, QL, SAMCOUPE, SMS, SNES, SPECTRUM, TGFX16, TI-99_4A, VECTREX, VIC20, ZX81

Una cosa che mi ha lasciato piacevolmente stupito è stata la gestione dei 2 schermi diversi (HDMI il primo e il TV CRT RGB tramite cavo VGA > SCART il secondo) in simultanea. Si possono usare contemporaneamente qualunque core supporti entrambe le uscite video e confrontare le immagini, i colori e tutto quello che potrebbe interessare.

Per caso ho scoperto che i core del Mister riescono a leggere i giochi da dentro i file zippati, ma in sola lettura, pertanto non sono andato oltre, niente file ZIP per me.

Chi di voi non conosce lo ScummVM? Il programma che vi permette di giocare alle avventure classiche della storia come Monkey Island 1 & 2, Zak McKracken, Loom, Maniac Mansion e tante altre? Beh sappiate che non è stato realizzato in FPGA, ma esiste comunque per il Mister!

E' un programma che s'installa manualmente tramite uno script (è molto più facile farlo che spiegarne la procedura) e che sfrutta solamente il lato Linux e il chip ARM, un vero "must" per gli appassionati di questo genere (rigiocare a Sam'n'Max o Full Throttle con il parlato in italiano è davvero stupendo!). Attualmente siamo alla versione 2.2 e supporta anche Monkey Island 3, che è nato su Windows 9x e occupa ben 2 CD.

Per godere appieno dello ScummVM, suggerisco di impostare la risoluzione sullo schermo HDMI a 640x480 come se fosse un PC DOS in VGA, in questo modo l'interfaccia e i giochi utilizzeranno la porzione più ampia possibile dello schermo (4:3 e con le bande nere ai lati) mentre le risoluzioni PAL & NTSC per i core FPGA hanno una loro impostazione indipendente (usando un CRT tramite RGB SCART ho messo rispettivamente 720x576 PAL 50 Hz e 720x480 NTSC 60Hz, ma se usassi un solo schermo HDMI per tutto, li avrei impostati a 1280x720 a 50 e 60 Hz).

Sempre usando Linux, è possibile installare delle avventure testuali (come la trilogia Zork per esempio), tramite uno script apposito chiamato Interactive Fiction.

Durante quest'anno trascorso, sono stati aggiornati tutti i core o quasi, inoltre ne hanno inseriti di nuovi, tra cui citiamo il Chip8, il HT1080Z (clone del TRS-80 model I),

l'AY-3-8500 "Pong-on-a-chip", gli Oric-1 and Oric Atmos, l'Arduboy e finalmente una buona riproduzione hardware dell'Atari S. E a proposito di aggiornamenti sui core, quello più degno di nota sicuramente è il Minimig Aga (Amiga), nell'ultima release l'hanno talmente ottimizzato da avere ottenuto un aumento di velocità così elevato (nella configurazione più pompata) da essere diventato il 33% più veloce di un'Amiga 1200 reale con scheda acceleratrice ACA1233n con cpu 68030 a 40 MHz e 128mb di ram, inoltre hanno migliorato moltissimo la riproduzione della cpu 68020 aggiungendo molte istruzioni che erano mancanti, rendendo compatibili un maggior numero di giochi in versione AGA che prima andavano in blocco. Inoltre adesso supporta ben 384MB della RAM DDR3 della scheda madre del Mister (la DE10 Nano), che saranno configurati come FAST RAM mentre nei 32/128 MB SDRAM saranno configurati i 2 MB di CHIP RAM. Devo dire che ho provato Gloom Deluxe e come prestazioni è diventato fluidissimo!

Come 128MB?, vi starete chiedendo... Eh già, è stato messo in commercio anche il Mister con SDRAM da 128MB. La trovo utile in particolare per alcuni giochi del Neogeo e poco altro, ma se dovete acquistarlo, per la poca differenza economica rispetto alla versione da 32MB, vi conviene comprarlo direttamente con 128MB e starete tranquilli per tutta la vita del vostro Mister.

A proposito di Neogeo, da qualche mese è stato rilasciato il core ufficiale ed in forma gratuita anche la versione 4.0 dell'UNIVERSE BIOS, perfettamente compatibile con il suddetto core.

Parlando del Sega Megadrive, hanno aggiunto il supporto

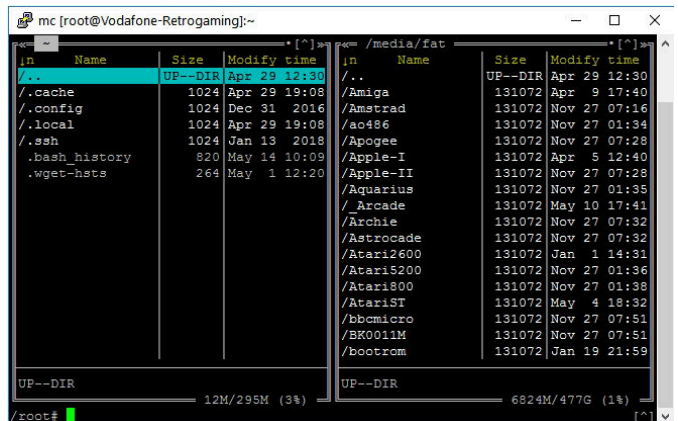


Figura 3 - MC su Putty via SSH

ai CD, creando il core MegaCD, con pieno supporto ai file CUE/BIN, ampliando di fatto il parco giochi disponibile per questa piattaforma e nel momento in cui sto scrivendo, mi è arrivata un'email in cui viene annunciato che da qui a breve vedrà la luce anche il core del PC Engine CD! Ve l'ho detto o no che il Mister è in costante evoluzione?

Sul lato core arcade, il numero dei giochi riprodotti sta diventando sempre più elevato. Sono stati ricreati diversi giochi della piattaforma Sega System 1 (Flicky, Wonderboy per citarne un paio) e sono in arrivo un certo numero anche per la piattaforma CPS1 (Capcom Play System 1), non è da escludere che nel momento in cui starete leggendo quest'articolo, questi giochi non siano già diventati disponibili.

Sul core C64 è possibile fare avere l'autostart di una cartuccia CRT come ad esempio l'Action Replay 6.0 (dotata di Fast Load, che è la mia preferita e la preferisco allo Jiffy





Dos personalmente) copiando il suo file .CRT all'interno della cartella C64 e chiamandola boot3.rom, in questo modo ogni volta che faremo partire il C64, ci ritroveremo come se avessimo la cartuccia già inserita. Comodo, non trovate?

Parlando sempre del C64, nelle versioni più recenti del core è stato aggiunto il supporto ai file formato T64 e al secondo drive (ID 9) con tanto di switch on/off per poterlo accendere e spegnere a piacimento (molte demo e non solo quelle non funzionano se rilevano un drive 9 acceso), davvero una funzione utile specialmente con programmi tipo il GEOS e simili a mio avviso. Inoltre proprio recentemente è stata migliorata parecchio la riproduzione del chip VIC II, permettendo in questo modo di risolvere diversi glitch presenti in alcuni giochi e in diverse demo, un miglioramento davvero notevole e degno di nota, al punto di poterlo definire la riproduzione del VIC II quasi perfetta!

Purtroppo non è tutto rose e fiori e ci sono anche delle cospicue che non ci potrebbero far piacere ma d'altronde non si può avere tutto dalla vita no? (beh il punto 2 è facilmente risolvibile, per il punto 3 possiamo solo sperare per un miglioramento nel tempo come accaduto per il core Amiga e per il punto 1 non si può fare nulla, o così o niente)

1) Parlando di core Arcade verticali, la rotazione di 90 gradi per poterli giocare senza dover ruotare fisicamente lo schermo in verticale è possibile solo sull'uscita HDMI, su VGA / RGB si comporta esattamente come Mist & Mistica (quelli orizzontali non avranno problemi)

2) Il Mister non possiede un pulsante di accensione e spegnimento, ma si può avviare facilmente comprando per pochi euro un cavetto lungo qualche centimetro dotato d'interruttore ON/OFF

3) Il core 486 è paragonabile come prestazioni a un 486 SX 25 MHz, perfetto per giochi più modesti come Aladdin ma ci si deve accontentare per giochi 3D come Doom, speriamo in futuri miglioramenti come accaduto per il core Minimig Aga (Amiga)

Di seguito un paio di errata corrige riferite agli articoli precedenti che per correttezza, riporto qui (un grazie a chi me le ha segnalate) :

L'interruttore della ventola sul Mister non serve ad accenderla e spegnerla ma a regolarne la velocità su alta e bassa, non è possibile spegnerla completamente, ma impostandola su bassa velocità quasi non si sente, specie se si è impegnati a giocare

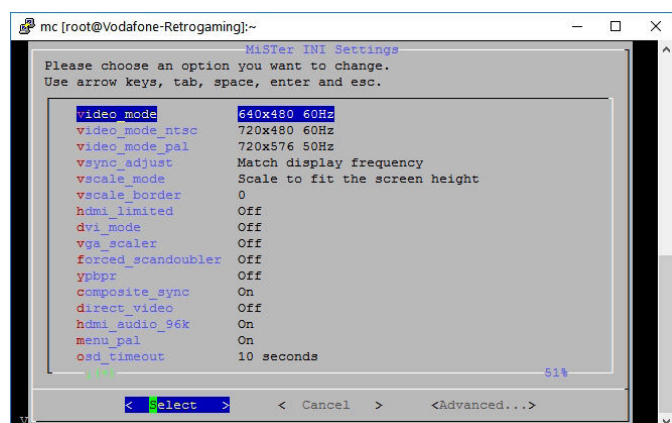


Figura 4 - Settings via SSH

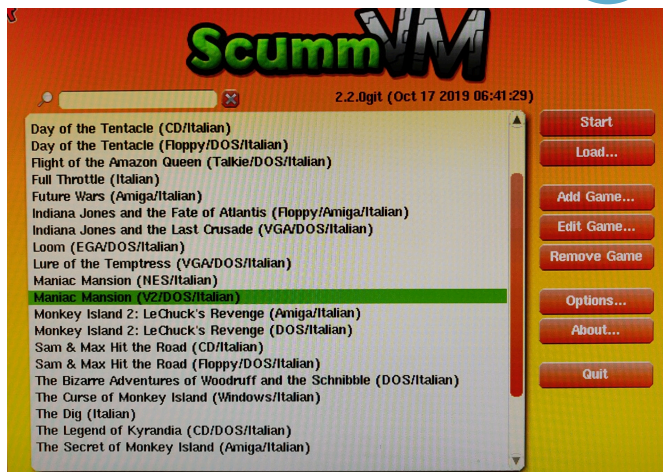


Figura 5 - ScummVM 2.2

La MicroSD da 8GB che danno insieme non è fornita dal venditore del Mister, ma è data insieme alla scheda madre DE10 Nano anche nel caso la acquistaste singolarmente, all'atto pratico, non cambia nulla per chi acquista il Mister completo e pronto all'uso, ma ci tenevo alla precisazione

Come potete vedere il Mister è una piattaforma in costante sviluppo. In un anno è cambiata davvero moltissimo e chissà tra un anno cos'altro potremo avere a disposizione. Se vi doveste chiedere tra il Mister e altri apparecchi come per esempio la Mistica quale potrebbe essere la scelta giusta, vi espongo il mio pensiero e la mia esperienza per aiutarvi a decidere meglio.

Il Mister costa tra una cosa e l'altra circa 300 Euro, se preso nella versione con 128MB SDRAM, cavo video RGB SCART e aggiungendo il cavetto per avere l'interruttore on/off comodo da usare (trovate il link più in basso) ma a mio parere li vale tutti e sapete perché?

Perché si può avere in un solo apparecchio decine di computer, console e arcade a disposizione, giochi a volontà, con qualità RGB per tutti i core compatibili, con pochissimo spazio occupato e usando lo stesso schermo TV/monitor per tutte le macchine. Inoltre quanto costerebbero ognuna di queste macchine se dovessi comprarle originali (ammesso di riuscire a trovarne in vendita e in che condizioni?)

Prendiamo ad esempio l'Apple IIe, o l'Amiga1200 cui dovrei abbinare (per avvicinarmi anche solamente al core Amiga del Mister) a una scheda ACA1233n con O30 a 40mhz a 128mb di RAM.

Facendo un paragone con le console originali (Megadrive, Super Nintendo, PC Engine ecc.), se si volessero usare con la stessa comodità (giochi su SD), si dovrebbe dotare ciascuna con una scheda Everdrive, di conseguenza si dovrebbero spendere altri soldi.

O prendiamo un altro esempio, la console NeoGeo AES, quanto costa nuda e cruda? Molto probabilmente supera da sola il costo del Mister (per non parlare del costo di ogni singolo gioco o anche in questo caso dell'Everdrive).

Per concludere se una persona mi chiedesse oggi, se per risparmiare una volta 120/130 euro ne valesse la pena di andare su un sistema diverso, gli direi di no, che la spesa si fa una volta sola con il prodotto migliore disponibile oggi (e per ancora parecchio tempo), questo anche per un altro motivo:

già adesso alcuni sistemi non ci sono per gli altri FPGA come il Mist e la Mistica, (come il Neogeo per esempio), immaginate quelli di domani... Voi di corridoio non





confermate dicono che stanno realizzando o tentando di realizzare il core della Playstation 1.

Altra grossa differenza come menzionato prima, riguarda i core verticali che sul Mister li puoi giocare lo stesso, pur avendo le bande nere ai lati (anche se solo tramite uscita HDMI) mentre sugli altri sistemi devi ruotare fisicamente lo schermo. Inoltre come visto prima il Mister è raggiungibile via rete Windows (router di casa permettendo), via SSH e via FTP, gli altri no.

Il Mister utilizza come file system anche EXFAT, quindi un singolo file può essere grande anche 16GB, sugli altri l'unico formato supportato è il FAT32 con dimensione massima del file di 3,99GB, Mister ha lo ScummVM, gli altri no, possiede ben 7 porte USB contro le 4 degli altri sistemi e per chi è interessato un'uscita HDMI compatibile con qualunque TV o monitor di moderna generazione.

Ultima ma non meno importante, guardando al futuro è probabile che nel tempo sempre più sviluppatori dei core FPGA si possano spostare sulla piattaforma Mister, abbandonando quelle precedenti. Se così fosse, si rischierebbe di rimanere con un prodotto che nonostante sia valido, potrebbe non ricevere più aggiornamenti nel tempo, senza contare che già adesso alcuni core non sono identici giacché la limitata capacità FPGA dei sistemi diversi dal Mister ne limita le funzionalità implementabili. Infatti su Mist e Mistica il core Minimig AGA (Amiga) non supporta più di 24 MB di FAST RAM e le prestazioni massime raggiungibili sono la metà rispetto allo stesso core presente sul Mister (A500, A600 e A1200 base hanno su entrambe le piattaforme le stesse identiche prestazioni, equivalenti in tutto e per tutto a quelle delle

macchine originali che sono riprodotte) oppure il core C64 che su Mist e Mistica non supporta il secondo drive 1541 su ID 9. Tutto questo perché il chip FPGA del Mister, lo ricordiamo essendo 4,5 volte più capiente della Mistica, permette di implementare molte più cose.

La configurazione di joystick e gamepad USB è tanto semplice e immediata sul Mister quanto più macchinosa sugli altri sistemi, poiché sul primo si fa direttamente dall'OSD stesso mentre sulla seconda devi agire sul file MIST.INI (andando anche a tentativi se non si è molto pratici).

Tendo a precisare che la Mistica e il Mist sono comunque degli ottimi prodotti e che i core realizzati per questi sistemi sono comunque molto validi e fedeli alle macchine che vengono riprodotte.

Il punto però è un altro, se oggi non possedessi nessuno dei 2 sistemi FPGA e dovessi acquistarne uno, sinceramente a parte il prezzo leggermente più basso (che ricordo si deve pagare una volta sola) non riesco a trovare altri motivi per acquistare il Mist o la Mistica oggi.

Se state correndo sul sito di Manuel Fernández a ordinare un Mister in questo momento, sappiate che non sarà disponibile sul sito, vi dovrete mettere in contatto con lui via email e prenotare lo stesso in quel modo, poiché ha sempre un sacco di prenotazioni e tipicamente ci vogliono fino a 3 settimane da quando lo prenotate a quando vi dirà che è pronto.

to PC > fat (\\192.168.1.3) (M:) >

Nome	Ultima modifica	Tipo
_*Arcade	10/05/2020 17:41	Cartella di file
_*Computer	10/05/2020 17:29	Cartella di file
_*Console	14/05/2020 23:46	Cartella di file
_*CPS1	16/05/2020 20:40	Cartella di file
_*Utility	12/05/2020 22:52	Cartella di file
Amiga	09/04/2020 17:40	Cartella di file
Amstrad	27/11/2019 07:16	Cartella di file
ao486	27/11/2019 01:34	Cartella di file
Apogee	27/11/2019 07:28	Cartella di file
Apple-I	05/04/2020 12:40	Cartella di file
Apple-II	27/11/2019 07:28	Cartella di file
Aquarius	27/11/2019 01:35	Cartella di file
Archie	27/11/2019 07:32	Cartella di file
Astrocade	27/11/2019 07:32	Cartella di file
Atari800	27/11/2019 01:38	Cartella di file
Atari2600	01/01/2020 14:31	Cartella di file
Atari5200	27/11/2019 01:36	Cartella di file
AtariST	04/05/2020 18:32	Cartella di file
bbcmicro	27/11/2019 07:51	Cartella di file
BK0011M	27/11/2019 07:51	Cartella di file
bootrom	19/01/2020 21:59	Cartella di file
C16	27/11/2019 01:39	Cartella di file
C64	10/05/2020 17:28	Cartella di file
cheats	27/11/2019 09:47	Cartella di file
Chip8	06/02/2020 15:17	Cartella di file
CO2650	27/11/2019 09:48	Cartella di file
CoCo3	01/05/2020 12:13	Cartella di file
Coleco	27/11/2019 02:10	Cartella di file
config	14/05/2020 23:56	Cartella di file
Electron	27/11/2019 09:52	Cartella di file
Filters	30/11/2019 20:19	Cartella di file
Filters_backup	30/11/2019 20:18	Cartella di file
font	28/12/2019 18:32	Cartella di file
GameBoy	27/11/2019 09:56	Cartella di file
games	06/05/2020 21:35	Cartella di file
Gamma	30/11/2019 20:19	Cartella di file
GRA	15/01/2020 15:50	Cartella di file

Figura 6 - MicroSD raggiunta via rete

Riferimenti:

- <https://manuferhi.com/c/mister-fpga>
Produttore e venditore del Mister
- <http://www.atari-forum.com/viewforum.php?f=117>
Forum di discussione e richiesta di aiuto
- <https://tinyurl.com/ycjapgwf>
Gruppo italiano interamente dedicato ai sistemi FPGA Mister, Mistica & Mist
- <https://tinyurl.com/y54rmza3>
Gruppo internazionale interamente dedicato al Mister
- <https://github.com/MiSTer-devel>
Github dove si possono vedere i costanti aggiornamenti che avvengono nei core

Ringraziamenti:

Ci tengo molto a fare dei ringraziamenti ad alcune persone o meglio alcuni amici che hanno reso possibile la realizzazione di quest'articolo sul Mister, davvero GRAZIE di cuore per la vostra collaborazione:

Angelo Caltabiano; Rino Rinaldi; Mauro Sanna; Matteo Alvares; Paolo Barbieri; Wladimiro Porcelli; Carlo Piacentini; Massimiliano Marongiu; Cristiano Rosadini; Alessandro Bertolotti; Danilo Savioni; Marco Galliani; Massimiliano Conte; Fortunato Pavone; Tommaso Tautonico; Vincenzo Scarpa; Moreno Ligabue





LM80C Color Computer

Un computer autocostruito nel 2019 basato sullo Z80 - parte 3

di Leonardo Miliani

La volta precedente abbiamo analizzato come, grazie ai decoder di indirizzi, la CPU riesca a selezionare determinati chip con cui scambiare dati. È difatti tutto l'insieme di circuiti integrati che lavorano insieme che formano un computer, non è solo la CPU. La CPU, in inglese "Central Processing Unit" (tradotto in italiano significa "Unità di Elaborazione Centrale"), è sì il componente principale ma non è l'unico. La CPU è in pratica come il direttore di un'orchestra. L'orchestra è composta da molti strumenti che vengono suonati in modo sincronizzato ed armonico con gli altri proprio grazie al direttore d'orchestra che impartisce i tempi ed i momenti di attacco ai vari componenti. Portando l'analogia sul piano di un computer, dunque, i vari strumenti sono le periferiche che lo compongono e la CPU il nostro "direttore d'orchestra".

Scegliere determinati strumenti al posto di altri porta alla composizione di un certo tipo di orchestra, così come i circuiti integrati creano le caratteristiche di un sistema nel suo complesso. La scelta che avevo davanti per l'LM80C era la stessa che ebbero alla fine degli anni '70 e durante tutti gli anni '80 del XX secolo i costruttori di computer e console: realizzare completamente in proprio un computer, fabbricandone anche i vari singoli componenti, oppure scegliere la via dell'assemblaggio e realizzare un sistema utilizzando parti prodotte da altri. Per alcuni produttori la prima era una strada già spianata: ad esempio, Commodore possedeva la MOS Technology, e così i suoi computer erano basati su chip progettati e realizzati "in casa". Anche Texas Instruments, con i suoi computer della serie TI-99, seguiva lo stesso percorso, essendo un produttore di circuiti integrati. Altri, invece, si affidavano a ciò che trovavano sul mercato: l'esempio più famoso sono i noti computer MSX, basati su componenti disponibili sui listini di vari produttori. Siccome non possiedo ancora un mio stabilimento di produzione, ho optato per l'assemblaggio di una macchina, ricorrendo però a componenti prodotti non in modo esclusivo per un singolo sistema. Ciò ha permesso non solo di scegliere chip ancora facilmente reperibili in circolazione ma anche di attingere a risorse quali documentazione e software che hanno accelerato e semplificato la progettazione del computer.



Figura 1: L'Atari 800XL montava i chip custom ANTIC, GTIA e POKEY per alleggerire la CPU dalla gestione di video e suono



Inoltre, se avessi optato per chip nati in esclusiva per un certo sistema (ad esempio il VIC-II o il SID del Commodore 64) non solo mi avrebbe legato ad una determinata architettura ma, data la richiesta sul mercato dell'uso di questi chip molto ricercati e non più in produzione da anni, mi avrebbe messo di fronte ad un grosso esborso economico che non volevo affrontare in alcun modo.

Come abbiamo già avuto modo di vedere nei precedenti articoli, l'LM80C è un sistema basato sul microprocessore Zilog Z80. A questo ho affiancato dei chip ausiliari che gestiscono diversi sotto-sistemi del computer, seguendo la filosofia in voga negli anni '80, ossia alleggerire il lavoro della CPU delegando parte dei suoi compiti ad altri sotto-sistemi. Uno dei sostenitori di questo modo di operare è stato Jay Miner, un ingegnere statunitense che ha progettato i chip grafici alla base della console Atari 2600 e dei computer Atari ad 8 bit degli anni '80 (figura 1) nonché l'ideatore del progetto Amiga. Affidare compiti gravosi quali, ad esempio, la gestione dell'audio e della grafica a chip specialistici non solo permette di recuperare una buona fetta di potenza computazionale della CPU da dirottare ad altri compiti ma si impegna in quei compiti chip appositamente progettati. Molti computer economici, come lo ZX80 ad esempio, percorsero una strada diversa, ottimizzando i costi ma non le prestazioni: la CPU, infatti, si occupava anche della generazione del segnale video, e questo compito riduceva drasticamente le prestazioni del sistema. Introducendo nello ZX81 l'istruzione FAST, che disattivava il video durante l'esecuzione di compiti particolarmente gravosi, la Sinclair cercava di porre una pezza a questo problema, permettendo alla CPU di disimpegnarsi dalla generazione dell'immagine e dedicarsi completamente ai calcoli matematici. Proprio perché la generazione degli elementi grafici dell'immagine è uno dei più pesanti compiti alcuni costruttori non si limitarono a progettare chip capaci solo di generare il segnale video ma si adoperarono per integrare nei loro prodotti anche funzioni aggiuntive. Una di queste è la gestione di particolari elementi molto in voga all'epoca perché utilizzati ampiamente nei giochi: gli sprite.

Quando ho gettato le basi del progetto LM80C, ho voluto fin da subito che il computer non solo fosse in grado di generare un'immagine video riproducibile su un comune apparecchio TV ma che potesse anche gestire grafica colorata e sprite. Mi serviva perciò un chip dedicato esclusivamente al comparto video. Doveva, però, essere anche un chip all'epoca abbastanza diffuso così che osse oggi facilmente reperibile ad un prezzo economico. La scelta è ricaduta sul TMS9918A di Texas Instruments, che risponde a tutti i requisiti richiesti. Il TMS9918A nasce nel 1979 come chip grafico per il computer TI-99/4 di Texas Instruments. La prima versione di questo chip si





chiama semplicemente TMS9918 (senza la "A" finale) ed è per l'epoca un buon chip. Ha 3 modalità video: una testuale di 40x24 caratteri, una grafica di 256x192 pixel basata su "tile" (tessere) ed una multicolore da 64x48 pixel. Poco dopo viene rivisto il progetto ed il chip, che diventa TMS9918A (figura 2), vede l'aggiunta di una modalità grafica di tipo "bitmap" (ossia con i pixel gestibili singolarmente). Il chip è definito "Video Display Processor", VDP, nella documentazione ufficiale.



Figura 2: TMS9918A Video Display Processor

Il chip è facilmente interfacciabile a qualsiasi CPU. Necessita infatti di soli 3 segnali elettrici per essere pilotato: CSW/CSR/MODE, ossia un segnale di "scrittura", uno di "lettura" ed uno che imposta la modalità, vale a dire se l'operazione in corso coinvolge la memoria VRAM oppure uno dei registri interni. La VRAM, la memoria video, è separata da quella di sistema e l'unico modo che ha la CPU di accedervi è tramite il VDP. Di solito la connessione viene fatta collegando banalmente i pin CSR e CSW del VDP a quelli RD e WR dello Z80. Per rendere più sicuri i segnali, invece di fare tali collegamenti in maniera diretta sono ricorso all'uso degli OR logici, i chip 7432. La scelta degli OR non è casuale: in logica booleana, il risultato di un OR è vero quando almeno uno dei due operandi è vero. Similmente, un 7432 restituisce un segnale alto con anche un solo ingresso alto. Siccome il segnale è attivo basso, necessito di un'uscita bassa: affinché l'uscita abbia tale livello è necessario quindi che entrambi gli ingressi siano a livello basso. Siccome i segnali di attivazione sono tutti a livello basso, gli OR ci assicurano che i pin CSW o CSR vengano selezionati solo quando si compone una giusta combinazione di segnali sugli ingressi. La figura 3 ci aiuta a capire meglio la cosa.

Il primo OR a sinistra è usato come "interruttore generale": solo nel caso che venga selezionato il VDP dal decoder di indirizzi (visto la volta precedente) e la CPU stia effettivamente tentando di accedere ad una periferica di I/O (stato definito dal pin IORQ) avremo due ingressi a livello basso e, conseguentemente, anche l'uscita dell'OR a livello basso. A questo punto l'uscita del primo OR diventa uno dei 2 input di entrambi gli OR a valle che sovrintendono all'attivazione del segnale di lettura o scrittura. Facciamo un esempio pratico. Immaginiamo di voler inviare un dato al VDP (quindi intendiamo scrivere sul chip grafico). Il segnale su VDP_SEL è basso grazie all'azione, a monte, del decoder di selezione delle periferiche. Ora, siccome si tratta di un'operazione di I/O, anche il segnale su IORQ è basso: avendo 2 ingressi a livello basso, anche l'uscita è a livello basso. Tale segnale entra nei 2 OR a valle. Visto che intendiamo inviare un dato al VDP, il

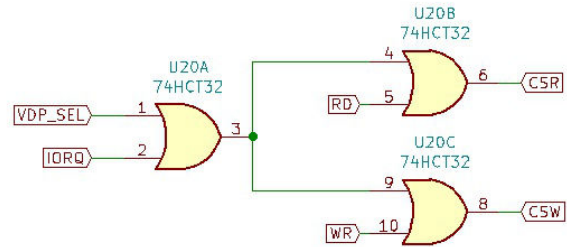


Figura 3: OR logici di selezione dei segnali di lettura e scrittura per il VDP

segnale su WR ("write") diventa basso e contemporaneamente quello su RD ("read") diventa alto. È facile a questo punto verificare che solo il segnale CSW sia basso dato che solo l'uscita della porta OR contrassegnata come U20C (pin 8 in figura) è bassa perché solo qui si hanno su entrambi gli ingressi 2 segnali a livello basso. CSR è invece a livello alto per via del segnale WR che è alto. In ultima istanza analizziamo come entra in campo il pin "MODE" del VDP (figura 4). Questo pin, a seconda del livello che assume, indica se l'operazione è riferita alla VRAM oppure se è diretta ad uno dei registri interni del chip video. Il primo caso si ha quando MODE assume un livello basso, mentre il secondo quando il livello è alto. A questo punto è facile avere questo comportamento collegando il pin MODE ad un pin del bus indirizzi della CPU, in modo da ottenere il cambio di stato con la semplice impostazione del numero di porta di I/O. Nel caso dell'LM80C il collegamento è effettuato con il pin A1. Siccome il VDP è mappato sulla porta logica 48 (0011 0000 in binario), la lettura o scrittura su tale porta imposta il pin MODE a livello basso, attivando quindi lo scambio di dati con la VRAM. Se invece accediamo alla porta 50 (0011 0010 in binario: notare lo stato del 2° bit, evidenziato in grassetto), la selezione del VDP non cambia ma la presenza di un segnale alto sul pin A1 genera contemporaneamente un livello alto sul pin MODE, che il VDP interpreta come uno scambio di dati con uno dei suoi registri.

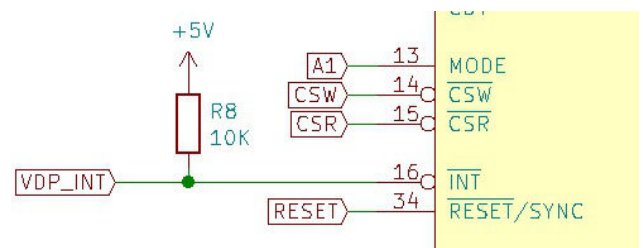


Figura 4: il pin "MODE" del VDP

Sicuramente ci sarebbe molto altro da dire sul VDP ma lo scopo di questo articolo era esaminare il modo in cui la CPU riesce a scambiare dati con il chip grafico, non tanto come questo funziona. Bene, anche per questa volta siamo giunti ai saluti, vi aspetto nel prossimo numero.

Link utili

- Pagina internet di riferimento del progetto: <https://www.leonardomiliani.com/en/lm80c/>
- Schemi elettrici e codice sorgente del firmware: <https://github.com/leomil72/LM80C>
- Pagina su Hackaday: <https://hackaday.io/project/165246-lm80c-color-computer>





Intervista a Francesco Sblendorio

Incontriamo l'ideatore e sviluppatore del progetto PETSCII-BBS

di David La Monaca (Cercamon)

Qui sulle pagine di RetroMagazine World continuiamo ad incontrare sviluppatori, tecnici, programmatori ed esponenti attivi della scena retrocomputing e retrogaming italiana. Stavolta è il turno di Francesco Sblendorio, di mestiere Senior Java Engineer, appassionato di home computer, retrocoding e autore di una soluzione aperta in Java per attivare rapidamente un software server che funzioni come BBS PETSCII, cioè in grado di supportare terminali basati sulla variante Commodore della codifica dei caratteri ASCII. Il framework PETSCII-BBS, i cui sorgenti Francesco ha pubblicato su GitHub, serve proprio a metter su una vera BBS (Bulletin Board System) come quelle di un tempo, cui ci si collegava via modem ed una linea telefonica PSTN. Nella nostra epoca molte BBS sono ancora attive ed accessibili anche via Internet. Una BBS costruita mediante PETSCII-BBS non fa eccezione. Dal lato client basta disporre di programmi di terminale che supportino questa codifica oppure che siano stati scritti appositamente per C64 (40 colonne) e C128 (80 colonne).

Andiamo quindi a conoscere Francesco che, con questo progetto, ha stupito la comunità di appassionati delle macchine Commodore e ha riscosso molto successo durante la presentazione del suo lavoro all'ultimo convegno "Once Upon A Sprite" tenutosi lo scorso ottobre a Milano.

Ciao Francesco e benvenuto nelle pagine di RetroMagazine World. Noi di RMW eravamo presenti alla tua conferenza OUAS 2019 e, come il resto del pubblico presente, abbiamo apprezzato la semplicità e l'efficacia al tempo stesso del tuo lavoro su PETSCII-BBS e così abbiamo pensato di trasferire queste nostre sensazioni ai nostri lettori. Ma prima di parlare delle tue scorribande nel mondo della programmazione e delle soluzioni a 8-bit, cerchiamo di conoscerti un po' meglio. Che ne dici di parlarci, per sommi capi, della tua storia personale, del tuo percorso formativo e della tua professione?

Sono nato nel 1977 a Torino, cresciuto a Bari e poi trasferito a Milano da ormai 15 anni. Ho frequentato il liceo scientifico per poi laurearmi in informatica, materia per cui ho avuto una passione fin dai primi anni di scuola. Altre mie passioni sono la musica (strimpello la chitarra e l'ukulele), la storia contemporanea e la scienza. Faccio il volontario per alcune associazioni che si occupano di divulgazione del pensiero critico. Oggi sono uno sviluppatore software professionista.

Una domanda di rito per una rivista come la nostra: qual è stato il tuo primo computer e che ricordi ne hai?

Il mio primo computer è stato un Commodore 16, un oggetto misterioso che è arrivato in casa mia quando avevo 8 anni. Inizialmente nutro una certa diffidenza verso quella macchina (sono i ricordi di un bambino che scopre qualcosa di completamente nuovo), poi con sempre maggiore interesse. Ricordo in particolare il corso su



Figura 1 - Francesco Sblendorio

cassetta "Video BASIC" del Gruppo Editoriale Jackson, che seguivo pedissequamente e con la voglia di arrivare alla fine della cassetta dove c'erano uno e a volte anche due videogiochi, che vedevo come "premio finale".

Con l'aiuto di mio padre che comprava un po' a caso riviste e libri che recassero la scritta "Commodore" ho fatto pian piano crescere quella passione che oggi è il mio lavoro. Ad un certo punto il C16 si è trasformato in un più potente e popolare Commodore 64 e ho continuato a coltivare la mia curiosità per la programmazione.

Sfogliavo le pagine di Commodore Computer Club inizialmente senza capirci molto, poi le sfogliavo più e più volte, copiando prima meccanicamente i listati presentati e pian piano iniziando a capirne il funzionamento e per poi replicarlo in programmi che provavo a scrivere da solo.

Anche Papersoft fa parte di questa pletora di ricordi, dove c'erano programmi anche per altre piattaforme (TI-99/4A, Spectrum, Apple II) che non possedevo, ma le cui caratteristiche riuscivo a intuire leggendo i listati pubblicati. Insomma, non proprio un passatempo per un bambino,

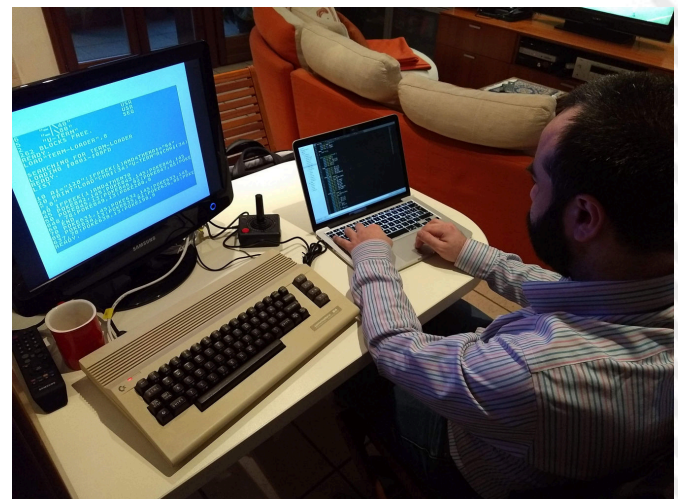


Figura 2 - Durante una sessione di coding





Figura 3 - "Commodore Therapy": relax durante i test

ma mi ci divertivo!

Essendo "nato" come informatico in questo modo, la passione per il retrocomputing è quindi sempre presente in me in quanto ricalca le mie origini e, perché no?, è condita da quel filo di nostalgia della mia gioventù.

Altra domanda obbligatoria su queste pagine: qual è stato il primo gioco che ricordi di aver caricato sul tuo primo home computer?

Lo ricordo benissimo, era "Munch it", un clone di Pacman per Commodore 16. Non vedevo l'ora di tornare a casa da scuola per giocare "al videogioco".

Quando hai scoperto il mondo della programmazione e hai capito che potevi creare qualcosa con un computer invece di limitarti al ruolo di utente di giochi e applicazioni?

Quasi subito, conseguenza della lettura, in primo tempo casuale e in seguito sistematica, della rivista Commodore Computer Club, che piombava in casa come dicevo prima un po' "a caso" grazie a mio papà.

Come hai imparato a programmare? Come molti negli anni 80 e 90, da autodidatta o hai frequentato corsi o studi specifici superiori e all'università?

Sono stato praticamente un autodidatta fino a 18 anni, anche se a un certo punto ho anche frequentato un corso in cui potevo mettere le mani sui PC con MS-DOS, per me inaccessibili diversamente. Poi ho intrapreso gli studi universitari che hanno dato forma e sostanza al guazzabuglio che c'era prima nella mia testa!

Puoi rivelarci il nome del sistema a 8/16 bit che più di ogni altro ti ha fatto appassionare e che ritieni il tuo preferito? E qual è il titolo che ti ha tenuto incollato allo schermo per più tempo?

Di sicuro l'Amiga, che mi ha fatto passare dagli 8 ai 16 bit, alla programmazione con linguaggi come Pascal e C e che quasi paradossalmente mi ha permesso di fare pratica con il mondo dell'MS-DOS tramite il lentissimo emulatore AmigaTransformer.

Per quanto riguarda la programmazione, hai iniziato con

il BASIC, come molti di noi, per poi passare all'assembly o ad altri linguaggi di alto livello?

Ho iniziato con il BASIC, mi sono avvicinato molto superficialmente all'assembly 6502, per poi passare a linguaggi di alto livello come Pascal, C, poi C++, Java, Python, ecc. Oggi guardo con interesse storico linguaggi come Modula-2 e Forth.

Quando si è verificato il (per certi versi triste) momento in cui, come tutti del resto, sei dovuto passare all'acquisto di un PC su architettura x86, per lavoro o passione?

Sono passato dal fido Amiga 500 a un PC 486DX2 nel 1994 ma non lo ricordo come un momento triste, anzi, è stato un salto di qualità notevole: passavo dai 7MHz dell'Amiga con 1MB di RAM e senza hard disk a un "bestione" a 66MHz, ben 4MB di RAM e 200MB di hard disk. Non è stato per lavoro perché ero uno studente, ma appunto per motivi di studio. Il sistema operativo MSDOS e Windows 3.11 mi hanno dato molte soddisfazioni. La mia parte nostalgica mi portava comunque a cercare e usare emulatori come CCS64 e UAE e ricordo con soddisfazione i primi tentativi riusciti di effettuare dump dei floppy C64 e Amiga facendoli poi girare sugli emulatori.

A proposito di lavoro e passione, quali sono i computer ed i sistemi vintage che utilizzi oggi? Sei anche un collezionista?

Non sono un collezionista, m'interessa più che altro riuscire a possedere i computer (retro) che poi utilizzo soprattutto per fini di studio. Per esempio, al momento ho due Commodore 128, ma soprattutto per motivi di "backup". Non mi interessano cose come il particolare modello o avere sigilli integri o i numeri di serie "bassi". Ho usato il Commodore 128 per studiare il sistema operativo CP/M e il modo in cui si poteva programmare. Sul sito di RetroAcademy ci sono diversi articoli che ho scritto su questo argomento e sul mio repository GitHub [<http://github.com/sblendorio>] ci sono alcuni progetti in Pascal, C e Modula-2 tutti scritti proprio per CP/M e in particolare per il Commodore 128.

Ho utilizzato una stampante Commodore MPS-803 (pagata 30 euro) per scrivere un piccolo driver che la rende utilizzabile per stampare testi e grafica (monocromatici) da un moderno computer con Windows, Linux o macOS (sono progetti che si trovano anch'essi su GitHub).

In questo stesso periodo, grazie ai dispositivi prodotti da Pasquale De Luna (la cartuccia KC64Eth e il modem KC64Wifi) e poi alla cartuccia 1541U2+ di Gideon Zweijtzer ho "scoperto" il Commodore 64 connesso in rete ed è l'argomento che più mi appassiona al momento. Ora sono passato dal C64 originale alla sua riproduzione Ultimate64, una motherboard anch'essa "made in Gideon Zweijtzer". Il mondo delle repliche FPGA mi intriga parecchio, perché permette di rendere "eterne" le nostre amate macchine vintage.

Credo di conoscere la risposta, ma ti faccio lo stesso la domanda. Ti piacciono le soluzioni tecniche e le periferiche moderne applicate ai vecchi computer come C64 & company?





Come puoi immaginare, mi piacciono da morire! Già quando nel 2013 scoprii l'esistenza delle interfacce SD2IEC ne rimasi entusiasta. Anzi, è forse proprio quando lessi dell'uso di queste interfacce come periferiche di massa o dell'esistenza del MIST e della tecnologia FPGA che ho "ripreso" in mano l'antica passione per i vecchi computer.

Oltre ai modelli di casa Commodore, quali altri computer a 8 e 16 bit del passato hanno solleticato il tuo interesse (sia per i giochi, sia per l'architettura, ecc.)?

Senza dubbio il Sinclair ZX Spectrum, per via del suo interfacciamento con l'utente del tutto particolare o, meglio, unico. In generale mi piacerebbe mettere le mani sulle macchine CP/M della fine degli anni '70 o dei primi anni '80, come i vari modelli KayPro, quando CP/M era al suo apice.

Quando hai (ri)cominciato a programmare sui sistemi retro? Qual è stata la molla che ti ha condotto sulla strada della perdizione (intesa anche come perdita di tempo e denaro) del retrocoding?

Nel 2001 ho conosciuto il newsgroup `it.comp.retrocomputing` (dove ho conosciuto persone attualmente impegnate in questo ambito, come Elia Bellussi del MuPin, Daniele Lena e Davide Bressanini di Brusaporto Retrocomputing, ma anche Enrico Barbisan che all'epoca organizzava Spresiano Retrocomputing) e il ricordo piacevole dei computer con cui ho iniziato la mia avventura nell'informatica si è riaperto. Così ho iniziato a cercare le macchine a cui ero affezionato e quelle che avevo desiderato ma che non avevo avuto la possibilità di utilizzare. Come immaginabile, dopo poco tempo lo spazio in casa ha iniziato a esaurirsi e quello a disposizione dei computer si è man mano trasformato in una sorta di "magazzino". Li mettevo lì e non avevo spazio desktop per utilizzarli, così ho smesso. Dopo qualche anno mi sono trasferito a Milano e con Facebook ho riscoperto la passione attraverso i gruppi, e così ho deciso di dedicare una scrivania a solo due-tre "esemplari" e di dedicarmi a quelli, scrivendo codice e utilizzandoli attivamente.

Quali sono i principali strumenti di programmazione che utilizzi per i tuoi progetti di retrocomputing?

Principalmente il compilatore CC65 (quindi linguaggio C) per produrre codice oggetto per Commodore 64 (mi ero



Figura 4 - RetroAcademy BBS - schermata principale

incuriosito leggendo due articoli di Fabrizio Lodi) ma uso anche Java lato server (quindi per produrre codice che NON gira sui retrocomputer) al quale si conletteranno macchine vintage, come nel caso delle BBS che si possono creare con PETSCII-BBS. Di recente, con l'annuncio della "10liners competition", l'amico Fabrizio Caruso mi incita a più riprese a partecipare con listati in BASIC e credo che stavolta l'avrà vinta e ci proverò anch'io.

Parliamo ora del progetto PETSCII-BBS, il framework Java che consente a chiunque di attivare una propria BBS su Internet per terminali C64 e C128. Come ti è venuta quest'idea?

Non solo per C64 e C128, ma anche per C16 e Plus/4 e teoricamente anche per i Commodore PET (l'importante è avere un modo che possa connetterli a Internet, per esempio tramite userport). L'idea è nata per scherzo: quando lavoravo in JobRapido (è un motore di ricerca), un collega che conosceva la mia passione per il mondo retro mi disse in modo scherzoso "Francesco, riusciresti a scrivere una versione di JobRapido che funziona con il Commodore 64?". E questo è il mio collega una settimana

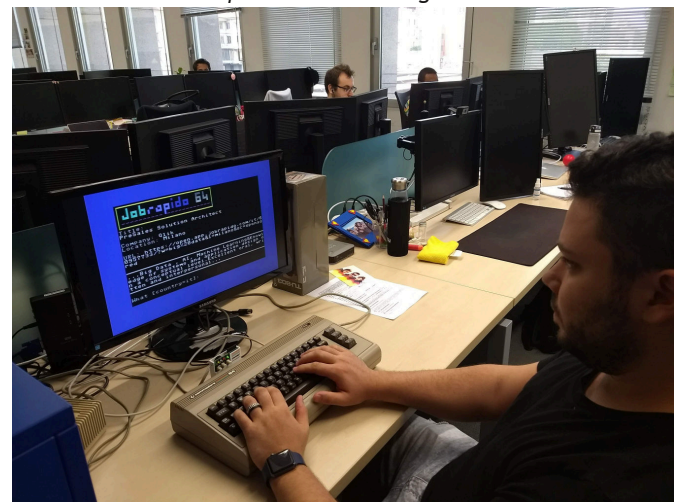


Figura 5 - Collega di lavoro di Francesco che usa JobRapido su C64

dopo:

Avevo appena acquistato la scheda KC64Eth di Pasquale De Luna per collegare il C64 a Internet tramite lo standard RR-NET e avevo provato per la prima volta a collegarmi a delle BBS tramite KipperTerm. La richiesta scherzosa del mio collega mi ha fatto porre la domanda: "Come faccio a implementare un servizio simil-BBS, programmabile a mio piacimento, che funzioni con il C64?". Così ho trovato in giro per la Rete diversi software per realizzare BBS, che però offrono tutti le stesse funzioni: download di software, board di messaggi e altro, tutto customizzabile con grafica PETSCII, ma tutto sommato nulla di "libero" e in ogni caso non facilmente "piegabile" alla volontà di uno sviluppatore (che è il punto di vista da cui partivo).

Ho quindi pensato di scrivere un semplice framework che, mascherando la gestione della connessione fra server e client (socket, gestione degli errori, eventuali protezioni da accessi "spuri"), esponesse solo funzionalità di input e output da e verso il terminale a cui si è connessi, assieme a funzioni di utilità per maneggiare la codifica PETSCII.





Figura 6 - Schermate BBS da C128 e C64

Il risultato è stato PETSCII BBS Builder.

Perché hai scelto proprio Java per costruire il lato server della tua applicazione?

Semplicemente è il linguaggio che utilizzo per lavoro e sul quale ho una solida competenza. Di conseguenza mi è venuto molto semplice implementare la funzionalità "core", legata quindi al multithreading e alla gestione dei socket TCP. Si potrebbe implementare la stessa funzionalità anche in Python, o in C# o in altri linguaggi.

Quali sono state a grandi linee le tue scelte di design e d'implementazione della piattaforma server?

Sono partito dall'idea di scrivere un nucleo ("core") che implementi e "mascheri" la connessione TCP, la gestione del multithread e la conversione del charset PETSCII, in modo da fornire all'utente-programmatore una sorta di DSL con cui realizzare le funzionalità senza conoscere i dettagli mascherati dal core, e così è stato.

Il progetto fornisce principalmente due package: il package denominato "core" e quello denominato "tenants" in cui ci sono numerosi esempi che utilizzano il core, che spaziano da giochi come Forza-4 e il gioco del Quindici, a parser di feed JSON di Wordpress, a servizi di messaggistica.

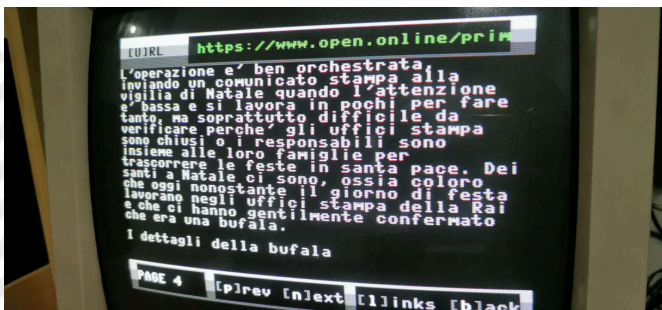


Figura 7 - RetroAcademy BBS - web browser

È online e attiva quella che ho chiamato "BBS Retroacademy", all'indirizzo "bbs.sblendorio.eu" oppure "bbs.retroacademy.it", porta 6510, che nasce come esempio di ciò che si può realizzare con il mio framework. In realtà lo scopo del progetto è fornire il core e spingere potenziali utenti-programmatori a realizzare i propri servizi.

Quanto è complicato per gli utenti di PETSCII-BBS creare una propria BBS online, una volta installata la parte server?

Non è affatto complicato. Bisogna, prima di installare il pacchetto JAR sul server e poi implementare la propria classe che rappresenta la BBS (estendendo la classe base PetSCIIThread) [NdA: fare riferimento al riquadro 1].

C'è un semplice esempio fornito con il framework che realizza di fatto una vera e propria BBS con poche righe di codice e usa solo le funzioni I/O messe a disposizione da PETSCII-BBS. In pratica fa soltanto cose semplicissime: stampa una stringa di benvenuto e chiede il nome utente al visitatore, poi lo saluta ripetendo il nome immesso dall'utente. L'esempio, volutamente semplice, serve solo per mostrare il DSL (Domain Specific Language) fornito e che permette di astrarre dalle funzionalità di rete, concentrandosi in questo caso solo su input, output ed elaborazione.

Fra le demo già presenti nel progetto ci sono esempi di complessità crescente e c'è persino un'implementazione della storica avventura testuale Zork (grazie a Roberto Manicardi, utilizzando il progetto ZMPP di Wei-ju Wu).

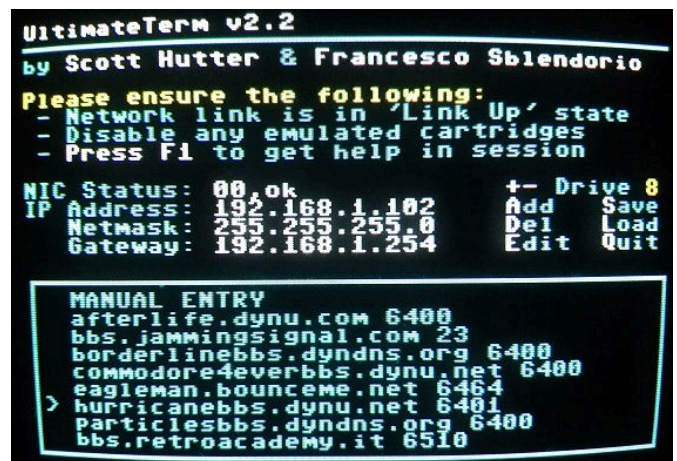


Figura 8 - Ultimate Term in azione

Veniamo adesso allo sviluppo dei terminali client e delle varie utility che sono presenti nell'immagine D64 "Ultimate Term and Demos". Puoi parlarci brevemente di questi tool?

Certamente. Avevo appena iniziato i miei esperimenti con il core del PETSCII BBS Builder sulla neonata "BBS Retroacademy" e lo avevo testato con la cartuccia KC64Eth che implementa il protocollo RR-NET. Avevo già la cartuccia 1541U2+ che come noto include una porta Ethernet, ma dal momento che non supporta il protocollo RR-NET non si può impiegare con i programmi di emulazione terminale che lo utilizzano.

Di lì a poco, l'amico Federico Di Dato mi segnalò che aveva trovato un nuovo programma chiamato "Ultimate Term", postato sul gruppo Facebook "1541 Ultimate" dedicato alla 1541U2+ e alla scheda Ultimate64. "Ultimate Term" era opera di Scott Hutter, che però dai miei test andava sistematicamente in crash nel momento in cui si collegava alla BBS Retroacademy.

Fui contento del fatto che potenzialmente anche gli utenti della 1541U2+ e/o Ultimate 64 potessero collegarsi alle BBS e quindi anche a quelle prodotte col mio framework,



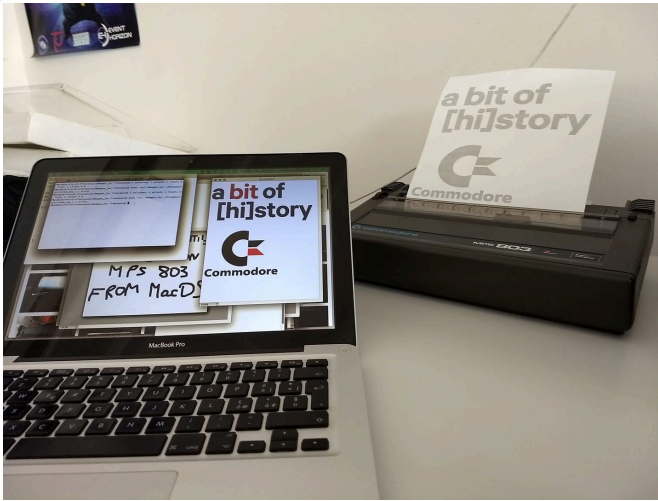


Figura 9 - Stampa grafica da PC su MPS803

così scaricai il programma per vederlo andare in crash e capirne il perché. Contattai così Scott Hutter, che subito mi fece notare che il programma era open source e che il progetto era liberamente scaricabile e clonabile da GitHub (<https://github.com/xlar54/ultimateii-dos-lib>).

In realtà il progetto, un po' come il mio per la BBS, aveva come scopo esporre un core, una libreria (in questo caso in C per il Commodore 64) in grado di offrire al programmatore in modo trasparente le funzionalità di rete della 1541U2+/U64 e forniva come esempio un abbozzo di emulatore di terminale, che era per l'appunto Ultimate Term fornito assieme alla libreria.

A differenza della libreria, curata in ogni particolare, gli esempi tra cui Ultimate Term erano allo stato embrionale (essendo per l'appunto solo degli esempi). Proprio su Ultimate Term (ai tempi giunto alla versione 1.3) notai un paio di bug: uno era il buffer overflow che lo mandava in crash con la BBS Retroacademy, l'altro era un problema di "rendering", che creava glitch e rallentava l'output.

Iniziai quindi a correggere questi bug e il risultato mi piaceva molto, così Scott Hutter mi rese co-autore del repository che da quel momento è mantenuto e aggiornato da me. Si trattava ancora di un esempio embrionale, non consentiva altro che collegarsi ad una BBS senza poter per esempio fare download, senza poter mantenere una lista di BBS se non in modo hard-coded ed era ancora ottimizzabile in termini di velocità di I/O sullo schermo.

Pian piano e con i consigli di Antonio Savona ho ottimizzato ulteriormente il programma e l'ho dotato della gestione di bookmark, della possibilità di download tramite il protocollo XModem e con la possibilità di inviare comandi DOS al disk drive. Al momento lo sviluppo è fermo, ma mi piacerebbe fornire al software ulteriori funzioni, ad esempio implementare anche la funzionalità di upload XModem e aggiungere Punter come protocollo sia per il download che per l'upload.

Sappiamo che PETSCII-BBS si può interfacciare con molti sistemi di collegamento che offrono a C64 e C128 un link verso la rete Internet. Puoi dirci quali sono i principali e, secondo te, quali sono i migliori per rapporto

prezzo e prestazioni?

Ne conosco tre:

1. Tramite il modem Wifi. Ce ne sono parecchi in rete, tra loro molto simili. Personalmente utilizzo e consiglio KC64Wifi di Pasquale De Luna. Arriva a 9600 bps se utilizzato con il software CCGMS (<https://csdb.dk/release/?id=156523>). Il costo del modem è di 23 euro spedito, si tratta della soluzione più economica. La velocità è, tra le tre soluzioni proposte, la più bassa, ma per navigare le BBS non è affatto un problema.

2. Tramite una cartuccia RR-NET compatibile. In questo caso ci si connette tramite la porta Ethernet della cartuccia, usando per esempio il software KipperTerm (<https://csdb.dk/release/?id=174537>). Il costo della cartuccia può variare dai 40 ai 60 euro. La velocità raggiunge i 10Mbps, notevolmente maggiore rispetto al Wifi modem. Per la navigazione sulle BBS si nota la velocità di refresh dello schermo sensibilmente maggiore.

3. Tramite la cartuccia 1541U2+ o con la Ultimate64. Fino a ieri era necessario utilizzare UltimateTerm (<https://csdb.dk/release/?id=181558>) o CCGMS Ultimate (<https://csdb.dk/release/?id=174485>). La velocità in questo caso non l'ho misurata ma è paragonabile a quella del caso RR-NET. Con l'ultimo aggiornamento del firmware è però possibile attivare l'emulazione modem attraverso Ethernet, quindi si può utilizzare CCGMS in modo "classico", impostando come dispositivo di connessione un modem Swiftlink a 19200 bps. Il costo in questo caso è di 150 euro (per la cartuccia 1541u2+ da collegare al C64) oppure circa 200 euro per l'intera scheda Ultimate64 (che sostituisce un intero C64). E' l'opzione più costosa ma include ovviamente tutti gli altri innumerevoli vantaggi dei prodotti stessi!

È possibile collegarsi alle BBS costruite con PETSCII-BBS usando un programma client di terminale che gira su un emulatore VICE opportunamente configurato?

C'è chi ci è riuscito, io no! Anzi, se qualcuno vuole scrivere una guida su questo sarebbe molto gradito da parte mia. Collegarsi alle BBS PETSCII da PC è per me essenziale per testare PETSCII BBS Builder e ho trovato un modo più semplice, usando due programmi in alternativa:

1. SyncTerm <https://sourceforge.net/projects/syncterm/>

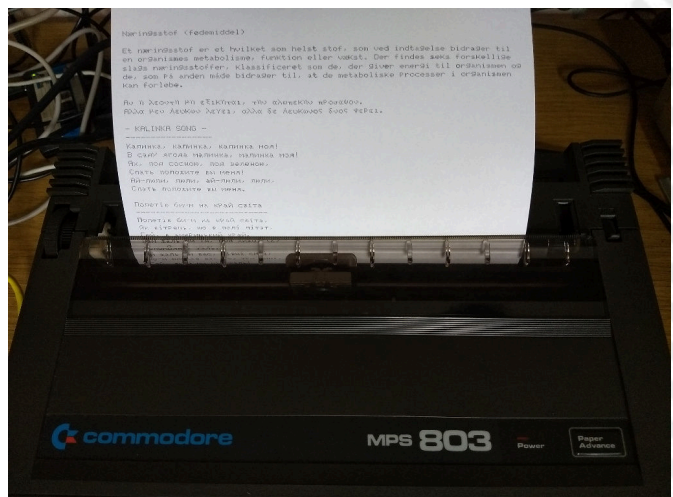


Figura 10 - Stampa testo su Commodore MPS803





```

public class WelcomeBBS extends PetsciiThread {

    // NEVER forget default (empty) constructor
    public WelcomeBBS() {}

    @Override
    public void doLoop() throws Exception {

        // clear screen
        cls();

        println("This is your brand-new BBS");
        println();
        print("Enter your name: ");

        // flush output
        flush();

        // clear input buffer
        resetInput();

        String name = readLine();
        println();
        println("Welcome, " + name + "!");
        println("Press a key to exit");
        flush();
        readKey();

    }
}

```

Riquadro 1 - esempio di creazione di una semplice BBS

disponibile per Mac, Windows e Linux

2. CGTerm - <https://csdb.dk/search/?search=cgterm>

Qui c'è un video fatto dall'amico Dani Hibari per collegarsi tramite SyncTerm - <https://www.youtube.com/watch?v=2FbMa9RagR0>

Una delle BBS create con PETSCII-BBS è "RetroAcademy BBS", un board che ad oggi offre tanti servizi (Televideo, news da vari siti d'informazione, giochi online, servizi di chat e persino un browser web solo testo). Sai di altre BBS in giro per il mondo che sono state attivate grazie al tuo framework? Qual è stato il feedback che hai ottenuto da parte dei tanti appassionati del C64/128 da quando hai reso pubblico il tuo lavoro?

La "Retroacademy BBS" è nata come demo, come "case study" incluso nel progetto per mostrare solo un esempio di cosa si può fare con il framework. E mostra anche il "come". Nonostante la sua natura di "demo" sta avendo però un discreto successo. Non ho stilato delle statistiche sull'utilizzo, ma vedo spesso condivisioni sui vari gruppi Facebook di screenshot C64, C128 e a volte anche di Plus/4 connessi proprio a tale BBS. L'applicazione che riscuote maggior successo è la lettura del Televideo RAI e il download di file dal famoso sito CSDB attraverso la BBS e ultimamente la chat interna. Al momento in cui scrivo ci sono almeno 2 BBS che utilizzano il mio framework:

The Old Net - theoldnet.com:6400

Temporal Vortex BBS - bbs.c64.xyz:6400

C'è inoltre un video a tema retro + notte di halloween su YouTube in cui è stato utilizzato il mio framework per produrre la "fake BBS" del video: <https://www.youtube.com/>

[watch?v=I6i0e3JrV7A](https://www.youtube.com/watch?v=I6i0e3JrV7A)

Parlando di RetroAcademy BBS, hai fatto tutto da solo o hai chiesto supporto di altri retroprogrammatore? Cosa prevedi nel futuro della BBS, diventata ormai un punto di riferimento per tutti i 64isti dotati di interfaccia di rete?

Ci sono diverse persone che hanno contribuito e che ringrazio per lo stato attuale della Retroacademy BBS:

Piero Corasaniti, che ha contribuito con la versione del CSDB-Download che permette il download dei file D64 "interi", destinato al download su SD2IEC

Roberto Manicardi, che ha contribuito all'interfacciamento dei giochi Infocom (Zork il più famoso) con la BBS, estendendo ZMPP (un interprete open source di giochi Infocom)

Wei-ju Wu, l'autore di ZMPP (<http://zmpp.sourceforge.net/>) Richard Bettridge a.k.a. ssshake, che ha contribuito con il browser testuale sviluppato per la sua BBS "theoldnet" (<https://github.com/ssshake/theoldnet-bbs>)

Felice Pagano, che mi ha aiutato a ottimizzare il codice

Henrik Wetterström, per il suo progetto "droid64" (<http://droid64.sourceforge.net/>) che ho utilizzato nella sezione di download da CSDB

Jatin Thakur per il motore di AI che ho usato nel gioco Forza 4 (<https://github.com/jn1772/Connect4AI>)

Brian "bigbhowell" per il motore di AI che ho usato nel gioco del "tris" (<https://github.com/bigbhowell/tic-tac-toe>) Gli artisti che hanno fornito i loro lavori nella sezione "PETSCII art gallery" (Shine, Markku Reunanen, John Canady, Manuel Vio, Ivan Mavilla, Davide Fichera)

Per il futuro, dopo aver implementato la chat, e avendo quindi capito come gestire più utenti, che eventualmente si selezionano fra loro in una sessione, penso di realizzare qualche gioco multiplayer online. Anche qualcosa di semplice come la classica battaglia navale, ma con un giocatore umano remoto: penso che sia divertente ed istruttivo! Essendo open source, mi auguro che il framework venga utilizzato da altri utenti, come quelli che citavo prima (le BBS "oldnet" e "vortex"), con cui scambiare idee e nuovo codice. Per esempio, al momento manca al framework la funzionalità di "receive" per quanto riguarda il protocollo XModem e manca totalmente la gestione del protocollo Punter (sia "send" che "receive"). Se qualche volenteroso vuol dare una mano, è sicuramente il ben accetto!

Prevedi sia possibile aprire ad altre piattaforme retro la possibilità di collegarsi alle BBS create col tuo framework? Anche quelle che non supportano per ovvie ragioni la codifica PETSCII?

Ho dato la priorità alla codifica PETSCII perché mi sono appassionato all'applicazione al Commodore 64. Penso che in futuro adatterò il framework anche allo standard ANSI/VT-100 in modo da poter utilizzare gli emulatori di





terminale su PC e Amiga. Non è una cosa complicata, ma mi piacerebbe avere qualcuno che mi aiuti. Faccio ancora appello a chi mi ascolta/legge!

A quali altri progetti legati al mondo del retrocomputing stai attualmente lavorando o prevedi di lavorare a breve? Ci sono altre piattaforme home computer che hanno attirato il tuo interesse oltre a quelle Commodore 8-bit?

Come detto, mi piace molto il mondo CP/M, per cui ho realizzato quattro giochi in Pascal e in Modula-2: Gorilla (in Modula-2) <https://github.com/sblendorio/gorilla-cpm>, Queens (in Turbo Pascal 3) <https://github.com/sblendorio/queens-cpm>, Hanoi (in Turbo Pascal 3) <https://github.com/sblendorio/hanoi-cpm>, Hangman (in C) <https://github.com/sblendorio/hangman-cpm>

In particolare il primo e l'ultimo (Gorilla e Hangman) utilizzano una libreria che ho chiamato "xterm" (nel senso di "cross-term") che è disponibile sia in C che in Modula-2. In Modula-2: <https://github.com/sblendorio/mod-xterm-cpm>, in C: <https://github.com/sblendorio/c-xterm-cpm>

Tale libreria permette di utilizzare funzioni di gestione dello schermo (cancella schermo, posiziona cursore, cambia colore, ecc.) indipendente dal tipo di terminale usato (per esempio il C128 utilizza lo standard ADM-31, l'Amstrad CPC utilizza Zenith Z19). Altri due progetti riguardano la stampante Commodore MPS 803, per la quale ho scritto due "driver" da utilizzare in congiunzione con il pacchetto "opencbm", che permettono di stampare con questa stampante plain-text UTF-8 e grafica monocromatica collegandola a un normale PC o Mac (con il "cavo" XUM1541): cbmage, per le immagini: <https://github.com/sblendorio/cbmage>; cbmtext, per i file di testo UTF-8: <https://github.com/sblendorio/cbmtext>

Outro

Ringraziamo Francesco per il tempo che ci ha dedicato, per la sua grande disponibilità e per lo spirito di apertura che ha mostrato riguardo ai suoi progetti. Come abbiamo avuto modo di verificare di persona, PETSCII-BBS e la sua implementazione principale RetroAcademy BBS funzionano alla grande e continuano a riscuotere grande interesse presso moltissimi fan del C64/C128.

La sensazione nell'usare hardware reale risalente ai primi anni '80 per connettersi alla Rete e accedere a servizi davvero utili ed efficienti è pazzesca, soprattutto per chi, come il sottoscritto e molti dei nostri appassionati lettori, è nato ed è cresciuto con accanto un home computer "limitato" come il C64. Il framework PETSCII-BBS e le sue implementazioni in forma di BBS e servizi online sono un tipico esempio di come il piccolo/grande 8-bit di casa Commodore sia ancora in auge in mille progetti e soluzioni utili grazie all'esperienza, alla creatività e all'immaginazione di sviluppatori e tecnici di ieri e di oggi.

Sono la dedizione e la passione come quelle dimostrate da Francesco che cerchiamo di portare all'attenzione dei lettori sulle pagine di RMW, gettando luce sui idee e progetti che coniugano solide basi e moderne tecnologie dell'informatica personale. *Just for fun!*

Link e riferimenti bibliografici

- Progetto PETSCII BBS Builder su GitHub: <https://github.com/sblendorio/petscii-bbs>
- Gioco "Gorilla" in Modula-2 per sistemi CP/M: <https://github.com/sblendorio/gorilla-cpm>
- Libreria "XTerm" (cross-terminal) in Modula-2 per sistemi CP/M: <https://github.com/sblendorio/mod-xterm-cpm>
- Libreria "XTerm" (cross-terminal) in C per sistemi CP/M: <https://github.com/sblendorio/c-xterm-cpm>
- Gioco "Queens" in Turbo Pascal per C128 in modalità CP/M: <https://github.com/sblendorio/queens-cpm>
- CBMage: stampare bitmap con una Commodore MPS 803 da PC o Mac: <https://github.com/sblendorio/cbmage>
- CBMtext: stampare testo UTF-8 con una Commodore MPS 803 da PC o Mac: <https://github.com/sblendorio/cbmtext>
- Ultimate Term 2.3 per C64 e C128, di Scott Hutter & Francesco Sblendorio: <https://csdb.dk/release/?id=181558>
- Sorgente di Ultimate Term: <https://github.com/xlar54/ultimateii-dos-lib>
- Scheda KC64Wifi di Pasquale De Luna: <http://www.codingkoala.com/kc64wifi/>
- Scheda KC64Eth di Pasquale De Luna: <http://www.codingkoala.com/kc64eth/>
- Sviluppare in C per Commodore 64 parte 1, di Fabrizio Lodi: <https://www.retroacademy.it/2018/10/23/retrocomputers/sviluppare-in-c-per-il-commodore-64-parte-1/6902/>
- Sviluppare in C per Commodore 64 parte 2, di Fabrizio Lodi: <https://www.retroacademy.it/2018/12/01/retrocomputers/sviluppare-in-c-per-il-commodore-64-parte-2/6933/>
- KipperTerm: <https://csdb.dk/release/index.php?id=174537>
- Specifiche RR-NET: <http://wiki.icomp.de/wiki/RR-Net>
- Progetto droid64: <http://droid64.sourceforge.net/>
- Progetto ZMPP (Z-Machine Preservation Project): <http://zmpp.sourceforge.net/>





FAST Basic – compilare Locomotive Basic in CP/M

di Francesco Fiorentini

Abbiamo visto che il Locomotive Basic è un linguaggio versatile e piuttosto veloce, ma, essendo un linguaggio interpretato, la sua velocità è limitata dalla necessità di tradurre le istruzioni in LM al momento dell'esecuzione. Adesso che abbiamo imparato a scrivere codice in Locomotive Basic, potremmo sentire l'esigenza di mettere il turbo al nostro lavoro o più semplicemente creare un file eseguibile per distribuire i nostri programmi. Come fare?

Possiamo passare a linguaggi compilati... e prima o poi lo faremo oppure possiamo compilare i nostri lavori con un compilatore Basic.

A questo proposito abbiamo provato un compilatore rilasciato nel 1988 da DMV Software (azienda tedesca) che risponde al nome di Fast Basic Compiler. Una particolarità di questo software è che gira sotto ambiente CP/M 2.2 o CP/M 3.0. Ma come facciamo ad eseguire questo programma sull'Amstrad CPC? Seguiteci e ve lo mostreremo.

Prima di cominciare, però, facciamo la lista della spesa di quello che ci occorre.

1. **Fast Basic Compiler.** Possiamo scaricarlo da qui: <https://www.cpc-power.com/index.php?page=detail&onglet=notices&num=4705>, allo stesso indirizzo trovate anche un manuale in tedesco (per chi ha familiarità con questa lingua) ed una breve, ma comoda, guida in inglese (molto più alla mia portata).

2. **CP/M 2.2.** Possiamo scaricarlo da qui: http://www.cpcwiki.eu/index.php/CP/M_2.2

3. Un disco con i nostri **programmi basic.**

4. Un **Amstrad CPC** reale o un emulatore **WinAPE** o **Caprice**. Io non ho un CPC fisico, vado di emulatore WinAPE, scaricabile qui: <http://winape.net/>



Figura 1 - Comando |cpm

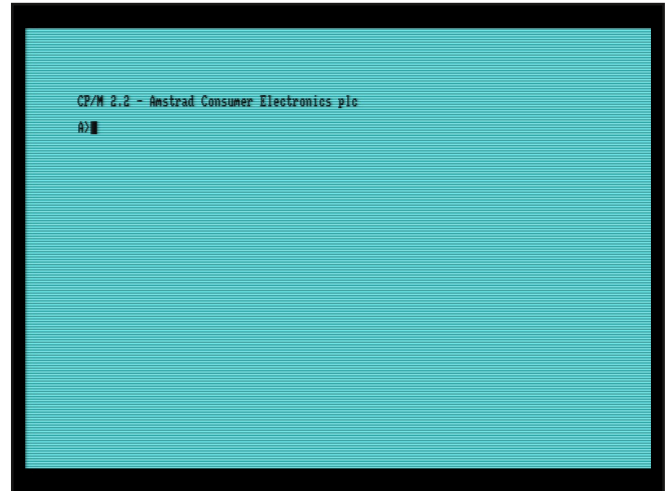


Figura 2 - L'ambiente CP/M 2.2

A questo punto abbiamo tutto l'occorrente per cominciare, ma prima di mettere mano alla procedura fatemi spendere due parole sul CP/M.

Il CP/M è un sistema operativo per micro e personal computer della Digital Research sviluppato nel 1974 per il processore Intel 8080 ma compatibile anche con i processori Zilog Z80. Nella seconda metà degli anni 70 e nella prima metà degli anni 80 era divenuto uno standard *de facto* in ambiente industriale con migliaia di programmi a disposizione. L'Amstrad, con il CPC, che monta un processore Z80, forniva un disk drive CP/M 2.2 in bundle per allargare il parco software a disposizione dell'utente. Bene, dopo la doverosa introduzione, passiamo all'azione.

Carichiamo il CP/M

Inseriamo il disco del CP/M 2.2 nel drive A e digitiamo il comando |cpm (Fig. 1).

Dopo un breve caricamento dovremmo trovarci di fronte al prompt (Fig. 2) del Sistema Operativo CP/M 2.2.

Proviamo a lanciare il comando DIR per visualizzare il contenuto del disco inserito nel drive A, come indicato dal prompt dei comandi: A> (Fig. 3).



Figura 3 - Contenuto del disco CP/M 2.2





```

CP/M 2.2 - Amstrad Consumer Electronics plc

A)dir
A: AMSDOS COM : DISCKIT2 COM : LOGO2 COM : LOGO2 SUB
A: DRKEYS COM : FWRESET COM : SUBMIT COM : RITDEMO BIN
A: ROINTIME DEM : BOOTGEN COM : CLOAD COM : CSAVE COM
A: DDT COM : DUMP COM : ED COM : FILECOPY COM
A: LOAD COM : MOVCPM COM : PTP COM : SETUP COM
A: STAT COM : SVSGEN COM : XSUB COM : DISC BAS
A)dir
A: BC COM : BC DOC : DEMO1 BAS : DEMO2 BAS
A: DEMO3 BAS : DEMO4 BAS : DEMO5 BAS : DEMO6 BAS
A: DEMO1 BAS : DEMO2 BAS : DEMO3 BAS : DISCOM1 BIN
A)

```

Figura 4 - Contenuto del disco Fast Basic

Carichiamo il Fast Basic ed il disco con i programmi Basic. Adesso che l'ambiente CP/M ed i suoi comandi intrinseci sono in memoria, possiamo rimuovere quel disco e caricare il disco del Fast Basic nel drive A.

Una volta fatto, lanciamo di nuovo il comando DIR per vedere il contenuto del disco Fast Basic.

Dovreste vedere una cosa simile alla Fig. 4.

Adesso inseriamo il nostro disco con i programmi in Basic nel drive B e vediamo il contenuto con DIR B:. Ovviamente il contenuto del vostro disco potrebbe essere sensibilmente diverso dal mio.

Compiliamo un programma

Lanciamo il comando "**BC B:SKIFFULL.BAS /LE**" che significa: compila il file SKIFFULL.BAS che si trova nel disco B: (ignoriamo per il momento i parametri /LE).

Ovviamente voi dovrete sostituire SKIFFULL.BAS con il nome del vostro programma Basic.

Una volta eseguito, se il vostro programma non ha routine che il compilatore non è in grado di compilare, dovrete vedere qualcosa simile a Fig. 5.

Giusto per confermare che il nostro programma è stato trasformato in eseguibile (BIN), proviamo a controllare di nuovo il contenuto del disco B: (Fig. 6).

Come vedete, adesso oltre al programma SKIFFULL.BAS, c'è anche l'eseguibile SKIFFULL.BIN.

```

A: AMSDOS COM : DISCKIT2 COM : LOGO2 COM : LOGO2 SUB
A: DRKEYS COM : FWRESET COM : SUBMIT COM : RITDEMO BIN
A: ROINTIME DEM : BOOTGEN COM : CLOAD COM : CSAVE COM
A: DDT COM : DUMP COM : ED COM : FILECOPY COM
A: LOAD COM : MOVCPM COM : PTP COM : SETUP COM
A: STAT COM : SVSGEN COM : XSUB COM : DISC BAS
A)dir
A: BC COM : BC DOC : DEMO1 BAS : DEMO2 BAS
A: DEMO3 BAS : DEMO4 BAS : DEMO5 BAS : DEMO6 BAS
A: DEMO1 BAS : DEMO2 BAS : DEMO3 BAS : DISCOM1 BIN
A)dir B:
B: SKIFF10 BAS : SKIFF BAS : SKIFF $$$ : MDUMP BAS
B: SKIFF10 BIN : SKIFFULL BAS : MDUMP BIN
A)BC B:SKIFFULL.BAS /LE
CPC Basic Compiler Ver 1.03 (c) 1988 by Peter Hoepfner
compiler pass 1
compiler pass 2
End of compilation
no errors found
A)h:
B)dir
B: SKIFF10 BAS : SKIFF BAS : SKIFF $$$ : MDUMP BAS
B: SKIFF10 BIN : SKIFFULL BAS : MDUMP BIN : SKIFFULL BIN
B)

```

Figura 6 - Il nostro programma .BIN

```

CP/M 2.2 - Amstrad Consumer Electronics plc

A)dir
A: AMSDOS COM : DISCKIT2 COM : LOGO2 COM : LOGO2 SUB
A: DRKEYS COM : FWRESET COM : SUBMIT COM : RITDEMO BIN
A: ROINTIME DEM : BOOTGEN COM : CLOAD COM : CSAVE COM
A: DDT COM : DUMP COM : ED COM : FILECOPY COM
A: LOAD COM : MOVCPM COM : PTP COM : SETUP COM
A: STAT COM : SVSGEN COM : XSUB COM : DISC BAS
A)dir
A: BC COM : BC DOC : DEMO1 BAS : DEMO2 BAS
A: DEMO3 BAS : DEMO4 BAS : DEMO5 BAS : DEMO6 BAS
A: DEMO1 BAS : DEMO2 BAS : DEMO3 BAS : DISCOM1 BIN
A)dir B:
B: SKIFF10 BAS : SKIFF BAS : SKIFF $$$ : MDUMP BAS
B: SKIFF10 BIN : SKIFFULL BAS : MDUMP BIN
A)BC B:SKIFFULL.BAS /LE
CPC Basic Compiler Ver 1.03 (c) 1988 by Peter Hoepfner
compiler pass 1
compiler pass 2
End of compilation
no errors found
A)

```

Figura 5 - Compilato con successo!

Torniamo al Locomotive Basic

Inseriamo nuovamente il disco del CP/M nel drive A e lanciamo il comando **AMSDOS** (Fig. 7).

Adesso rimuoviamo il disco contenente i programmi Basic dal drive B, inseriamolo nel drive A e lanciamo il comando CAT (catalog) per vederne il contenuto (Fig. 8).

Lanciamo il programma compilato

Lanciamo il nostro eseguibile con il comando run "SKIFFULL.BIN" e godiamoci la velocità del gioco una volta compilato (Fig. 9 e Fig. 10).

Purtroppo non posso farvi vedere la velocità, ma fidatevi quando vi dico che è circa due volte più veloce della versione interpretata... tanto che adesso è veramente difficile da giocare.

Un po' di informazioni tecniche...

Il compilatore funziona su tutti i modelli Amstrad CPC 464, 664 e 6128 con almeno un disk drive.

La sintassi, come abbiamo visto, è la seguente:

```

BC <filename> /L
BC <filename> /E
BC <filename> /LE

```

```

A: DEMO3 BAS : DEMO4 BAS : DEMO5 BAS : DEMO6 BAS
A: DEMO1 BAS : DEMO2 BAS : DEMO3 BAS : DISCOM1 BIN
A)dir B:
B: SKIFF10 BAS : SKIFF BAS : SKIFF $$$ : MDUMP BAS
B: SKIFF10 BIN : SKIFFULL BAS : MDUMP BIN
A)BC B:SKIFFULL.BAS /LE
CPC Basic Compiler Ver 1.03 (c) 1988 by Peter Hoepfner
compiler pass 1
compiler pass 2
End of compilation
no errors found
A)h:
B)dir
B: SKIFF10 BAS : SKIFF BAS : SKIFF $$$ : MDUMP BAS
B: SKIFF10 BIN : SKIFFULL BAS : MDUMP BIN : SKIFFULL BIN
A)dir
A: AMSDOS COM : DISCKIT2 COM : LOGO2 COM : LOGO2 SUB
A: DRKEYS COM : FWRESET COM : SUBMIT COM : RITDEMO BIN
A: ROINTIME DEM : BOOTGEN COM : CLOAD COM : CSAVE COM
A: DDT COM : DUMP COM : ED COM : FILECOPY COM
A: LOAD COM : MOVCPM COM : PTP COM : SETUP COM
A: STAT COM : SVSGEN COM : XSUB COM : DISC BAS
A)amsdos

```

Figura 7 - Comando AMSDOS





```

ParadOS V1.2+ ©2015 BitWise Systems.
BASIC 1.1
Ready
cat
Drive A: user 0
MDUMP .BAS 2K SKIFF10 .BAS 1K
MDUMP .BIN 1K SKIFF10 .BIN 1K
SKIFF .??? 6K SKIFFFULL.BAS 5K
SKIFF .BAS 6K SKIFFFULL.BIN 14K
128K free
Ready

```

Figura 8 - AMSDOS comando CAT

```

ParadOS V1.2+ ©2015 BitWise Systems.
BASIC 1.1
Ready
cat
Drive A: user 0
MDUMP .BAS 2K SKIFF10 .BAS 1K
MDUMP .BIN 1K SKIFF10 .BIN 1K
SKIFF .??? 6K SKIFFFULL.BAS 5K
SKIFF .BAS 6K SKIFFFULL.BIN 14K
128K free
Ready
run SKIFFFULL.BIN

```

Figura 9 - Eseguiamo il file .BIN

Dove i parametri hanno questa funzionalità:

L - omette il numero di linea nel programma compilato (per salvare spazio)

E - da usare se non servono gli interrupts (AFTER, EVERY, ON SQ)

Tutti i comandi del Locomotive Basic standard sono supportati ad eccezione dei seguenti:

AUTO, CHAIN, CHAIN MERGE, CONT, DELETE, EDIT, LIST, LOAD (BASIC program), MERGE, RENUM, RESUME NEXT, SAVE (BASIC program), TROFF, TRON

Anche le estensioni del Locomotive Basic 1.1 sono supportate!

Attenzione: sembra che la dimensione massima dei programmi compilabili sia attorno ai 17KB. Non ho potuto provarlo, quindi prendete questa informazione con il beneficio di inventario.

Se volete velocizzare ulteriormente i vostri programmi, potete ottimizzare le prestazioni del compilato preferendo nel codice Basic le variabili intere a quelle in virgola mobile (ove possibile ovviamente).

Floating point:

```
10 FOR i=0 TO 10000
20 NEXT i
```

Non compilato 11.1s, compilato 8.2s

Integer:

```
10 FOR i%=0 TO 10000
20 NEXT i%
```

Non compilato 5.9s, compilato 1.1s

A questo punto non avete più scuse...
Attendo i vostri programmi compilati!

```

SKIFF
di Francesco Fiorentini

Livello:
(R)ookie
(E)asy
(M)edium
(H)ard
(I)mpossible

Istruzioni:
Passa tra le porte evita alberi e rocce
Usa le frecce ←→ per spostarti

```

Figura 10 - SKIFF eseguito con successo!!!

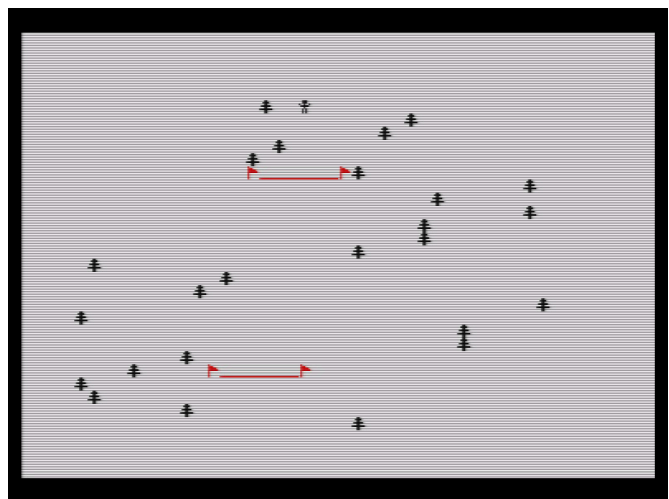


Figura 11 - WOW è velocissimo





Star Watcher

di Francesco Fiorentini

Nel numero scorso abbiamo visto Visible Solar System, un programma di esplorazione spaziale rilasciato nel 1983 dalla Commodore per il C64. In questo numero è la volta di un programma astronomico per Amstrad CPC rilasciato da AMSoft nel 1984.

AMSoft era una società di produzione software controllata direttamente da Amstrad e creata appositamente allo scopo di produrre software per la linea CPC. Quest'approccio contribuì non poco alla fortuna della linea CPC, specialmente in UK, Francia e Spagna. Possiamo quindi affermare che i programmi prodotti e rilasciati da Amsoft erano software ufficiale Amstrad. Il programma che andiamo a recensire si chiama Star Watcher e dimostra tutta l'attenzione che veniva posta prima di rilasciare un pacchetto sotto egida AMSoft.

Il manuale

Ho sempre apprezzato i manuali a corredo del software, soprattutto quando sono ben curati ed esaustivi. Il manuale di Star Watcher è un chiaro esempio della cura con cui venivano rilasciati i software negli anni 80.

Il manuale si compone di più di 50 pagine contenenti dettagliate informazioni riguardo il software, un glossario di termini astronomici e ben 3 appendici:

Appendice 1: catalogo delle costellazioni

Appendice 2: funzioni trigonometriche

Appendice 3: istruzioni per il caricamento del software

Ma non è tutto, il software è strutturato in due moduli ben distinti, il programma Star Watcher stesso ed un modulo educativo, atto a trasmettere all'utente conoscenze basilari di astronomia per poter utilizzare il software al meglio. In questo aspetto il manuale gioca un ruolo fondamentale: le prime 3 dispense del corso di astronomia sono infatti nel manuale ed il computer non è necessario.

Vediamo in dettaglio gli argomenti del corso:

- 1) spiegazione di come il cielo è mappato e le convenzioni utilizzate per descrivere la posizione delle stelle (computer non necessario)
- 2) come le stelle sembrano muoversi in relazione a chi le osserva dalla Terra (computer non necessario)
- 3) denominazione delle stelle (computer non necessario)
- 4) forma e apparenza nel cielo delle costellazioni (richiesto il programma didattico)
- 5) relazione posizionale delle costellazioni (richiesto il programma didattico)

Se come me, siete abbastanza digiuni di nozioni astronomiche, troverete il manuale molto interessante e, non fosse altro per imparare qualcosa di nuovo, vi consiglio di procurarvi il software e leggervi il manuale con attenzione.

Prima di proseguire è però doveroso riportare alcune

Star Watcher - Amstrad CPC

Software e manuali possono essere scaricati da:
https://www.cpcwiki.eu/index.php/Star_Watcher

nozioni minime per apprezzare il programma Star Watcher:
Latitude (Latitudine): è la distanza angolare tra un parallelo e l'Equatore. Il valore della latitudine è compreso tra 0° (Equatore) e 90° (Polo), seguito dall'indicazione Nord (abbreviato N) o Sud (abbreviato S), a seconda che la località si trovi nell'emisfero settentrionale oppure nell'emisfero meridionale.

Longitude (Longitudine): è la distanza angolare tra un meridiano e il meridiano di riferimento, generalmente Greenwich. Il valore è espresso in gradi e seguito da Est (abbreviato E) o Ovest (abbreviato W) a seconda che la misura sia fatta a est o a ovest del meridiano di Greenwich.

Magnitude (Magnitudine): è la misura della luminosità di un corpo celeste da un punto di osservazione (generalmente la Terra). Ideato dal greco Ipparco, contemplava una scala da 1 (stella più luminosa) a 6 (stella meno luminosa) e comprendeva solo oggetti celesti visibili ad occhio nudo. Attualmente, grazie alle moderne tecnologie, la scala è stata allargata enormemente. Il telescopio spaziale Hubble può infatti osservare oggetti fino a luminosità 30.

Celestial Poles (Poli Celesti): Nord e Sud; sono i punti nel cielo immediatamente sopra i poli terrestri.

Celestial Equator (Equatore Celeste): porzione di cielo direttamente sopra l'Equatore.

RA Right Ascension (Ascensione Retta): è la longitudine proiettata sulla sfera celeste anziché sulla superficie terrestre. L'ascensione retta è misurata in ore, minuti e

2. Sky Co-ordinates

(Computer not required)

2.1 Down to Earth

The Earth is almost a sphere, spinning in space on an axis which passes through the north and south poles. On the surface of the earth, the location of any point can be defined by its **LATITUDE** and **LONGITUDE**, a pair of measurements which are related to the position of the poles. To define longitude, it is best to look down on the north pole. Imagine a straight line drawn on the earth's surface from the north to south pole. You will find that whichever direction you start out in, all straight lines from the north pole lead to the south pole. Early navigators selected the particular north pole to south pole line which passes through Greenwich in London, and called it the **PRIME** or **GREENWICH** meridian. From this line, you can measure the angle at the north pole to any other line from the north to south pole, and this angle is called the **LONGITUDE** of the line. Obviously, the longitude of the Prime Meridian is zero. The angle to any other line can be measured in a clockwise or anticlockwise direction. The clockwise direction is **WEST** and anticlockwise is **EAST**. By convention, the angle is measured in the direction which makes it less than 180 degrees.

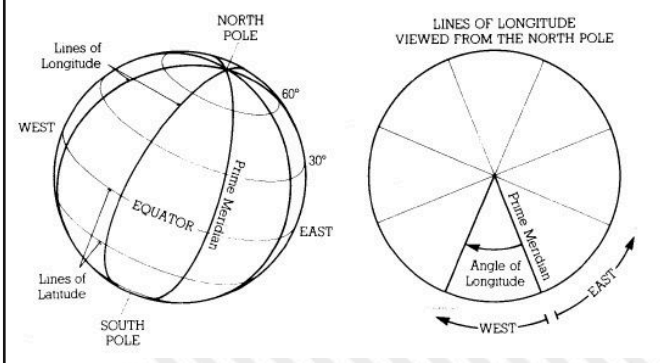
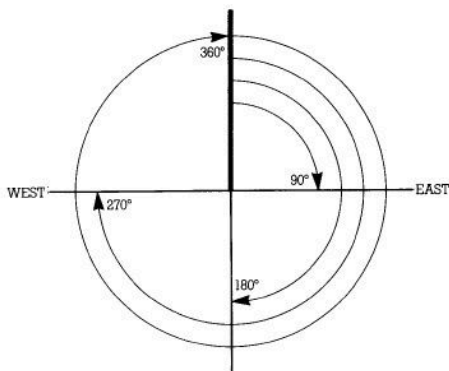


Figura 1 - Estratto del manuale



**MEASURING ANGLES**

The angle between two intersecting lines is measured in **DEGREES**, and the symbol ° is used to denote them. A degree is defined as 1/360th part of a circle; that is, the angle measured all the way around a circle is 360°. Hence a **RIGHT ANGLE**, or quarter circle has an angle of 90°, whilst the angle around half a circle is 180°.



For more accurate measurements, it may be necessary to define an angle in terms of less than one degree. For this purpose, one degree can be subdivided into 60 **MINUTES**, and the symbol ' is used to denote them. Hence half a degree can be written as 30', and four and a quarter degrees is 4° 15'.

Figura 2 - Ancora dal manuale...

secondi, corrispondenti alla rotazione terrestre: 24 ore di ascensione retta sono un giro completo.

Dec Declination (Declinazione): è la latitudine proiettata sulla sfera celeste anziché sulla superficie terrestre. I punti a nord dell'equatore celeste hanno declinazione positiva, quelli a sud, negativa.

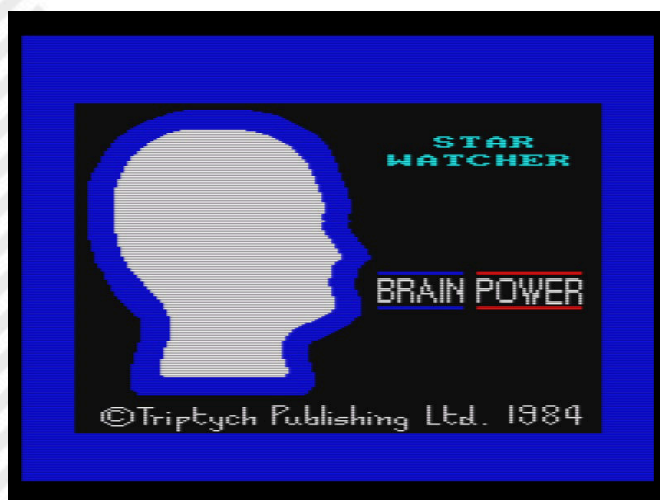
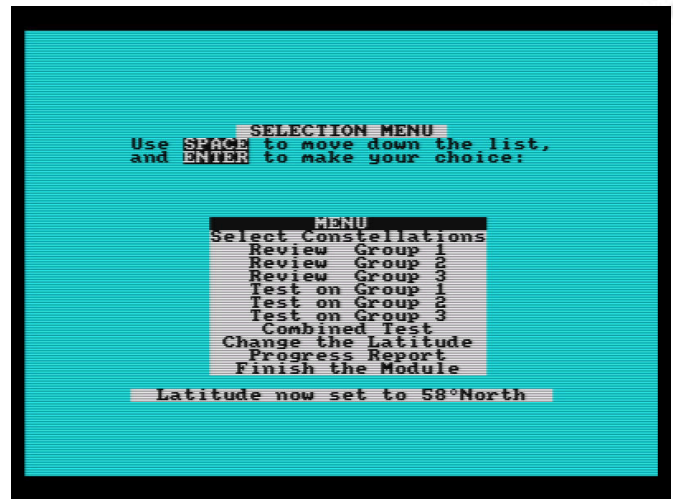
Az Azimuth: è la misura dell'angolo rispetto a Nord della proiezione di un oggetto astrale sull'orizzonte dell'osservatore.

Queste informazioni sono tutte riportate nel manuale.

Il modulo educativo

Lanciando il comando 'run "teach [enter]' è possibile avviare il modulo educativo, modulo che vi ricordo va ad integrare i primi 3 capitoli del manuale. Dopo uno splash screen che ci ricorda che questo software fa parte di un pacchetto più ampio chiamato Brain Power, verremo accolti da una schermata che ci richiede di inserire la nostra latitudine.

Il motivo è semplice, al fine di aiutarci ad apprendere il nome delle costellazioni e la loro posizione nel cielo, il software, richiedendoci la nostra posizione, è in grado di

**Figura 3 - Software del pacchetto Brain Power****Figura 4 - Menù del modulo educativo**

disegnare le costellazioni come apparirebbero sopra di noi, con tanto di altre stelle a contorno.

Lo scopo del programma educativo è quello di aiutarci ad identificare le costellazioni tramite la loro forma (modulo 4) e tramite la loro forma e posizione nel cielo rispetto alle altre stelle (modulo 5).

**Figura 5 - Orsa Minore senza altre stelle****Figura 6 - Orsa Minore con le altre stelle**

Se il primo modulo è relativamente semplice, non si può dire assolutamente lo stesso del secondo. Per aiutarci in questo percorso di apprendimento potremo osservare le costellazioni una alla volta oppure divise in 3 gruppi.



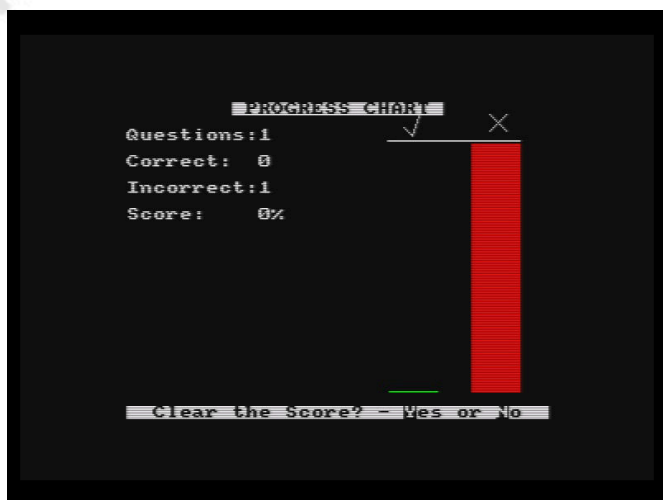


Figura 7 - Il report dei progressi (pochi...)

Alla fine del percorso di apprendimento potremo testare la nostra capacita' tramite alcuni test. Interessante anche il report dei risultati. Una volta soddisfatti del nostro percorso educativo, potremo terminare il modulo e passare al programma vero e proprio.

Star Watcher

Lanciando il comando 'run "apply [enter]' e' possibile avviare il programma Star Watcher (e finalmente direte voi...) che ci accoglierà di nuovo con una richiesta di informazioni, questa volta più dettagliate rispetto al modulo educativo.

Dovremo infatti inserire le seguenti informazioni:

- Data odierna
- Ora corrente locale
- Ora locale al mezzogiorno di Greenwich (noon GMT)
- La nostra latitudine
- La nostra longitudine

Una volta inserite queste informazioni potremo finalmente accedere al programma.



Figura 8 - Forniamo le nostre coordinate

Ci aspetteremmo di essere accolti da una bella schermata con un cielo stellato... e invece no!

La prima schermata è inesorabilmente vuota, ma contiene già tutto il pannello di controllo, diviso in 6 aree per gestire il programma.



Figura 9 - Dettaglio della finestra principale

Ecco il dettaglio della schermata principale (Fig. 9):

- 1) REAL TIME CLOCK: mostra continuamente l'ora e la data correnti in base ai dati prima forniti.
- 2) DATA DISPLAY: questa è l'area dove il computer mostra le informazioni di ciò che è visibile nella SKY WINDOW e dove puoi definire cosa visualizzare.
- 3) SKY WINDOW: la finestra dove vengono mostrate le stelle. La porzione di cielo è larga 90° gradi ed alta 45° gradi. Il colore della finestra varia a seconda della situazione. Di notte il cielo è nero e le stelle sono bianche. Di giorno il cielo sarà blu e le stelle nere per ricordarci che non sono visibili.
- 4) MESSAGE LINE: il menù dove scegliere le azioni. Puoi spostarti tra le azioni con lo [SPACE] e confermarle con [ENTER].
- 5) STATUS DISPLAY: informazioni utili per l'utente.
- 6) DIRECTION INDICATORS: suddivisa in 2 parti, la parte superiore indica la direzione dove stiamo guardando, la parte inferiore l'angolo in altezza dall'orizzonte.

Disegniamo le stelle

Beh, a questo punto direi che siamo pronti per disegnare il nostro primo cielo stellato! Come suggerisce il manuale andiamo a vedere una porzione di cielo pre-impostata. Utilizzate la barra spaziatrice fino a visualizzare in (4) la voce 'Horizontal Plot' (abbreviazione di disegna lo schermo utilizzando le coordinate dell'orizzonte) e poi premete [ENTER] per confermare. Premete nuovamente [ENTER] alla richiesta del numero di passaggi 'Number os Steps'



Figura 10 - Horizon Plot





Figura 11 - Il 'nostro' cielo, celeste perchè di giorno

e dopo una breve attesa vedrete la schermata in figura 10.

Disegniamo le 'nostre' stelle

Adesso siamo pronti per disegnare il cielo della nostra posizione. Scorriamo quindi fino a trovare 'Starting Data' e dopo aver premuto [ENTER] andiamo ad inserire i nostri parametri in (2). Quando saremo soddisfatti dei dati inseriti utilizzeremo la barra spaziatrice fino a visualizzare in (4) la voce 'Horizontal Plot' e dopo la richiesta del numero di passaggi ed una breve attesa vedremo la schermata in figura 11. Come anticipato, visto che localmente è pieno giorno, il cielo è colorato di celeste e le stelle sono nere.

Provate inoltre ad a scegliere l'opzione 'Complete View' per aumentare il numero di stelle presenti nella nostra porzione di cielo; figura 12.

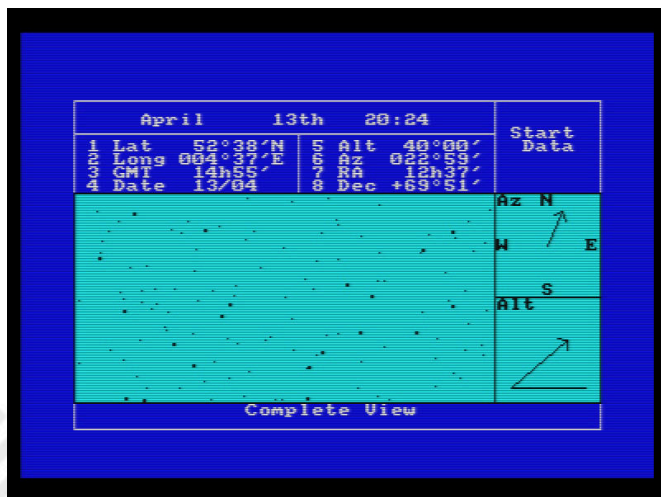


Figura 12 - Il 'nostro' cielo con tutte le stelle

Selezione della magnitudine delle stelle

Provate a questo punto a selezionare l'opzione 'Star Selection'. Dopo un breve caricamento verrà mostrata la schermata in fig. x. Da qui possiamo scegliere per ogni costellazione quale stelle visualizzare. Spostandoci con i tasti cursore sulla costellazione desiderata basterà premere i numeri 0, 1 o 2 per le seguenti opzioni:

- 0) Nessuna stella della costellazione verrà disegnata
 - 1) Solo le stelle che caratterizzano la forma della costellazione verranno disegnate
 - 2) Tutte le stelle della costellazione verranno disegnate
- Possiamo anche applicare la scelta a tutte le costellazioni contemporaneamente utilizzando il campo 'ALL'.

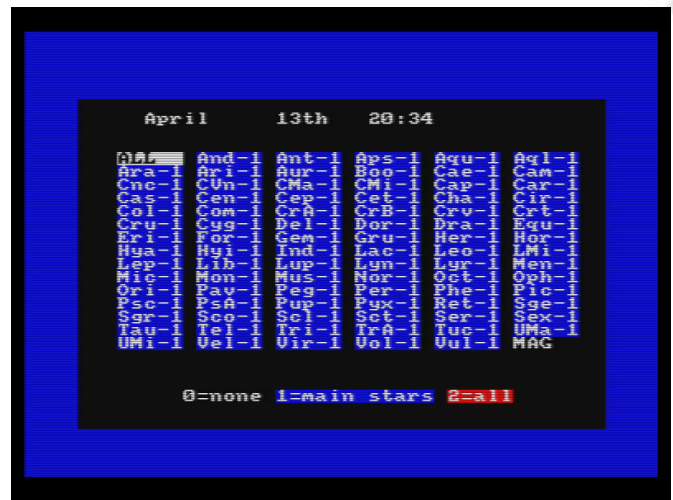


Figura 13 - Selezione della magnitudine

Oppure scegliere una magnitudine minima utilizzando il campo 'MAG' (Fig. 13).

Altre funzionalità

Quanto appena descritto è soltanto una parte delle funzionalità del programma che potremmo riepilogare in questa tabella come riportato anche nel manuale:

Complete View: ridisegna la vista corrente con tutte le costellazioni in modo 2 ma soggette al limite di magnitudine correntemente settato.

Continuous Plot: disegna il cielo con un intervallo di 1 minuto utilizzando i parametri forniti in Data Display.

Copy to Printer: invia lo schermo attuale alla stampante

Cursor Mode: attiva un cursore e ridisegna lo schermo centrandolo alla posizione del cursore una volta premuto [ENTER].

Equatorial Plot: disegna il cielo con RA e Dec forniti in Data Display.

Finishing Data: edita Data Display per un disegno finale in multi step view.

Horizon Plot: disegna il cielo con Alt e Az forniti in Data Display.

Real Time Plot: disegna il cielo con un intervallo di 1 minuto utilizzando latitudine e longitudine forniti inizialmente.

Star Selection: permette di scegliere quali costellazioni disegnare e un livello minimo di magnitudine da utilizzare.

Starting Data: edita i data nel Data Display.

Conclusioni

Il software è molto rapido nel calcolare la posizione da disegnare, reperire le stelle nel database e disegnarle nella parte dello schermo visibile. Non sono in grado di misurarne l'affidabilità, ma posso evidenziare alcune mancanze.

E' vero che il programma si chiama Star Watcher, ma sarebbe stato utile includere anche i pianeti del sistema solare. Un'altra cosa di cui, da inesperto, sento la mancanza, è la connessione tra le varie stelle che formano le costellazioni. Un'opzione per attivarla/disattivarla nella finestra principale avrebbe aiutato molto. Per non parlare della mancanza di una qualsiasi informazione nella finestra principale. Vedere il nome di una costellazione accanto alla sua raffigurazione avrebbe aiutato molto i neofiti. Ovviamente il software era destinato ad astronomi già rodati o a convinti principianti. L'utilizzatore casuale avrebbe trovato il software poco convincente.





Giocare a vite infinite sul C64 - "La sfida continua"

di Marco Pistorio

In questo articolo mi riferirò diverse volte all'articolo "Vite infinite con il Commodore 64", che venne pubblicato all'interno del numero 3 di RetroMagazine. Potete eventualmente scaricare questo numero dal nostro sito ufficiale qualora non fosse già in vostro possesso.

Introduzione

Dopo diversi anni, sono riuscito a giocare nuovamente con un titolo per C64 che trovo molto interessante nei miei "anni verdi". Si tratta precisamente di "PastFinder", che conobbi con il titolo "Rifornimenti spaziali", contenuto all'interno della cassetta "Top Playgames" numero 13, che uscì nelle edicole durante il mese di settembre del 1986.

Il gioco si distingue, a mio avviso, per un discreto gameplay, per le mappe stimolanti in cui serve una mano ben ferma ed ottimi riflessi per procedere attraverso i vari scenari e livelli che si fanno via via più ostici. Ogni qualvolta si riesce a completare una certa area del gioco, se ne scopre una un po' più ampia. Da allora nacque il desiderio in me di riuscire a vedere tutta la mappa di gioco, compito assolutamente non facile senza trucchetti, magari impiegando trainers realizzati da qualcuno dei tanti gruppi di cracking.

A dire il vero non ho valutato molto questa possibilità anche perchè, se posso, preferisco fare da me. Così ora colgo questa occasione per raccontare anche a voi, amici lettori, questa piccola avventura, cercando di fornire anche a voi gli strumenti e le conoscenze minimali per riuscire ad ottenere simili risultati!

Nello scorso articolo vi presentai una tecnica per riuscire a giocare con vite infinite ai vostri giochi preferiti. Tale tecnica funziona grossomodo in questi termini:

- a) lancio il gioco all'interno di VICE
- b) faccio delle "istantanee" della RAM ogni qualvolta perdo una vita durante il gioco
- c) esamino queste "istantanee" con un tool o un programma simile che mi aiuti a comprendere quale locazione di memoria contenga l'informazione delle vite a disposizione
- d) intervengo quindi sul codice del gioco per evitare che il numero delle vite contenute nella locazione di memoria che ho individuato diminuisca.

Vi consiglio caldamente di rileggere con attenzione

quel precedente mio articolo che vi ho segnalato, soprattutto se qualcosa fino a questo momento non dovesse risultarvi chiara.

Nel frattempo sono trascorsi oltre due anni da quell'articolo ed ho riscritto il tool in Visual Basic.Net. Potete scaricare il tool insieme al relativo codice sorgente sfruttando i link nel riquadro presente alla fine di questo articolo.

Descrizione della attività

Mettiamoci adesso all'opera e creiamo due o tre "istantanee" della RAM del C64, istantanee che creeremo quando avremo a disposizione inizialmente 4 vite, poi 3 ed infine 2 vite.

Come crearle? In modo molto semplice, aprendo il monitor del VICE e digitando la seguente stringa:

s "img1.mem" 0 0000 FFFF (invio)

avendo l'accortezza di cambiare il nome del file partendo da img1.mem e proseguendo via via con img2.mem, img3.mem e così via.

Questa istruzione è stata già illustrata nel precedente articolo sul numero 3 di RetroMagazine, ma vi spiegherò comunque brevemente di cosa si tratti, sebbene sia anche facilmente intuibile. In pratica salviamo sul nostro hard disk, all'interno di un certo file (img1.mem nel caso in esempio), nel file system (ecco il significato dello 0 subito dopo il nome del file), copiando il contenuto della RAM compresa tra gli indirizzi \$0000 ed \$FFFF in notazione esadecimale, che corrispondono all'intervallo 0-65535 in notazione decimale.

A questo punto interviene il nostro tool. Dopo aver copiato questi files .mem all'interno della stessa cartella dell'eseguibile del tool, potrete lanciarlo. Se avete seguito scrupolosamente le istruzioni fin qui fornite, potete impostare nella prima casella di testo, destinata a contenere il nome del primo file da confrontare, il seguente nome di file: **img1.mem**
Nella seconda casella di testo andrà invece: **img2.mem**

Impostate infine un nome di file per contenere il risultato del confronto, suggerisco un semplice: **out.txt**





```

out.txt - Blocco note di Windows
File Modifica Formato Visualizza ?
img1.mem:img2.mem
(0008) Valori:1,0
(0069) Valori:5,4
(0094) Valori:1,0
(00FD) Valori:4,3
(4118) Valori:61,60
(4119) Valori:61,60
(411A) Valori:61,60
(411B) Valori:61,60
(411C) Valori:61,60
(411D) Valori:61,60
(411E) Valori:61,60
(411F) Valori:61,60
(4120) Valori:61,60
(4121) Valori:61,60
(4122) Valori:61,60
(4123) Valori:61,60
(4124) Valori:61,60
(4125) Valori:61,60
(4126) Valori:61,60
(4127) Valori:61,60

```

Figura 1

Fate click su **Start** per far partire il confronto ed esaminate il risultato della elaborazione (vedi figura 1). Il risultato ci dice che vi sono diverse locazioni di memoria il cui contenuto differisce di 1 unità. La nostra attenzione dovrebbe focalizzarsi sulla locazione **00FD**.

Perchè? Perchè la prima istantanea è stata fatta quando avevamo **4** vite, la seconda quando ne avevamo **3**. I due valori quindi corrispondono perfettamente.

Se avessimo a disposizione una terza istantanea, generata quando le vite durante il gioco fossero diventate **2**, potremmo lanciare nuovamente il tool e verificare se il dato continua ad essere corretto.

Se abbiamo a disposizione questa terza istantanea, possiamo procedere così.

Nella prima casella di testo, metteremo stavolta **img2.mem**. Nella seconda **img3.mem**. Come file risultato opterei per **out2.txt**.

Verifichiamo nuovamente (vedi figura 2)

Notiamo ancora, all'interno della locazione di memoria **00FD**, i valori che ci aspettavamo, ovvero **3** e **2**. L'accordo risulta quindi perfetto ancora una volta e possiamo pertanto considerare la nostra ipotesi di partenza corretta.

La strada adesso si fa un pò più in salita. Dobbiamo individuare la zona della RAM dove il contenuto di questa istruzione viene decrementato. Teniamo presente inoltre che l'indirizzo **00FD** si trova in pagina zero. Analizzando la tabella alla fine del precedente articolo sul numero 3 di RetroMagazine scopriamo che

```

out2.txt - Blocco note di Windows
File Modifica Formato Visualizza ?
img2.mem:img3.mem
(0008) Valori:0,1
(0066) Valori:3,4
(00B0) Valori:6,5
(00FD) Valori:3,2
(104F) Valori:18,17
(105F) Valori:16,15
(4050) Valori:62,63
(4051) Valori:62,63
(4052) Valori:62,63
(4053) Valori:62,63
(4054) Valori:62,63
(4055) Valori:62,63
(4056) Valori:62,63
(4057) Valori:62,63
(4058) Valori:62,63
(4059) Valori:62,63
(405A) Valori:62,63
(405B) Valori:62,63
(405C) Valori:62,63
(405D) Valori:62,63

```

Figura 2

```

VICE monitor - [Monitor]
File Debug View Window
(C:$727c) h 0000 ffff c6 fd
767f
76fe
87d0
(C:$727c)

```

Figura 3

l'istruzione che cerchiamo potrebbe essere una istruzione del tipo : **dec \$FD** ovvero **C6 FD** in codice macchina (notazione esadecimale). Anche stavolta una semplice istruzione di ricerca all'interno del monitor di VICE ci potrà fornire ulteriori e preziosi indizi utili. A tale scopo, digitiamo quindi:

h 0000 FFFF C6 FD (invio)

(vedi figura 3) ed otteniamo tre indirizzi dove potrebbe avvenire il decremento della locazione contenente le nostre vite.

Se non vogliamo studiare con attenzione porzioni di codice assembly, possiamo provare semplicemente a riempire con 2 opcode NOP (EA) i due bytes in corrispondenza di ciascuno dei tre indirizzi che la ricerca ci ha fornito.

Così, sempre da monitor di VICE, digitiamo

a 767f (invio)
nop (invio)
nop (invio)
(invio)

```

VICE monitor - [Monitor]
File Debug View Window
(C:$727c) h 0000 ffff c6 fd
767f
76fe
87d0
(C:$727c) a 767f
.767f nop
.7680 nop
.7681
(C:$7681) ■

```

Figura 4

come da figura 4 e proviamo a riprendere il gioco, uscendo dal monitor di VICE. Non otteniamo alcun risultato palpabile, quindi procediamo allo stesso modo, partendo però dalla seconda istruzione segnalata da VICE.





Operiamo quindi così:

```
a 76FE (invio)
nop (invio)
nop (invio)
(invio)
```

e giochiamo nuovamente.

Stavolta accade un fatto alquanto strano e certamente indesiderato. Appena perdiamo una vita il gioco si conclude.

Qui è la logica che deve guidarci, insieme ad una infarinatura di assembly 6502/6510 che male non fa di certo.

Ciò che accade significa che in quella zona di memoria cambia qualcosa, proprio quando perdiamo una vita. Osserviamo il codice assembly di quella zona in maniera più precisa.

Considero quindi un'area di memoria che inizia qualche byte prima (**76F4**) e termina qualche byte dopo (**7706**) e proviamo a disassemblare tale area entrando nuovamente nel monitor di VICE e digitando:

```
d 76F4 7706 (invio)
```

Otteniamo il risultato che vedete rappresentata dalla figura 5.

Tra le due istruzioni NOP, aggiunte da noi al posto della istruzione che avrebbe decrementato la locazione di memoria contenente le vite, vediamo due istruzioni BPL (Branch if PLus ovvero "Salta se positivo"). Seguendo una linea di pensiero squisitamente logica, potremmo immaginare che, durante lo svolgimento del gioco, alla perdita di una vita, il controllo sarebbe passato alla istruzione **DEC \$FD** (che abbiamo rimosso mettendo al suo posto due istruzioni **NOP** che non fanno assolutamente nulla) e poi, se il numero contenuto in tale locazione fosse rimasto positivo, sarebbe stato effettuato il salto, come prevederebbe l'istruzione **BPL \$7717**. E' probabile che, intervenendo in quella porzione di codice, l'informazione nell'accumulatore non sia più attendibile. Anzi ora è "sporca", come si dice in gergo tecnico.

```
VICE monitor - [Monitor]
File Debug View Window
(C:.$7708)
(C:.$7708)
(C:.$7708)
(C:.$7708)
(C:.$7708)
(C:.$7708)
(C:.$7708)
(C:.$7708)
(C:.$7708) d 76f4 7706
.C:76f4 A2 07 LDX #$07
.C:76f6 A9 8E LDA #$8E
.C:76f8 9D 56 43 STA $4356.X
.C:76fb CA DEX
.C:76fc 10 FA BPL $76F8
.C:76fe EA NOP
.C:76ff EA NOP
.C:7700 10 15 BPL $7717
.C:7702 E6 FD INC $FD
.C:7704 A5 95 LDA $95
.C:7706 05 02 ORA $02
(C:.$7708) █
```

Figura 5

Che si fa allora? Impostiamo senza alcuna condizione un salto a quella locazione. Come?

```
VICE monitor - [Monitor]
File Debug View Window
(C:.$7708)
(C:.$7708)
(C:.$7708)
(C:.$7708)
(C:.$7708)
(C:.$7708)
(C:.$7708) d 76f4 7706
.C:76f4 A2 07 LDX #$07
.C:76f6 A9 8E LDA #$8E
.C:76f8 9D 56 43 STA $4356.X
.C:76fb CA DEX
.C:76fc 10 FA BPL $76F8
.C:76fe EA NOP
.C:76ff EA NOP
.C:7700 10 15 BPL $7717
.C:7702 E6 FD INC $FD
.C:7704 A5 95 LDA $95
.C:7706 05 02 ORA $02
(C:.$7708) a 76fe JMP $7717
7701
(C:.$7701)
```

Figura 6

Entriamo nel monitor di VICE e digitiamo:

```
a 76FE JMP $7717 (invio)
(invio)
```

come da figura 6. A questo punto chiudiamo il monitor di VICE e riprendiamo a giocare. Come per magia, non perderemo più vite durante il gioco!

Per riassumere quindi, per giocare a vite infinite con questo gioco, basterà aprirlo con VICE, poi aprire il monitor di VICE e digitare:

```
a 76FE JMP $7717 (invio)
(invio)
```

quindi possiamo uscire dal monitor di VICE per giocare regolarmente non perdendo alcuna vita. Il "crack" è servito, adoperando **UNA SOLA** istruzione assembly, composta in tutto da 3 bytes.

Conclusioni

Questo è uno di quei casi, di cui vi accennavo nel numero 3 di RetroMagazine, dove, nonostante la tecnica vi permetta di individuare la locazione di memoria contenente il numero delle vite, è necessaria una conoscenza (anche minima) dell'assembly per riuscire ad ottenere il risultato sperato.

Spero che questo approfondimento, chiamiamolo "ripasso", abbia acceso il vostro interesse e possa esservi di stimolo per sperimentare per conto vostro queste idee con i vostri videogiochi preferiti. Nel contempo, spero che questo articolo vi abbia fatto riflettere sulla importanza di conoscere l'assembly affinché queste "magie" diventino più alla vostra portata, qualora non lo fossero già.

Un saluto a tutti voi, amici lettori ed alla prossima!

Materiale a corredo dell'articolo:

http://www.retromagazine.net/download/TOOL_Confronta_immagini_memoria_vice.zip





RetroMath: Codici segreti

di Giuseppe Fedele

Il problema di trasmettere messaggi segreti è antico come la scrittura. La crittografia (dal greco *kryptos* “nascosto” e *graphia* “scrittura”) studia i metodi per cifrare un messaggio in modo che non sia comprensibile a persone non autorizzate a leggerlo.

Il più antico esempio di crittografia risale a 4500 anni fa circa nel periodo dell'Antico Regno in Egitto ed è relativo ad alcuni geroglifici scolpiti in antichi monumenti. Plutarco (46-48 d.C. – 125-127 d.C.), nella Vita di Lisandro, descrive l'uso della scitola (Figura 1), un rudimentale sistema crittografico che veniva sfruttato dagli spartani, in particolare in tempo di guerra, per brevi comunicazioni. La scitola era un piccolo bastone di legno con avvolto un laccio di cuoio su cui era inciso un messaggio che poteva essere decifrato solo avendo un bastone del corretto diametro di quello con il quale era stato scritto.



Fig.1 - Scitola spartana

Si sa che Giulio Cesare (II sec. d.C.) riceveva informazioni e trasmetteva ordini ai suoi luogotenenti mediante un sistema di codifica rudimentale consistente nella traslazione (traslitterazione) delle lettere del messaggio, noto come Codice Cesare. In questo codice ogni lettera viene sostituita da quella che la segue dopo n posti nell'alfabeto, immaginato come una sequenza circolare, per cui dopo Z ritorna A. Fissiamo un numero da 0 a 25 (alfabeto inglese) che sarà la chiave segreta K . L'operazione di cifratura tramite il Codice di Cesare consiste nel sommare K ad ogni carattere del messaggio in chiaro: il crittogramma, cioè, si ottiene spostando in avanti di K posti ogni carattere del messaggio in chiaro. Se per esempio poniamo $K=3$ allora il messaggio

$m = \text{'attaccheremo domani'}$

diventa

$c = \text{'dwwdffkhuHpr grpdql'}$

Il Codice Cesare è collegato all'aritmetica modulare. Siano a, b due interi e n un intero positivo. Allora si dice che a e b sono congruenti modulo n $a \equiv b \pmod{n}$ se la differenza $a-b$ è un multiplo di n . Si può facilmente osservare che valgono le seguenti proprietà

$$b \equiv a \pmod{n}$$

$$a - b \equiv 0 \pmod{n}$$

Allora, nel Codice Cesare, la posizione della lettera cifrata corrisponde alla formula

$$c(x) = x + K \pmod{n}$$

dove x è la posizione della lettera che si vuole cifrare, $c(x)$ la posizione della lettera cifrata, K la chiave ed n il numero di caratteri dell'alfabeto. Il messaggio 'DAZIO', con chiave $K=3$, verrà quindi cifrato in

- 'D', $x = 3$, $c(x) = 6 \pmod{26} = 6$, che corrisponde a 'G'
- 'A', $x = 0$, $c(x) = 3 \pmod{26} = 3$, che corrisponde a 'D'
- 'Z', $x = 25$, $c(x) = 28 \pmod{26} = 2$, che corrisponde a 'C'
- 'I', $x = 8$, $c(x) = 11 \pmod{26} = 11$, che corrisponde a 'L'
- 'O', $x = 14$, $c(x) = 17 \pmod{26} = 17$, che corrisponde a 'R'.

La decifrazione del messaggio avviene attraverso la formula inversa

$$c^{-1}(x) = x - K \pmod{n}$$

I possibili cifrari di Cesare nella lingua inglese sono 25, dato che se una lettera si sposta di 25 posizioni ritorna al punto di partenza, facendo così coincidere il messaggio cifrato con quello in chiaro e rendendo inutile l'operazione. Visto il numero esiguo delle possibilità, l'avversario che volesse forzare il codice potrebbe provare a decifrare il messaggio con tutte le possibili chiavi da $K=1$ a $K=25$ sperando di imbattersi in un messaggio di senso compiuto. Anche i cifrari più elaborati basati sulla corrispondenza fra lettere dell'alfabeto, anche se variate periodicamente, sono vulnerabili poiché, quando si usa la stessa chiave, lettere uguali sono codificate con lettere uguali. E' quindi possibile indovinare le lettere più frequentemente usate in una data lingua e risalire al messaggio in chiaro. Si pensi alla macchina Enigma (Figura 2) usata da tedeschi durante la seconda guerra mondiale. Sfruttando la frequenza statistica delle lettere alfabetiche della lingua tedesca, il gruppo di scienziati inglesi (matematici, linguisti e crittografi) guidati da Alan Turing (Figura 3) riuscì a forzare l'algoritmo di crittografia.



Fig.2 - Macchina cifrante Enigma

Le macchine Enigma nella versione per l'esercito avevano, inizialmente, tre rotori che potevano essere estratti e cambiati. Il primo compito per un operatore di Enigma era di decidere in quale posizione andava impostato ogni singolo rotore. C'erano cinque rotori tra cui scegliere e che potevano essere inseriti nei tre alloggiamenti di Enigma. La

essere inseriti nei tre alloggiamenti di Enigma. La





configurazione delle macchine tedesche veniva cambiata ogni 24 ore seguendo un determinato protocollo da tenere in totale segretezza in quanto, se gli alleati ne fossero venuti in possesso, avrebbero potuto decifrare facilmente ogni messaggio. L'unico modo per decifrare il crittogramma era quindi quello di possedere una macchina Enigma configurata esattamente come quella con cui si era cifrato il messaggio.

A differenza dei cifrari descritti nei quali si sostituisce un

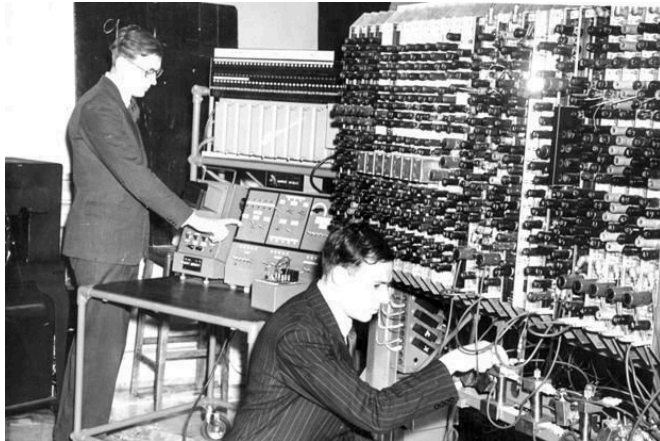


Fig.3 - Alan Turing

carattere con un altro, nel 1929 il matematico Lester S. Hill ideò un algoritmo di cifratura basato su aritmetica modulare e algebra lineare. L'idea è quella di dividere il testo in chiaro in gruppi di due lettere associando ad ogni lettera la posizione corrispondente da 0 a 25. Il messaggio "CIFRARIO DI HILL" verrà quindi composto in

CI	FR	AR	IO	DI	HI	LL
2,8	5,17	0,17	8,14	3,4	7,8	11,11

Per cifrare il messaggio, utilizziamo una matrice

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

con determinante pari ad 1, cioè $ad-bc=1$. In questo caso la matrice inversa è

$$M^{-1} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Quindi se scegliamo come $M = \begin{pmatrix} 3 & 5 \\ 1 & 2 \end{pmatrix}$

la sua inversa è: $M^{-1} = \begin{pmatrix} 2 & -5 \\ -1 & 3 \end{pmatrix}$

La matrice scelta e la sua inversa rappresentano rispettivamente la chiave di codifica e di decodifica.

Per ogni coppia di lettere/cifre si moltiplica la matrice di codifica per la coppia di numeri associata e il risultato viene traslato nell'intervallo 0-25 attraverso le operazioni di modulo; per la coppia "CI" si avrà

$$\begin{pmatrix} 3 & 5 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 2 \\ 8 \end{pmatrix} = \begin{pmatrix} 46 \\ 18 \end{pmatrix} \pmod{26} = \begin{pmatrix} 20 \\ 18 \end{pmatrix}$$

La coppia "CI" è quindi codificata in "US". Analogamente la coppia "FR" viene codificata in "WN" poiché

$$\begin{pmatrix} 3 & 5 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 5 \\ 17 \end{pmatrix} = \begin{pmatrix} 100 \\ 39 \end{pmatrix} \pmod{26} = \begin{pmatrix} 22 \\ 13 \end{pmatrix}$$

Per decifrare il messaggio si procede allo stesso modo considerando la matrice di decodifica. La coppia cifrata "US" verrà decifrata come

$$\begin{pmatrix} 2 & -5 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 20 \\ 18 \end{pmatrix} = \begin{pmatrix} -50 \\ 34 \end{pmatrix} \pmod{26} = \begin{pmatrix} 2 \\ 8 \end{pmatrix}$$

mentre la coppia "WN"

$$\begin{pmatrix} 2 & -5 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 22 \\ 13 \end{pmatrix} = \begin{pmatrix} -21 \\ 17 \end{pmatrix} \pmod{26} = \begin{pmatrix} 5 \\ 17 \end{pmatrix}$$

Il problema di tutti questi metodi di cifratura/decifratura è che se A vuole mandare un messaggio a B, dovrà preliminarmente comunicare a B quale chiave segreta utilizzare. Per cui anche la chiave dovrebbe essere trasmessa in modo che non sia intercettata da malintenzionati.

Per ovviare a questo inconveniente, nel 1978 tre professori: Ronald Rivest, Adi Shamir e Leonard Adleman del MIT realizzarono una procedura di calcoli matematici che prenderà il nome di "algoritmo RSA", dalle iniziali dei suoi inventori. L'algoritmo RSA si basa su alcune proprietà dei numeri primi (cioè i numeri interi positivi maggiori di 1 divisibili solo per 1 e per se stesso; esempio di numeri primi sono: 2,3,5,7,11,13,17,19,23,29,...). Una proprietà è che, dati due numeri primi, è molto facile stabilire il loro prodotto, mentre è molto più difficile determinare, a partire da un determinato numero, quali numeri primi hanno prodotto quel risultato dopo essere stati moltiplicati tra loro.

Come funziona RSA? Facciamo un semplice esempio

Giuseppe deve inviare a Marco un testo (codificato ad esempio tramite la posizione di ogni lettera nell'alfabeto: 0-25).

1. Marco genera due numeri primi p e q e li moltiplica tra di loro ottenendo il numero N che viene reso pubblico mentre i due numeri p e q devono rimanere segreti (per p e q si usano numeri molto grandi, nell'esempio consideriamo numeri piccoli):

$p = 5, q = 11$	$N=55$
-----------------	--------

2. Marco trova il valore della funzione di Eulero $\Phi(N)$ e memorizza il valore in b . La funzione di Eulero è definita per ogni intero positivo n ed è pari al numero degli interi compresi tra 1 e n che sono coprimi con n (due numeri x e y sono coprimi se non hanno divisori in comune, cioè se il loro massimo comun divisore è pari a 1). Tale funzione si dimostra essere $\Phi(N) = (p-1)(q-1)$





$\Phi(55) = (5 - 1)(11 - 1) = 40$	$b=40$
-----------------------------------	--------

3. Marco calcola il primo intero c che sia coprimo con b

$c = 2$	$MCD(2,40) = 2$	NO	$c=3$
$c = 3$	$MCD(3,40) = 1$	SI	

c è la seconda chiave pubblica.

4. Marco calcola il numero d tale che $d \times c \equiv 1 \pmod{b}$

$d = 2$	$2 \times 3 \pmod{40} =$	6	NO	$d=27$
$d = 3$	$3 \times 3 \pmod{40} =$	9	NO	
$d = 4$	$4 \times 3 \pmod{40} =$	12	NO	
\vdots	\vdots	\vdots	\vdots	
$d = 26$	$26 \times 3 \pmod{40} =$	38	NO	
$d = 27$	$27 \times 3 \pmod{40} =$	1	SI	

d è la chiave per decifrare e deve rimanere segreta.

Per trasmettere un messaggio a Marco, Giuseppe prepara la stringa m costituita da q lettere e associa ad ogni lettera un valore numerico (ad esempio la posizione della lettera nell'alfabeto inglese, 0-25):

$$m = (m_1, m_2, \dots, m_q)$$

Quindi Giuseppe legge le chiavi pubbliche N e c di Marco e trasmette i numeri di m uno alla volta cifrandoli come

$$h_k = m_k^c \pmod{N}, \quad k = 1, \dots, q$$

Per trasmettere ad esempio il numero 12, Giuseppe calcola e trasmette

$$h = 12^3 \pmod{55} = 23$$

Quando Marco riceve la cifra $h=23$, utilizza la chiave segreta d per decifrare il messaggio

$$h^d \pmod{N} = 23^{27} \pmod{55} = 12$$

Il codice riportato nel riquadro implementa le operazioni descritte nell'esempio su una stringa di caratteri minuscoli ed utilizza il compilatore online Pascal che si può trovare all'indirizzo:

https://www.onlinegdb.com/online_pascal_compiler

Il codice, chiaramente semplificato, non ha tutta una serie di controlli per gestire le varie eccezioni. Implementa varie funzioni tra cui $\text{pot}(x,y)$ che effettua l'elevamento a potenza tra x e y interi. Da considerare che il massimo numero intero ammissibile è $\text{MAXINT}=32767$, per cui m_k^c potrebbe andare in overflow (mi raccomando di scegliere valori dei numeri primi adeguati ai nostri retrocomputers!!!).

La funzione $\text{MCD}(x,y)$ sfrutta la proprietà che se due numeri x e y sono divisibili per un terzo numero z , allora anche la loro differenza lo sarà:

$$x = kz, \quad y = nz \quad \rightarrow \quad x - y = (k - n)z$$

e questo spiega perché calcolare MCD tra x e y è lo stesso che calcolarlo tra $x-y$ e y o tra x e $y-x$:

$$\text{MCD}(x, y) = \text{MCD}(x - y, y) = \text{MCD}(x, y - x)$$

Attraverso sottrazioni ripetute si arriva necessariamente ad un momento in cui $x=y$, ossia si ha differenza nulla. A questo punto vuol dire che sia il primo che il secondo numero sono divisibili per x , che è quindi il MCD cercato. Le altre funzioni implementano i vari steps previsti nella fase di cifratura dall'algorithm.

Bibliografia:

- [1] Alko R. Meijer. Algebra for Cryptologists. Springer, 2016.
- [2] S. Rubinstein-Salzedo. Cryptography. Springer, 2018.
- [3] K. Martin. Everyday Cryptography. Oxford University Press, 2017.

```

program RSA;
var p,q : INTEGER;
var N,b,c,d : INTEGER;
var i : INTEGER;
var str : STRING[10];
var mk,he : INTEGER;
function pot(x,y:INTEGER) : INTEGER;
var p,i:INTEGER;
begin
  p:=1;
  for i:=1 to y do
    p:=p*x;
  if y>=0 then
    pot:=p
  else begin
    pot:=0;
    writeln('Error!!!');
  end
end;
function MCD(a,b:INTEGER) : INTEGER;
begin
  while a<>b do
    if a>b then
      a:=a-b
    else
      b:=b-a;
  MCD:=a;
end;
function EulerPhi(p,q:INTEGER) : INTEGER;
begin
  EulerPhi:=(p-1)*(q-1);
end;
function firstPublicKey (p,q:INTEGER) : INTEGER;
begin
  firstPublicKey:=p*q;
end;
function secondPublicKey (b:INTEGER) : INTEGER;
var c:INTEGER; find:BOOLEAN;
begin
  find:=false;
  c:=2;
  while not find do
    if MCD(c,b)=1 then
      find:=true
    else
      c:=c+1;
  secondPublicKey:=c;
end;
function decryptKey (b,c:INTEGER) : INTEGER;
var d:INTEGER; find:BOOLEAN;
begin
  find:=false;
  d:=2;
  while not find do
    if (d*c) mod b=1 then
      find:=true
    else
      d:=d+1;
  decryptKey:=d;
end;
function encryptChar (mk,c,N:INTEGER) : INTEGER;
begin
  encryptChar:=pot(mk,c) mod N;
end;
begin
  p := 5;
  q := 11;
  N := firstPublicKey (p,q);
  b := EulerPhi (p,q);
  c := secondPublicKey (b);
  d := decryptKey (b,c);
  writeln('-----');
  writeln('N = ',N);
  writeln('Phi(N) = ',b);
  writeln('c = ',c);
  writeln('d = ',d);
  writeln('-----');
  str:='giuseppe';
  writeln(str);
  for i:=1 to length(str) do begin
    mk:=ord(str[i])-ord('a');
    he:=encryptChar (mk,c,N);
    writeln(str[i], ' ',mk,'->',he);
  end;
end.

```





ASTRONAVE FARMER (risveglio in astronave)

di Antonino Porcino

Astronave Farmer è un adventure testuale scritto in BASIC da Mario Pettenghi e pubblicato come listato *type-in* sulla rivista *MC MicroComputer* n. 40/aprile 1985.

Conobbi questo gioco nel lontano 1986 grazie a due miei amici che lo digitarono sul Laser 500 e successivamente mi ci fecero giocare, evitando che la visione del listato mi spoilerasse la soluzione.

Rimanemmo molto colpiti da questo semplice quanto affascinante adventure, tanto da spingerci alla lettura del noto libro "Scrivere un gioco di avventura" di Enrico Colombini. Tentammo anche di creare un adventure tutto nostro, senza però mai portarlo a compimento.

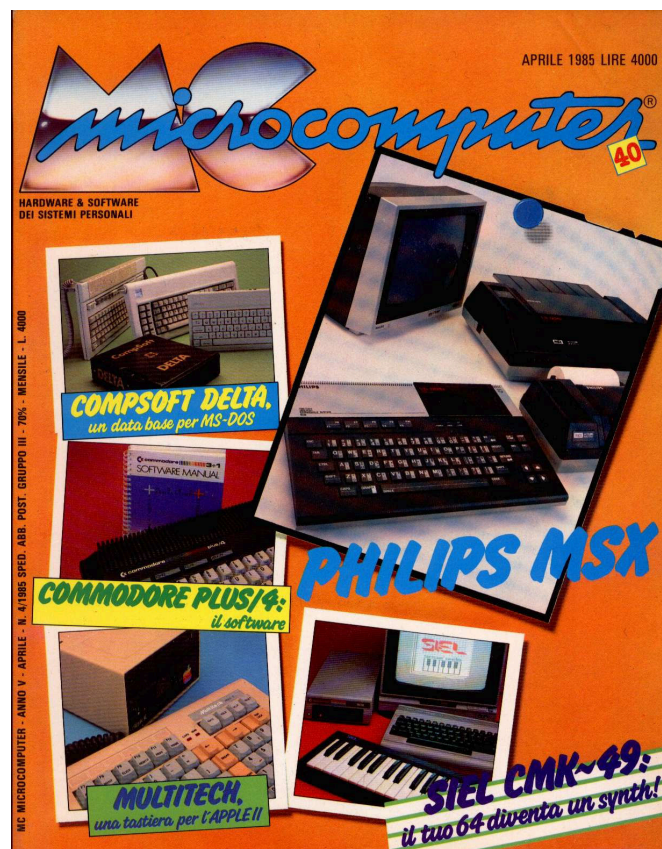


Di Astronave Farmer, oltre all'avventura in sé, mi affascinava il fatto che il computer potesse comprendere le frasi digitate in lingua italiana, reagendo con un comportamento intelligente. C'era un alone di mistero intorno al cosiddetto "parser", l'algoritmo che analizzava il testo inserito comprendendone il significato; non avevo la più pallida idea di come potesse funzionare... Per me era quasi magico!

Per molti anni ho cercato invano di ritrovare il listato di RETROMAGAZINE WORLD ANNO 4 - NUMERO 23

questo gioco, speravo che prima o poi saltasse fuori da qualche parte. Non ero neanche sicuro del suo nome, poiché i miei amici lo avevano cambiato in *Astronave Farmet* (con la "T" finale); a detta loro "suonava meglio".

Poi finalmente il link giusto, ed ecco il numero di *MC MicroComputer* con il listato bello e pronto in formato PDF!



Non potevo far altro che digitarlo e assaporare il piacere di giocarci nuovamente dopo quasi trentacinque anni, peraltro sullo stesso computer di allora: il Laser 500.

Che emozione rivedere quella scritta insistere sullo schermo... "COSA DEVO FARE?"

```

-----
Ora sono nella mia cabina. La mia capsula criogenica e' spenta.
L'arredamento e' ridotto all'essenziale. Posso uscire solo dalla porta sud.
DIREZIONI POSSIBILI: sud
OGGETTI VISIBILI: nessuno
-----
COSA DEVO FARE? █
  
```

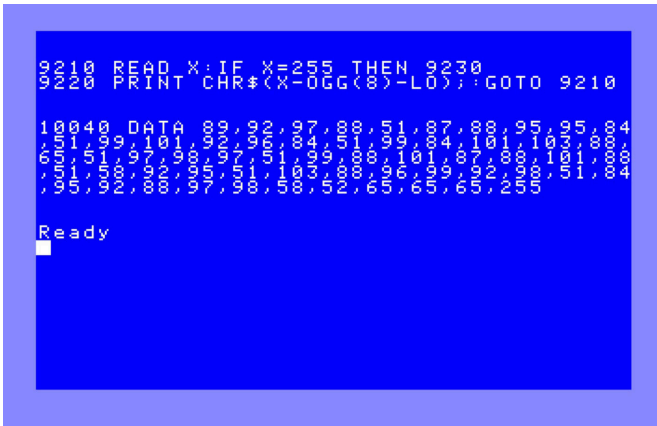
Il sorgente originale è scritto in MBASIC per lo Sharp MZ80B, ma risulta compatibile con la maggior parte dei BASIC degli home computer dell'epoca poiché non utilizza





nessuna istruzione specifica dello Sharp; i codici per pilotare lo schermo (CLS e NEWLINE) sono parametrizzati in delle variabili stringa, per cui è molto facile adattarlo a qualsiasi macchina.

Una curiosità: il finale del gioco è criptato tramite delle istruzioni "DATA", l'autore ha voluto così evitare che venisse rivelato durante la digitazione del listato:

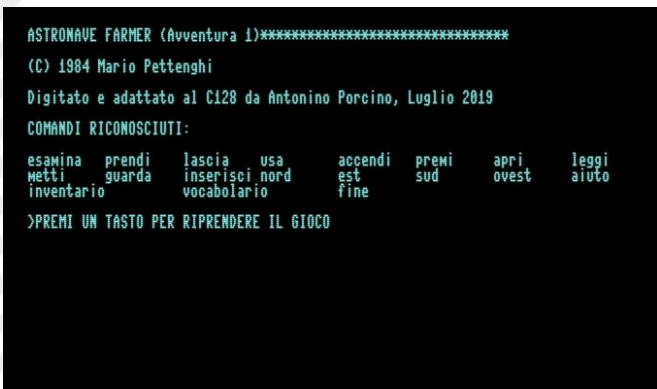


Stranamente, il listato pubblicato su MC Microcomputer contiene due bug, uno dei quali impedisce il completamento dell'avventura. Alla linea 2185 manca l'istruzione "O=21" che collega l'uscita ovest alla stanza n.21 sulla mappa, bloccando quindi il giocatore.

Il gioco è previsto per schermi a 80 colonne, ma ne ho fatto una versione anche per 40 colonne considerato che molti home computer non avevano modalità testo che arrivassero a 80 caratteri per linea.

Sul repo di GitHub, (<https://github.com/nippur72/8-bit-projects/tree/master/astronave-farmer>) oltre al sorgente originale ridigitato in formato testo, trovate alcuni adattamenti già pronti per essere giocati.

Usate il comando "VOCABOLARIO" per avere la lista dei comandi disponibili:



Se fate il port su altri computer, mi piacerebbe averne notizia in modo da aggiungere il programma alla lista.

Versioni disponibili:

Laser 500: questa è la versione più fedele all'originale per MBasic di Mario Pettenghi, poiché il BASIC del Laser 500 è perfettamente compatibile. Potete giocarci direttamente sul mio emulatore on-line all'indirizzo: nippur72.github.io/laser500emu/?load=astronave_farmer.bin

Commodore 64: adattato per girare a 40 colonne. In questa versione ho dovuto simulare la funzione "INSTR()" che manca nel CBM BASIC V2.0; ho anche dovuto riscrivere alcuni "IF THEN ELSE" a causa della mancanza dell'istruzione "ELSE".

Commodore 128: in modalità 80 colonne. L'unica differenza col sorgente originale è che l'istruzione "ELSE" nel BASIC del 128 deve essere preceduta dal carattere ":" quando è sulla stessa riga dell'"IF".

Commodore 16/Plus 4: simile alla versione 40 colonne per C64.

Buon divertimento!

Noi di **RetroMagazine World** non potevamo certo ignorare la richiesta/sfida di Antonino.

Ecco la versione per **Amstrad CPC** in **Locomotive Basic**, in 80 colonne, a firma di **Francesco Fiorentini**:

```

ASTRONAVE FARMER (Avventura 1)*****
(C) 1984 Mario Pettenghi
Digitato da Antonino Porcino, Luglio 2019
Adattato per Amstrad CPC da Francesco Fiorentini (RetroMagazine), Maggio 2020
COMANDI RICONOSCIUTI:
esamina   prendi   lascia   usa       accendi   premi
apri      leggi   sud      metti    nord      inserisci
est       ovest   aiuto    guarda  inventario
fine                                            vocabolario
>PREMI UN TASTO PER RIPRENDERE IL GIOCO

```

Come indicato da Antonino, il listato originale contiene comandi standard, riconosciuti da molti interpreti BASIC degli home computer dell'epoca. Il Locomotive Basic ha digerito facilmente il tutto: è bastato modificare il codice per pulire lo schermo (CLS) ed un paio di altri piccoli accorgimenti per farne un porting per questa macchina.

Come se non bastasse parlando con **Ermanno Betori** è saltato fuori come anche lui avesse giocato questo gioco e che addirittura ne avesse fatto un porting per **TI99/4A**. La particolarità della versione per TI99 scritta da Ermanno è di contenere le descrizioni in un data file separato rispetto al listato originale.

Ovviamente abbiamo già provveduto ad inviare il tutto ad Antonino per inserire i listati nel suo repository GitHub.

Francesco Fiorentini (RMW)





```

0 REM ASTRONAVE FARMER (c) 1983 Mario Pettenghi & MComputer
1 DEFINT A-Z:CO=8:VE=11:SO6=20:OG6=10
2 DIM CO$(8),VE$(11),SO6$(20),OG6(10),OG6$(10),FL(9)
3 CL$=CHR$(26):HO$=CHR$(21):UP$=CHR$(11):DO$=CHR$(10):RI$=CHR$(12):BE$=CHR$(7):WIDTH 78
10 GOTO 500
20 LINE INPUT "COSA DEVO FARE? ";FR$:PRINT:X=0:Y=0
30 FOR J=1 TO CO:IF INSTR(1,FR$,CO$(J))<>0 THEN X=J
40 NEXT:ON X GOSUB 100,100,100,100,8000,9000,9500,9700:IF X THEN RETURN
50 FOR J=1 TO VE:IF INSTR(1,FR$,VE$(J))<>0 THEN X=J
60 NEXT:FOR J=1 TO SO6:IF INSTR(1,FR$,SO6$(J))<>0 THEN Y=J
70 NEXT:ON X GOSUB 1000,1100,1200,1300,1400,1500,1600,1700,1800,1900,1920:IF X THEN RETURN
80 PRINT DO$:BE$;".SARDO` STUPIDO MA NON TI CAPISCO!";DO$:GOTO 20
90 N=0:E=0:S=0:O=0:PRINT BE$;
91 IF LO>20 THEN ON LO-20 GOSUB 2210,2220,2230,2240,2250,2260:GOTO 94
92 ON LO GOSUB
2000,2010,2020,2030,2040,2050,2060,2070,2080,2090,2100,2110,2120,2130,2140,2150,2160,2170,218
0,2190,2200
94 PRINT:PRINT"DIREZIONI POSSIBILI: ";:IF N THEN PRINT"nord ";
96 IF E THEN PRINT "est ";
97 IF S THEN PRINT "sud ";
98 IF O THEN PRINT "ovest ";
99 RETURN
100 IF LO=2 AND X=3 AND FL(1)=0 THEN PRINT "LA PORTA DI SICUREZZA E' BLOCCATA!":RETURN
101 IF LO=10 AND X=2 AND OGG(3) THEN PRINT"SOLO GLI UFFICIALI POSSONO ENTRARE!":RETURN
102 IF LO=17 AND FL(6)=0 AND X=4 THEN PRINT "IL PORTELLONE NON SI APRE!!!":RETURN
103 IF LO=17 AND X=4 AND FL(4)=0 THEN PRINT"VUOI USCIRE DALLA NAVE MENTRE E' IN VOLO?
IDIOTA.":RETURN
108 ON X GOTO 110,120,130,140
110 IF N=0 THEN 150 ELSE LO=N:RETURN
120 IF E=0 THEN 150 ELSE LO=E:RETURN
130 IF S=0 THEN 150 ELSE LO=S:RETURN
140 IF O=0 THEN 150 ELSE LO=O:RETURN
150 PRINT "Non posso andare in quella direzione...":PRINT:RETURN
200 PRINT "OGGETTI VISIBILI:";
210 FL=0:FOR J=1 TO OGG:IF OGG(J)=LO THEN PRINT TAB(19);OGG$(J):FL=1
220 NEXT:IF FL=0 THEN PRINT TAB(19);"nessuno"
230 IF LO=8 AND FL(2)=0 THEN GOTO 9100
300 PRINT STRING$(78,"-"):RETURN
500 FOR J=1 TO CO:READ CO$(J):NEXT
540 FOR J=1 TO VE:READ VE$(J):NEXT
550 FOR J=1 TO SO6:READ SO6$(J):NEXT
560 FOR J=1 TO OGG:READ OGG(J),OGG$(J):NEXT
590 LO=1:PRINT CL$
600 PRINT STRING$(78,"-"):GOSUB 90
610 PRINT:GOSUB 200
620 PRINT:GOSUB 20
630 GOTO 600
1000 IF Y=0 THEN PRINT "Non ci trovo nulla di interessante!":RETURN
1010 IF LO=2 AND Y=10 THEN PRINT "E' una porta di sicurezza. Puoi aprirla con la sua scheda
di riconoscimento.":RETURN
1020 IF LO=4 AND Y=11 THEN PRINT "E' di metallo, ma lo sportello non ha serrature.":RETURN
1030 IF LO=6 AND Y=13 AND OGG(3)=69 THEN PRINT"Era il comandante C.P.Ustinov. Grand'uomo. Non
ha soldi in tasca, ma ha una":PRINT"tessera con foto sul petto e una brutta
cera.":OGG(3)=6:RETURN
1040 IF LO=18 AND Y=16 AND FL(3)=0 THEN PRINT"C'e` scritto qualcosa in un vecchio
linguaggio. Un appassionato di computer saprebbe certo decifrarlo...":RETURN
1050 IF LO=18 AND Y=16 AND FL(3) THEN PRINT"Le tue conoscenze di computer ti consentono di
leggere il codice di uscita":FL(6)=1 :RETURN
1060 IF LO=21 AND Y=18 AND OGG(8)<>0 THEN PRINT"E' DI OTTONE. C'E' UNA TORCIA SU DI
ESSO!":OGG(8)=21:RETURN
1090 PRINT "Mi bruciano gli occhi dallo sforzo.":RETURN
1100 FL=0:IF Y=0 THEN 1115
1102 IF LO=6 AND Y=13 THEN PRINT"No, amico, chiedilo a Frankenstein!":RETURN

```





```
1103 IF LO=4 AND Y=2 THEN PRINT"No, non vale la pena non e' nemmeno digitale.":RETURN
1104 IF LO=5 AND Y=3 THEN PRINT "E'bloccato con una catenella!":RETURN
1109 FOR J=1 TO OGG:IF INSTR(1,OGG$(J),SOG$(Y)<>>0 THEN FL=J
1110 NEXT
1115 IF FL=0 OR LO<>OGG(FL) THEN PRINT "Andiamo, cerca di essere serio!!":RETURN
1120 OGG(FL)=0:PRINT "Ok, eseguito.":RETURN
1200 FOR J=1 TO OGG:IF SOG$(Y)=OGG$(J) THEN FL=J
1210 NEXT
1220 IF FL=0 OR OGG(FL)<>>0 THEN PRINT"COME POSSO LASCIARE QUELLO CHE NON HO, TOTANO!":RETURN
1230 OGG(FL)=LO:PRINT"Ok... come vuoi tu!":RETURN
1300 IF Y=0 THEN 1390
1310 IF LO=4 AND Y=2 THEN PRINT"E' spento...":RETURN
1320 IF LO=11 AND Y=7 AND FL(5) AND OGG(7)=0 THEN LO=14:PRINT"OK, FUNZIONA, MA NON C'E' PIU'
ENERGIA PER UN SECONDO UTILIZZO":RETURN
1330 IF LO=11 AND Y=7 AND FL(5)=0 THEN PRINT "Per farlo devi attivare prima il
motore.":RETURN
1340 IF LO=11 AND Y=7 AND OGG(7) THEN PRINT "E' necessaria la chiave del quadro
comandi!":RETURN
1390 PRINT "PROVA A USARE IL CERVELLO, SE NON E' UN OPTIONAL!":RETURN
1400 IF Y=17 AND OGG(8)=0 THEN OGG(8)=-1:PRINT"OK....":RETURN
1408 IF Y<>2 THEN PRINT "ACCENDITI LA BARBA, IPPOGRIFO!":RETURN
1410 IF LO=4 AND OGG(6) THEN PRINT"Dovrei avere anche una cassetta per farlo!":RETURN
1420 IF LO<>4 THEN PRINT "Dove lo trovo? Su Audio Review?":RETURN
1440 PRINT"OK.. C'e' un messaggio: ";DO$:PRINT"SEI STATO RISVEGLIATO DALL'IBERNAZIONE PER UNA
MISSIONE DI EMERGENZA!"
1450 PRINT"L'ASTRONAVE FARMER HA AVUTO UN'AVARIA. DEVI RIPROGRAMMARE IL COMPUTER":PRINT"DI
BORDO PER UN ATTERRAGGIO DI FORTUNA."
1460 PRINT"PER FARLO TI SERVE UN FLOPPY DISC CHE CONTIENE I DATI NECESSARI. TROVALO":PRINT"E
INSERISCILO NEL LETTORE DEL COMPUTER DI BORDO."
1470 PRINT"E' IN GIOCO ANCHE LA TUA VITA. AUGURI.":RETURN
1500 IF LO=13 AND Y=15 THEN PRINT "Le luci vacillano per un istante. Tutto
ok.":FL(5)=1:RETURN
1590 PRINT "Forse una petunia ha un Q.I. superiore al tuo...":RETURN
1600 IF Y=0 THEN PRINT "Ora ti apriro' il cranio se continui a dire scemate!":RETURN
1610 IF LO=2 AND OGG(1)=0 AND Y=10 THEN PRINT"La porta si apre lentamente...":FL(1)=1:RETURN
1620 IF LO=4 AND Y=11 THEN PRINT"C'E' UNA CASSETTA QUI DENTRO!":OGG(6)=4:RETURN
1690 PRINT "NON CI RIESCO!!!":RETURN
1700 IF LO=5 AND Y=3 THEN FL(3)=1: PRINT"Ora sai tutto di computer!":RETURN
1710 PRINT "E' meglio un buon libro di fantascienza":RETURN
1800 IF Y<>6 THEN 1830
1810 IF OGG(5)<>>0 THEN PRINT "Non mi sembra di averne una...":RETURN
1820 FL(2)=1:PRINT "Non e' proprio la mia taglia, ma puo' andare!":RETURN
1830 PRINT"L'unica cosa che potrei mettere e' la tua testa in un tritatutto!":RETURN
1900 PRINT CL$;:PRINT:RETURN
1920 IF LO<>16 OR Y<>5 THEN 1970
1930 IF OGG(4)<>>0 THEN PRINT "Buona idea! Peccato che non ne ho uno...":RETURN
1940 PRINT "COMPLIMENTI! Il calcolatore e' riprogrammato. Stiamo atterrando!":PRINT "Ora devi
uscire dall'astronave e lanciare l'SOS alla terra! FAI PRESTO!":FL(4)=1:RETURN
1970 PRINT"Potrei dire una volgarita'... ":RETURN
2000 PRINT"Ora sono nella mia cabina. La mia capsula criogenica e'
spenta.":PRINT"L'arredamento e' ridotto all'essenziale. Posso uscire solo dalla porta sud."
2005 S=2:RETURN
2010 PRINT"Sono in una stanza dalle pareti di metallo. Ci sono uscite in ogni
direzione":PRINT"La porta a sud ha una serratura magnetica e sembra molto robusta."
2015 N=1:E=3:S=6:O=4:RETURN
2020 PRINT"Sono in un corridoio del settore ricreativo. L'illuminazione e' scarsa,
ma":PRINT"riesco a orientarmi. Nessun segno di vita, mi sento in pericolo."
2025 N=5:E=24:O=2:RETURN
2030 PRINT"Questa e' la sala riunioni. C'e' un grande tavolo con molte sedie, un
armadio.":PRINT"Sul tavolo vedo un registratore a cassette, sembra possa funzionare ancora."
2035 E=2:RETURN
2040 PRINT"Sono nella sala di lettura. Ci sono numerosi volumi e in bella vista c'e'
una":PRINT"copia recente di MComputer. Non c'e' anima viva..."
```





```

2045 S=3:RETURN
2050 PRINT"Mi trovo in un lungo corridoio. Per terra di fronte a me c'è un
cadavere.":PRINT"Riesco a sentire il ronzio dei motori dell'astronave."
2055 N=2:S=7:RETURN
2060 PRINT"Eccomi in una grande stanza. Ci sono porte dappertutto. Sono vicino al
cuore":PRINT"della nave. La moquette non è di mio gusto ma sopporto stoicamente il fatto."
2065 N=6:E=9:S=10:O=8:RETURN
2070 PRINT"Sono nella cabina del comandante. L'oblo' di fronte a me è incrinato. La
temperatura è di settanta gradi, le radiazioni sono mortali."
2075 O=7:RETURN
2080 PRINT"Questa è l'armeria. Gli scaffali sono vuoti, siamo in missione
pacifica.":PRINT"La porta a est è chiusa e non vedo come aprirla..."
2085 O=7:RETURN
2090 PRINT"A ovest c'è un corridoio bloccato da un robot in avaria. A sud vedo
lo":PRINT"splendore della galassia da un oblo' trinitron!"
2095 E=11:N=7 :RETURN
2100 PRINT"Ecco la sala ufficiali. C'è un ascensore con le porte spalancate. Un
tavolo":PRINT"pieno di bicchieri (vuoti). Vedo a est una sala con pannelli di controllo."
2105 E=13:O=10:RETURN
2110 RETURN: REM pardon!
2120 PRINT"E' la sala controllo traffico. Di fronte a me c'è un pulsante con la
scritta":PRINT"Interruttore ascensore', mi sembra troppo bello..."
2125 O=11:RETURN
2130 PRINT "Vedo l'ascensore, ora inutilizzabile. Sono in un lungo corridoio.":PRINT"A ovest
c'è la sala di sbarco, a est la sala del computer."
2135 E=15:O=17:RETURN
2140 PRINT"Sono nella sala del computer. Vedo i giganteschi banchi di memoria
della":PRINT"Sinclair, e le tastiere a sfioramento. Sono vicino alla sala controlli."
2145 N=18:E=16:O=14:RETURN
2150 PRINT "La sala controllo. Di fronte a me c'è il drive per floppy disc, ma
c'è":PRINT"anche una telecamera dall'aspetto sinistro..."
2155 O=15: IF OGG(3)=0 THEN 9110
2156 RETURN
2160 PRINT"Ecco la sala per lo sbarco. A ovest c'è il portellone elettronico, a est
un":PRINT"lungo corridoio ben illuminato."
2165 O=19:E=14:RETURN
2170 PRINT"Questa è la sala controllo sbarco, C'è un terminale acceso, un
vecchio":PRINT"Quantum Beef, con un monitor HP a cristalli tossicodipendenti!"
2175 S=15:RETURN
2180 PRINT "Sono sbarcato. Il cielo è verde chiaro. Spira un dolce vento. A ovest
vedo":PRINT"una costruzione. A nord una caverna. Un sentiero porta a sud."
2185 N=20:S=22:E=17:RETURN:'o=21
2190 IF OGG(8)=-1 THEN 9200
2192 PRINT"L'interno della caverna è piu' buio che in un adventure... I rumori
sono":PRINT"altrettanto ripugnanti. Anima di Scott Adams, aiutami tu!"
2195 S=19:RETURN
2200 REM
2210 PRINT"Sono all'interno di un tempio alieno. C'è un grande altare con una
statua":PRINT"di uno scarafaggio a trenta braccia, c'è odore di insetticida!"
2215 E=19:S=23:RETURN
2220 PRINT"Sono su una collina. Il panorama sotto di me' è entusiasmante. Vedo villaggi
":PRINT"alieni, carri trainati da strani animali e la reclame della coca-cola."
2225 N=19:RETURN
2230 PRINT "Sono in una grande sala con dipinti alle pareti, gioielli e armi. Dalla
porta":PRINT"a est scorgo il resto del villaggio, ma una forza misteriosa non mi permette di
proseguire."
2235 N=21:RETURN
2240 PRINT"Questa è la sala addestramento colonizzatori. Non riesco a orientarmi.":PRINT"Ci
sono strane attrezzature in ogni punto."
2245 E=25:S=24:N=25:O=24:RETURN
2250 PRINT "Sono sempre nei locali di addestramento. Non conosco quest'area della nave!"
2255 S=26:N=24:E=24:O=25:RETURN
2260 PRINT "Questa è la sala addestramento colonizzatori. Non riesco a orientarmi.":PRINT"Ci

```





```
sono strane attrezzature in ogni punto..."
2265 N=3:S=24:O=25:E=25:RETURN
8000 PRINT:IF LO=1 THEN PRINT"SEI SOLO ALL'INIZIO.":RETURN
8010 IF LO=2 THEN PRINT"STAI IN GUARDIA E CERCA CON PAZIENZA.":RETURN
8020 IF LO<27 AND LO>23 THEN PRINT"STUDIA I PUNTI....":RETURN
8030 IF LO=17 THEN PRINT"CI VORREBBE IL CODICE COMPUTERIZZATO...":RETURN
8040 IF LO=18 AND FL(3)=0 THEN PRINT"SE TI FOSSI PREPARATO PRIMA, MAGARI LEGGENDO QUALCOSA DI
UTILE...":RETURN
8050 IF LO=18 THEN PRINT PRINT"EPPURE QUALCOSA CI DEVE ESSERE...":RETURN
8060 IF LO=21 THEN PRINT"INSISTI, VEDO QUALCOSA...":RETURN
8070 IF LO=20 THEN PRINT"SE CI FOSSE MENO BUIO...":RETURN
8900 IF RND(1)>.5 THEN PRINT"SONO NEI GUAI COME TE." ELSE PRINT"MI HAI SCAMBIATO PER
L'HOBBIT?"
8910 PRINT:RETURN
9000 PRINT CL$;"Inventario:":PRINT"In questo momento ho con me:";DO$:FL=0
9010 FOR J=1 TO OGG:IF OGG(J)=0 THEN PRINT OGG$(J):FL=1
9020 NEXT:IF FL=0 THEN PRINT "un bel niente..."
9030 RETURN
9100 PRINT:IF LO=8 THEN PRINT "LA MIA PELLE STA PRENDENDO FUOCO. HO POCHI SECONDI DI VITA E
NON MI STO":PRINT"DIVERTENTO PER NIENTE. AAAARGH!":GOTO 9150
9110 PRINT DO$;DO$;"*** ATTENZIONE **** IDENTIFICAZIONE NEGATIVA. IL TUO TRACCIATO
METABOLICO":PRINT"NON CORRISPONDE AL CAMPIONE. NON SEI IL COMANDANTE USTINOV!"
9112 PRINT DO$;"Un raggio laser mi trapassa da parte a parte, con poca discrezione.
Muio!":GOTO 9150
9150 PRINT DO$;"Hai fallito miseramente. Vuoi ritentare? (S/N)"
9160 A$=INKEY$:IF A$="" THEN 9160
9170 IF A$="s" OR A$="S" THEN RUN
9180 PRINT CL$;"ADDIO...":CLEAR:END
9200 PRINT DO$;BE$;DO$:PRINT"HAI TERMINATO QUESTA AVVENTURA...";DO$
9205 PRINT "Di fronte a me c'e` una lastra di pietra alta venti metri: reca una scritta:";DO$
9210 READ X:IF X=255 THEN 9230
9220 PRINT CHR$(X-OGG(8)-LO);:GOTO 9210
9230 PRINT: PRINT DO$;"***COMPLIMENTI***";DO$:END
9500 PRINT CL$;BE$;"ASTRONAVE FARMER (Avventura 1)*****"
9510 PRINT DO$;"(C) 1984 Mario Pettenghi";DO$
9520 PRINT "COMANDI RICONOSCIUTI:";DO$
9530 FOR J=1 TO VE:PRINT VE$(J),:NEXT:FOR J=1 TO CO:PRINT CO$(J),:NEXT
9540 PRINT DO$;">PREMI UN TASTO PER RIPRENDERE IL GIOCO":A$=INKEY$
9550 A$=INKEY$:IF A$="" THEN 9550
9560 PRINT CL$:RETURN
9700 PRINT DO$;"*** SEI SICURO DI VOLER TERMINARE? ***"
9710 A$=INKEY$:IF A$="" THEN 9710
9720 IF A$="n" OR A$="N" THEN PRINT DO$;"Hai bisogno di una LUNGA vacanza...":RETURN
9730 IF A$="s" OR A$="S" THEN PRINT CL$:END
9740 GOTO 9710
10000 DATA nord ,est,sud,ovest,aiuto,inventario,vocabolario,fine: ' VERBI
10010 DATA esamina,prendi,lascia,usa,accendi,premi,apri,leggi,metti,guarda,inserisci:'VERBI
10020 DATA scheda,registratore,MC,tessera,floppy disc,tuta,ascensore,codice,portello
10021 DATA
porta,armadio,cassetta,cadavere,chiave,pulsante,monitor,torcia,altare,laser,portachiavi
10030 DATA 4,scheda,5,MC,69,tessera,8,floppy disc,9,tuta,32,cassetta,25,chiave,29,torcia,
4,laser,7,portachiavi
10040 DATA
89,92,97,88,51,87,88,95,95,84,51,99,101,92,96,84,51,99,84,101,103,88,65,51,97,98,97,51,99,88,
101,87,88,101,88,51,58,92,95,51,103,88,96,99,92,98,51,84,95,92,88,97,98,58,52,65,65,65,255

REM le seguenti linee risolvono due BUG sulla mappa
REM (Antonino Porcino 2019)

2075 E=7:RETURN
2185 N=20:S=22:E=17:o=21:RETURN
```





Introduzione ad ARExx – terza parte

di Gianluca Girelli

Questo articolo è comparso la prima volta sulle pagine di *Bitplane* nel maggio del 2013.

MATRICI, ISTRUZIONI CONDIZIONALI E PROCEDURE

Dopo aver imparato nel tutorial precedente come si lavora con le stringhe o, per meglio dire, quanto sono potenti in ARExx le istruzioni di gestione delle stringhe, entriamo nel dettaglio delle procedure, delle matrici e, soprattutto, delle istruzioni condizionali.

PROCEDURE

A differenza di altri linguaggi (ad esempio il Pascal), in cui la differenza tra funzioni e procedure è strettamente e sintatticamente formale (oltre che concettuale), ARExx gestisce i sottoprogrammi in modo molto semplice. Fin dall'inizio, infatti, sia l'autore del linguaggio che la comunità di sviluppo preferirono la semplicità di utilizzo e la facilità di lettura alla complessità formale dei linguaggi strutturati. Senza voler andare troppo in profondità nella teoria ci limiteremo a ricordare che, mentre una procedura generalmente serve ad effettuare delle azioni/operazioni, una funzione serve per "restituire" un risultato. Quindi: una procedura viene di solito invocata semplicemente scrivendone il nome, mentre una funzione (restituendo un risultato) viene assegnata ad una variabile.

Per i nostri scopi basti sapere che in ARExx i sottoprogrammi vengono invocati con il comando "CALL nome_sottoprogramma(espressione, ...)".

La lista di parametri in parentesi non ha limitazioni di lunghezza e viene valutata da sinistra a destra. Se il nostro sottoprogramma fosse in realtà una funzione, allora l'ultima espressione potrebbe essere la variabile che contiene il risultato finale. In tal caso potrebbe essere più conveniente (e concettualmente corretto) usare la seguente sintassi: "risultato = nome_sottoprogramma(espressione, ...)".

Come visto nel numero precedente, la subroutine viene indentificata da una "label" iniziale che funge sia da nome del sottoprogramma, sia da indirizzo logico di riferimento per la chiamata da parte del codice principale e termina con l'istruzione "return":

```
; esempio di chiamata
```

ENGINE (GAME)

Il motore grafico è il nucleo software di un videogioco o di qualsiasi altra applicazione con grafica in tempo reale.

Esso fornisce le tecnologie di base, semplifica lo sviluppo, e spesso permette al gioco di funzionare su piattaforme differenti come le console o sistemi operativi per personal computer. La funzionalità di base fornita tipicamente da un motore grafico include un motore di rendering ("renderer") per grafica 2D e 3D, un motore fisico o rilevatore di collisioni, suono, scripting, animazioni, intelligenza artificiale, networking, e scene-graph.

```
pull frase           ;legge la variabile "frase"
in input
call Parser(frase)   ;chiama il sottoprogramma
"Parser" passando la variabile "frase"

; struttura del sottoprogramma
Parser:
    istruzione 1     ;queste istruzioni usano il
contenuto di "frase"
    .....
    istruzione n
return
```

Detto questo, d'ora in poi useremo per semplicità il termine generico di "sottoprogramma" evitando, se possibile, il termine "procedura". Il motivo è il seguente: come sappiamo, in ARExx le variabili sono tutte "globali" e non è necessario dichiarare una variabile prima di usarla. Nonostante ciò, generalmente le variabili utilizzate all'interno di un sottoprogramma sono note solamente al processo chiamante. Per modificare l'ambito di visibilità di una variabile è stata allora inserita la parola riservata "PROCEDURE".

Tale istruzione può essere utilizzata all'interno di una routine per rendere certe variabili "sconosciute" alle istruzioni che seguono.

PROCEDURE però può anche essere usata, in congiunzione con "EXPOSE", per modificare questo comportamento. Prima di vedere un esempio, introduciamo un altro argomento: le matrici.

MATRICI

Abbiamo appena ricordato che in ARExx non è necessaria





nè la dichiarazione esplicita, nè l'assegnazione iniziale dei valori.

Dobbiamo però ricordare che se la variabile non viene inizializzata, essa conterrà il carattere (o i caratteri) che compongono il suo simbolo, trasformati in lettere maiuscole. In altre parole, il valore della variabile "amiga" sarà inizialmente "AMIGA". Questo tipo di variabile viene definito "semplice". Una variabile può anche essere composta, come "amiga.2000". In questo caso, la parte iniziale viene definita "stem" e comprende il nome di variabile fino al punto (compreso), mentre il resto della variabile viene definito "coda". Una variabile composta non inizializzata contiene al suo interno il valore della sua coda. In altre parole, se non inizializzata, "amiga.2000" conterrà inizialmente il valore "2000".

Una cosa interessante delle variabili composte è che se lo stem viene inizializzato con un dato valore, tutte le possibili variabili composte che iniziano con quello stem, avranno lo stesso valore. Cioè, se ad "amiga." assegno il valore "mitico", anche le variabili "amiga.500", "amiga.1200", "amiga.3000" e "amiga.4000" conterranno "mitico".

E' evidente l'enorme facilità, anche concettuale, che questo tipo di inizializzazione ha rispetto a quella tradizionale. Supponendo di voler inizializzare a zero un vettore di 100 elementi, basterà fare:

```
vettore. = 0
```

anzichè dover usare il classico:

```
for i=1 to 100
  vettore.i = 0
next i
```

Le sorprese però non finiscono qui. Poichè la "coda" di uno stem può contenere anche altri punti (es: a.b.c), lo stem rappresenta anche il nome della matrice multidimensionale ad esso associata. In questo caso, supponendo di voler inizializzare a zero una matrice

bidimensionale composta da 100x100 elementi, l'operazione:

```
matrice. = 0
```

equivale a:

```
for x=1 to 100
  for y=1 to 100
    matrice.x.y=0
  next y
next x
```

Qualcuno ha ancora dei dubbi sulla potenza di ARexx?

A questo punto, dopo aver parlato di sottoprogrammi e di variabili semplici e composte, risulterà chiaro il funzionamento del codice sotto riportato:

```
/* questo e' il programma principale */
j=1; x.1='a'
call alcune_variabili
say j k m      /* visualizzera' "1 7 M" */
exit

/* questo è un sottoprogramma */
alcune_variabili: procedure expose j k x.j
  say j k x.j /* visualizzera' "1 K a" */
  k=7; m=3 /* notare che "m" non e' esposta */
return
```

Si noti la chiamata alla subroutine "alcune_variabili", l'inizializzazione implicita delle variabili semplici e composte, l'uso dello stem nonchè quello delle istruzioni "PROCEDURE" ed "EXPOSE".

ISTRUZIONI CONDIZIONALI E CICLI ITERATIVI

Concludiamo il nostro tutorial con un riepilogo sulle istruzioni condizionali (peraltro già introdotte sul numero precedente con alcuni esempi di analizzatori sintattici) ed i cicli iterativi.

Come tutti i linguaggi più evoluti ARexx implementa costrutti per l'iterazione di istruzioni ma, invece di avere diversi tipi di istruzioni, utilizza la seguente sintassi:

```
DO [ripetitore][condizione];
  [lista istruzioni]
END
```

Porting

La portabilità, (in lingua inglese porting) indica in informatica un processo di un trasposizione, a volte anche con modifiche, di un componente software volto a consentirne l'uso in un ambiente di esecuzione diverso da quello originale.

Due attività collegate con il porting ma distinte da esso sono l'emulazione e la compilazione incrociata o cross-compiling.





dove:

- "ripetitore" può essere: FOREVER, un'espressione singola (da valutare) o un'espressione complessa nella forma "nome=espr1 [TO espr2][BY espr3][FOR espr4];
 - "condizione" può essere: [WHILE esprA] oppure [UNTIL esprB];
 - e "lista istruzioni" può essere una qualunque sequenza di istruzioni del linguaggio.

Per quanto riguarda i costrutti condizionali, essi si presentano nelle forme:

```
IF espressione[;] THEN[;] istruzione [ELSE[;]
istruzione]
```

oppure:

```
SELECT lista_when [OTHERWISE[;][lista_istruzioni]]
END;
```

dove:

- "lista_when" è costituita da uno o più costrutti nella forma "WHEN espressione[;] THEN[;] istruzione";
 - e "lista istruzioni" può essere una qualunque sequenza di istruzioni del linguaggio.

Nel frammento di codice che segue si può vedere l'uso dei costrutti appena citati così come utilizzato nell'avventura testuale "cyber.rexx", scritta a scopo didattico dall'autore dell'articolo. Giusto per introdurre l'ambito di utilizzo, si tratta del sottoprogramma che gestisce la "navigazione" all'interno del mondo di gioco, per semplicità implementato come una matrice bidimensionale di "X" righe ed "Y" colonne.

Ecco quindi spiegato il fatto di variare i valori della coppia "x,y" a seconda del movimento voluto: "vai est", ad esempio, significa spostarsi di una casella verso destra (y=y+1 a parità di x).

```
===== preso da cyber.rexx =====
```

Vai:

```
if find(Situazione.7.Pos,nome)>0 then say'Non
puoi andare a ' || nome
else do
  select
    when nome='NORD' then x=x-1
    when nome='SUD' then x=x+1
    when nome='EST' then y=y+1
    when nome='OVEST' then y=y-1
    when nome='UP' then x=x-1
    when nome='DOWN' then x=x+1
```

```
otherwise say "non capisco. riprova"; say;
return
end
Pos=(x-1)*maxcol+y
do i=1 to 6
  if Situazione.i.Pos~='' then say Situazione.i.Pos
end
end
return
=====
```

CONCLUSIONI

Alla fine di questo primo blocco di tre tutorial dovremmo aver accumulato tutta la conoscenza necessaria per scrivere il nostro primo, vero programma complesso. Nel prossimo futuro, basandoci anche sull'articolo pubblicato sul numero 17 relativo al "Game Coding", inizieremo a costruire un "engine" [BOX] di gioco per avventure testuali. Nonostante ARexx sia nato come programma di "scripting", il codice verra' redatto tenendo bene a mente i piu' moderni paradigmi di programmazione, in modo da facilitare l'eventuale "porting" [BOX] in altri linguaggi.

Ulteriori approfondimenti sull'uso e sulla sintassi di ARexx possono essere trovati nei testi citati in bibliografia.

Non perdetevi dunque il prossimo ciclo di articoli: "Game Coding con ARexx".

BIBLIOGRAFIA

- Mike Cowlshaw "The REXX Language: A Practical Approach to Programming" (1985) Prentice Hall. ISBN 0-13-780651-5.
- Chris Zamara, Nick Sullivan "Using Arexx on the Amiga" (1991) Abacus Software Inc. ISBN 1-55755-114-6.
- AmigaOS 2.0 Manuale di sistema
- http://it.wikipedia.org/wiki/Game_engine
- <http://it.wikipedia.org/wiki/Porting>

Utilizzo degli esempi

Come riportato nell'articolo sul num. 20, per utilizzare gli esempi bisogna salvare lo script in modalita' testo nel formato "nome_script.rexx". Per lanciare lo script basta digitare da shell:
 ">rx nome_script.rexx"
 o, piu' semplicemente:
 ">rx nome_script".





COMAL: la strana storia di un linguaggio dimenticato - seconda parte

di *András Vajda*

Continua la nostra collaborazione con il gruppo **Retro Programming Italia (RPI) by The Nerds**

Come secondo assaggio, nulla di meglio che un classico quicksort come proposto dal disco COMAL 80. Algoritmo oggettivamente non facile da implementare in BASIC V2 in meno di 60 LOC, mantenendo l'assoluta chiarezza del seguente listato. Chiunque abbia programmato intensivamente in BASIC V2 saprà sicuramente apprezzare, tra l'altro, la robustezza e l'intuitività dello statement **WHILE NOT EOD DO** che cancella con un colpo di spugna l'incubo dell'off-by-one e più in generale il temibile errore **OUT OF DATA**.

Si noti anche l'eleganza degli statement di assegnazione composita, che eliminano le inguardabili istruzioni di autoassegnazione come **A = A + 1**, probabilmente uno degli aspetti soggettivamente più odiati dai programmatori HLL alle prese con il BASIC V2.

```
0010 // save "quicksort"
0020 //
0030 // by UniComal ApS. may 1984
0040 //
0050
0060 PROC quicksort(REF a$( ),left,right,reclen)
CLOSED
0070 DIM pivot$ OF reclen, buffer $OF reclen
0080 partition(left,right,left,right) // sort
a$(left:right)
0090
0100 PROC partition(left,right,i,j)
0110 pivot$:=a$((left+right)DIV2) // get
middle element as pivot
```

```
0120 REPEAT // perform swappings
0130 WHILE pivot$>a$(i) DO i :=+1
0140 WHILE pivot$<a$(j) DO j :-1
0150 IF i<=j THEN swap(a$(i),a$(j));i:+1;j:-1
0160 UNTIL i>j
0170 IF left<j THEN partition(left,j,left,j) //
sort a$(left:j)
0180 IF i<right THEN
partition(i,right,i,right) // sort a$(i:right)
0190 ENDPROC partition
0200
0210 PROC swap(REFa$,REFb$)
0220 buffer$:=a$;a$:=b$;b$:=buffer$
0230 ENDPROC swap
0240 ENDPROC quicksort
0250
0260 PAGE
0270 DIM message$ (1:50) OF 35
0280 messageno :=0
0290
0300 WHILE NOT EOD DO
0310 messageno :=+1
0320 READ message$ (messageno )
0330 ENDWHILE
0340
0350 // sort error messages:
0360
0370 PRINT "sortingstarted:"
0380 quicksort(message$( ),1,messageno,35)
0390
0400 // print the sorted messages:
0410
0420 FOR i :=1 TO messageno DO PRINT i,"";message$(
i)
0430
0440 END "End of sort "
0450
0460 // start-of-data
0470
0480 DATA "format error"
0490 DATA "syntax error "
0500 DATA "type conflict"
0510 DATA "function argument error"
0520 DATA "statement too long or too complicated"
0530 DATA "system error"
0540 DATA "name too long"
0550 DATA "bracket error"
```

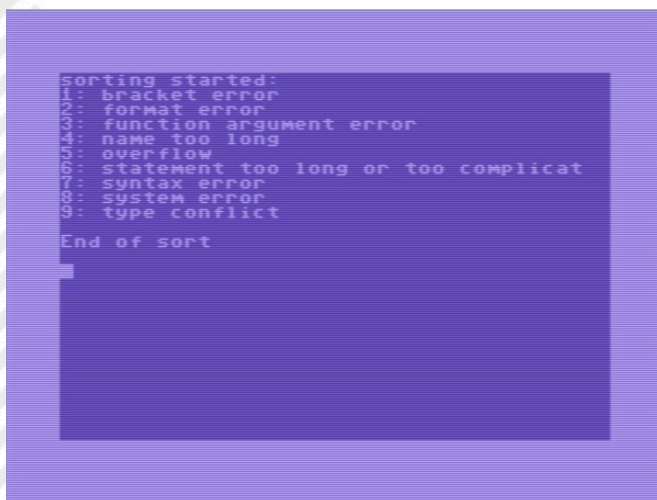


Figura 1: Risultato del Quicksort





```
0560 DATA "overflow"
0570
0580 // end-of-data
```

Un terzo esempio, sempre tratto dal disco originale, genera esaustivamente tutte le soluzioni al classico problema scacchistico detto 'delle otto regine': data una scacchiera standard bicolore da 64 caselle, si devono posizionare 8 regine in modo tale che nessuna di esse minacci le altre, ossia ponendo al più un solo pezzo per ogni riga, colonna e diagonale. Sebbene non si disponga di una formula chiusa per calcolare il numero delle soluzioni a partire dalla dimensione n della scacchiera data, è noto che con 8 regine esistono 92 soluzioni distinte in totale, derivate per rotazioni e riflessioni da 12 soluzioni di base, dette 'uniche'. Il breve programma qui proposto genera sequenzialmente, con accattivanti animazioni grafiche, tutte le 92 possibili configurazioni che risolvono il problema. Si noti che i dati degli sprites vengono elegantemente letti da un file dati sequenziale e non sono incorporati nel codice.

Si invita il lettore ad apprezzare la chiarezza del programma, grazie alla forte strutturazione, all'uso estensivo di nomi lunghi per le variabili e per le procedure, oltre alla presenza dei commenti: ricordando inoltre che tutto ciò avveniva nel lontano 1985 (ed era comunque possibile anche prima su Commodore PET e C64, con COMAL 0.1x), quando gli home BASIC su hardware di pari classe offrivano livelli di leggibilità, sintesi e mantenibilità nettamente inferiori e i compilatori per HLL come Pascal erano una rarità. Il programma, nello specifico, fa uso di backtracking: una soluzione decisamente classica sebbene dalle prestazioni non entusiasmanti.

```
0010 // SAVE "queens"
0020 //
0030 // (c) 1984 by UniComal ApS .
0040 //
0050 // Find all possible placements
0060 // o f 8 queens on a chess board
0070 // in such a fashion that none
0080 // is checking any other piece,
0090 // i.e. each row , column and
0100 // diagonal must contain at most
0110 // one piece .
0120
0130 PAGE
0140 m:=24 // size of squares //
0150 l :=8*m // size of board
0160 x0:=(320- 1 )-4
```

```
0170 y0:=(200- 1 ) DIV 2
0180 homex:=x0-27
0190
0200 red :=2; blue :=6
0210 black :=0; orange :=8
0220
0230 USE graphics
0240 graphicscreen ( 1 )
0250 // blue background //
0260 background ( blue )
0270 border (-1)
0280 clearscreen
0290
0300 USE sprites
0310 queen :=0
0320 DIM xpos ( 0 : 7 )
0330 // define queen shape //
0340 OPEN FILE 1 , "queen.spr",READ
0350 define(queen,GET$(1,64))
0360 CLOSE
0370
0380 DIM row(1:8)
0390 // row(i) = "no queen on i-th row" //
0400 DIM d1(1:2*8)
0410 // d1(i) = "no queen on i-th upleft to
lowright diagonal //
0420 DIM d2(1:2*8-1)
0430 // d2(i) = "no queen on i-th low left to
upright diagonal" //
0440
0450 //-----main-----//
0460
0470 clear.and.draw.board
0480 trycol(1)
0490 FOR i :=0 TO 7 DO
0500 slideto(i,homex)
0510 ENDFOR i
0520 END "End Queens"
0530
0540 //-----procedures-----//
0550
0560 PROC paint(c,x,y,w,h)
0570 // paint rectangle //
0580 viewport(x,x+w-1,y,y+h-1)
0590 pencolor(c)
0600 fill(x,y)
0610 ENDPROC paint
0620
0630 PROC c l e a r . and . draw . board
0640 FOR i :=1 TO 8 DO row(i):=TRUE
0650 FOR i :=1 TO 2*8 DO d1(i):=TRUE
```





```
0660 FOR i :=1 TO 2*8-1 DO d2(i):=TRUE
0670 // draw t ext //
0680 pencolor(orange)
0690 type(0,180,1,2,"EIGHT")
0700 type(0,160,1,2,"QUEENS")
0710 pencolor(black)
0720 type(0,128,1,2,"Number")
0730 type(0,112,1,2,"of")
0740 type(0,96,1,2,"boards")
0750 boards :=-1
0760 newboard
0770 // draw border //
0780 paint(black,x0-4,y0-4,8+1,4)
0790 paint(black,x0-4,y0,4,1)
0800 paint(black,x0-4,y0+1,8+1,4)
0810 paint(black,x0+1,y0,4,1)
0820 // definecolors of the queens //
0830 y:=y0+22
0840 FOR sprite:=0 TO 7 DO
0850 spritecolor(sprite,black)
0860 xpos(sprite):=homex
0870 spritepos(sprite,homex,y)
0880 identify(sprite,queen)
0890 showsprite(sprite)
0900 spritesize(sprite,0,0)
0910 y:+m
0920 ENDFOR sprite
0930 // draw board //
0940 y:=y0
0950 color :=red
0960 FOR row:=1 TO 8 DO
0970 x:=x0
0980 FOR col :=1 TO 8 DO
0990 paint(color,x,y,m,m)
1000 x:+m
1010 color:=red+orange-color
1020 ENDFORcol
1030 y:+m
1040 color :=red+orange-color
1050 ENDFOR row
1060 last:=-1
1070 viewport(0,319,0,199)
1080 ENDPROC clear.and.draw.board
1090
1100 PROC pause
1110 FOR i :=1 TO 7000 DO NULL
1120 ENDPROC pause
1130
1140 PROC slideto(sprite,x1)
1150 y1:=y0+sprite*m+22
1160 s tp :=SGN( x1-xpos(sprite))

1170 IF stp<>0 THEN
1180 FOR x:=xpos(sprite) TO x1 STEP stp DO
1190 spritepos(sprite,x,y1)
1200 ENDFOR x
1210 xpos(sprite):=x1
1220 ENDIF
1230 ENDPROC slideto
1240
1250 PROC place.queen(i,j)
1260 row(i):=FALSE; d1(i+j):=FALSE; d2(8+i-
j):=FALSE
1270 IF last>0 AND last<>j THEN
1280 slideto(last-1,homex)
1290 ENDIF
1300 slideto(j-1,x0+(i-1)*m)
1310 last :=-1
1320 ENDPROC place.queen
1330
1340 PROC remove.queen(i,j)
1350 row(i):=TRUE; d1(i+j):=TRUE; d2(8+i-j):=TRUE
1360 IF last >0 THEN
1370 slideto(last-1,homex)
1380 ENDIF
1390 last := j
1400 ENDPROC remove . queen
1410
1420 PROC trycol(j)
1430 FOR i :=1 TO 8 DO
1440 IF row(i) AND d1(i+j) AND d2(8+i-j) THEN
1450 place.queen(i,j)
1460 IF j<8 THEN
1470 trycol(j+1)
1480 ELSE // all queens placed //
1490 newboard
1500 pause
1510 ENDIF
1520 remove.queen(i,j)
1530 ENDIF
1540 ENDFOR i
1550 ENDPROC trycol
1560
1570 PROC newboard
1580 // draw new number //
1590 pencolor(black)
1600 boards :=+1
1610 type (0,24,3,6,STR$(boards))
1620 ENDPROC newboard
1630
1640 PROC type(x0,y0,xsize,ysize,text$)
1650 IF LEN(t ext $ ) MOD 2=1 THEN
1660 x0:+ xsize*8
```





```
1670 ENDIF
1680 textstyle(xsize,ysize,0,0)
1690 plottext(x0,y0,text$)
1700 ENDPROC type
```

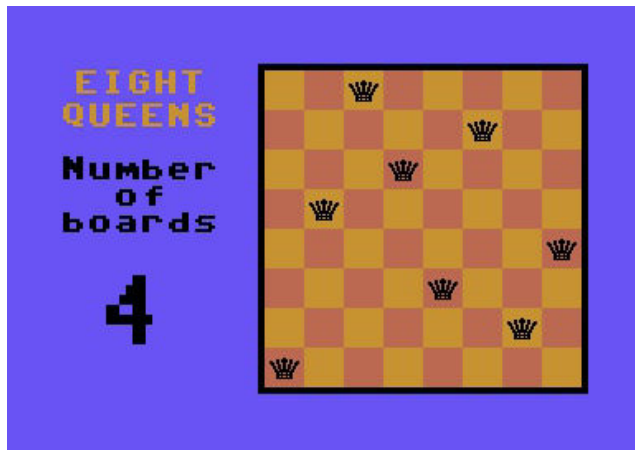


Figura 2: Il problema delle 'otto regine'

Bibliografia essenziale

I testi originali di Børge R. Christensen, creatore del linguaggio con Benedict Løfstedt, sono [Chr82, Chr84]. Altri testi generici sul linguaggio includono [Ath82, BP88, Gra85, Kel84, LO90].

COMAL 80 per C64 è interamente documentato nel manuale originale Commodore COMAL 80 (a cura di Frank Bason e Leo Højsholt-Poulsen, 1985) pubblicato in duplice edizione inglese e danese, relativo alla versione 2.0 su cartuccia con disco allegato. Specifici per C64 anche i testi di Lindsay [Lin83] e J. William Leary [Lea86].

Conclusioni

Si è presentata una brevissima cronistoria della definizione e creazione del linguaggio COMAL (in particolare le versioni milestone 75 e 80 e lo standard 1982/83 noto come COMAL KERNAL) da parte del matematico danese Børge R. Christensen e del suo collega informatico Benedict Løfstedt, seguita dalle principali tappe dello sviluppo commerciale europeo del linguaggio in numerose versioni, concentrandoci in particolare sulle principali release per PET e Commodore 64. Si sono proposti alcuni semplicissimi sorgenti di esempio al fine di incuriosire il lettore, mostrando solo alcuni aspetti della potenza e della leggibilità di un linguaggio di notevole innovatività nell'ambito degli home computer: con la speranza di avere suscitato anche qualche gradito ricordo in chi all'epoca avesse provato ad utilizzare il linguaggio su C64 come alternativa al BASIC V2 o su altre piattaforme.

Si è infine esposta una bibliografia di riferimento sul linguaggio, comprendente pressoché tutti i testi in lingua inglese ritenuti più autorevoli e di maggiore diffusione. Qualora dopo la pubblicazione del presente articolo si riscontrasse sufficiente interesse attorno al linguaggio, ciò che l'autore auspica, si potrà dare continuità alla trattazione con la presentazione di ulteriori esempi e quesiti risolti con l'uso di COMAL 80 per C64, anche in parallelo con le relative soluzioni in Assembly 6510 e BASIC V2.

Riferimenti bibliografici

- [Ath82] Roy Atherton, Structured programming with comal (ellis horwood series in computers and their applications), Ellis Horwood Ltd, Publisher, 1982.
- [BJ66] Corrado Boehm and Giuseppe Jacopini, Flow diagrams, turing machines and languages with only two formation rules, Commun. ACM 9 (1966), no. 5, 366-371.
- [BP88] Marcus D. Bowman and Stephen Pople, Computing studies in context: Comal programming guide, Heinemann Educational Publishers, 1988.
- [Chr82] Borge Christensen, Beginning comal, Harwood & Charles Pub Co, 1982.
- [Chr84] Borge R Christensen, Comal reference guide, Toronto PET users group, 1984.
- [Dij82] Edsger W. Dijkstra, How do we tell truths that might hurt?, SIGPLAN Not. 17 (1982), no. 5, 13-15.
- [Dij87] Edsger W. Dijkstra, Go to statement considered harmful, Commun. ACM 30 (1987), no. 3, 195-196.
- [FW08] Daniel P. Friedman and Mitchell Wand, Essentials of programming languages, 3rd ed., MIT Press, 2008.
- [Gra85] Ingvar Gratte, Starting with comal, Simon & Schuster (Paper), 1985.
- [Kel84] John Kelly, Foundations in computer studies with comal, The Educational Company of Ireland, Ltd., 1984.
- [Knu64] Donald E. Knuth, Backus normal form vs. backus naur form, Commun. ACM 7 (1964), no. 12, 735-736.
- [Lea86] J. William Leary, Introduction to computer programming with comal 80 and the commodore 64/128, COMAL Users Group, Ltd, 1986.
- [Lin83] Len Lindsay, Comal handbook, Brady (Robert J.) Co, U.S., 1983.
- [LO90] Thomas Lundy and Rory O'Sullivan, Beginning structured programming in basic and comal, Gill & Macmillan, 1990.
- [Seb15] Robert W. Sebesta, Concepts of programming languages (11th edition), Pearson, 2015.
- [Wir73] Niklaus Wirth, Systematic programming: An introduction, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1973.





Grafici in 3D con poche righe di BASIC

di Alberto Apostolo

Tutti conoscono Alberto Angela come abile conduttore di trasmissioni televisive italiane di divulgazione scientifica e di cultura generale. Ma chi ha vissuto gli anni '70 e '80 del XX secolo, ricorda anche il suo illustre genitore: Piero Angela il divulgatore per eccellenza.



Piero Angela ha condotto dal 1981 al 1994, "Quark", un programma trasmesso dalla R.A.I. La sigla di "Quark" era tratta dall'Aria sulla Quarta Corda di Bach con il grafico animato (Figura 2).

Anche se non sarà proprio come quello di "Quark", con qualche accorgimento possiamo anche noi costruire un grafico di una funzione $z=f(x,y)$ e magari aggiungere qualche strumento utile nello studio delle funzioni in due variabili che si incontrano in Analisi Matematica.

DALLA TEORIA ALLA PRATICA

In [MS91] si descrive in Appendice un programma per ottenere il grafico di $z=f(x,y)$ definita sul sottoinsieme $[-1,1] \times [-1,1]$ del piano XY.

Tale programma è stato usato come partenza dal sottoscritto per aggiungere successive modifiche e usare il sottoinsieme $[-A,A] \times [-A,A]$ dove A è un numero reale maggiore di zero.

Il grafico si rappresenta tracciando in modo opportuno le curve aventi equazione

$$\begin{cases} x=x_i = \text{cost.}, -1 \leq y \leq +1, i=0, \dots, 40 \\ z=f(x,y). \end{cases}$$



Figura 2, <https://www.youtube.com/watch?v=8CWOGfTdMHM>

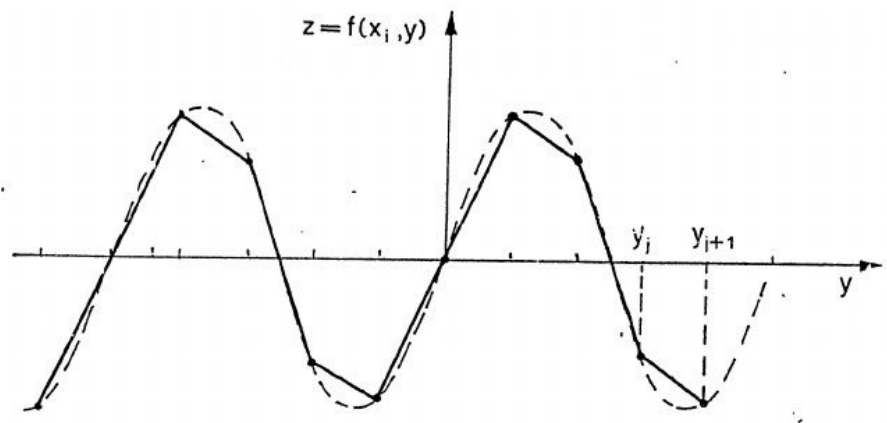


Figura 3

Sono così tracciate 41 curve su piani paralleli al piano yz (il valore 40 è stato fissato dagli autori del libro).

La curva più vicina a chi legge ha $x_0 = A$ mentre quella più lontana ha $x_{40} = -A$ (per ragioni legate al fatto che le curve più vicine nascondono quelle più lontane).

Ogni singola curva

$$z=f_i(y) = f(x_i, y),$$

si rappresenta suddividendo il sottoinsieme $[-A,A]$ in 160 intervalli equidistanti, con interpolazione lineare in ogni intervallo come nell'esempio in

Figura 3 [MS91] (anche 160 è un valore fissato dagli autori del libro). L'effetto tridimensionale è dato dalla sovrapposizione di due grafici successivi traslando il riferimento $(0,y,z)$ come nell'esempio in Figura 4 (sempre tratto da [MS91]), senza tracciare le parti di curve nascoste da quelle che si trovano in primo piano.

Per questo si introduce un sistema di coordinate (u,v) per considerare lo schermo del computer e si cerca di conoscere la relazione per passare dalle coordinate (x,y,z) alle coordinate (u,v) .

Sullo schermo il quadrato





$[-A,A] \times [-A,A]$

(visto con effetto tridimensionale) avrebbe la forma di un parallelogramma (Figura 5) con le coordinate calcolate secondo le regole del GWBASIC in modalità grafica 640x200 punti (dove l'origine si trova in alto a sinistra).

La relazione tra la coppia (u,v) e la terna (x,y,z) espressa in forma matriciale è la seguente:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

ossia

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a_{11}x + a_{12}y + a_{13}z + b_1 \\ a_{21}x + a_{22}y + a_{23}z + b_2 \end{pmatrix}$$

Per determinare b_1 e b_2 si impone che il punto $(0,0,0)$ abbia coordinate schermo $(290,100)$, una scelta dettata dal gusto personale e dalla necessità di "centrare" un po' il grafico.

I coefficienti a_{13} e a_{23} si ottengono imponendo che il punto $(0,0,1)$ si trovi alle coordinate $(290,100-K)$ dove K è un coefficiente di dilatazione dell'asse z al fine di non avere immagini troppo schiacciate.

I restanti valori a_{11} , a_{12} , a_{21} , a_{22} si ottengono tramite condizioni "furbe", cioè usando i punti $(A,0,0)$ e $(0,A,0)$.

Alla fine si avranno le formule:

$$\begin{aligned} U1 &= 290 + (-80 * X + 160 * Y) / A \\ V1 &= 100 + (40 * X) / A \\ V &= V1 - Z * K \\ U &= U1 \end{aligned}$$

Ora supponiamo di avere già tracciato n curve e di voler tracciare la curva $n+1$.

Si definisce la funzione $\max(u)$ che rappresenta il massimo delle ordinate v tracciate dalle prime n curve e la funzione $\min(u)$ che rappresenta il minimo (Figura 6).

Nel tracciare la curva $n+1$, le parti "visibili" devono avere ordinate v fuori dall'intervallo $(\min(u), \max(u))$ per ogni u appartenente

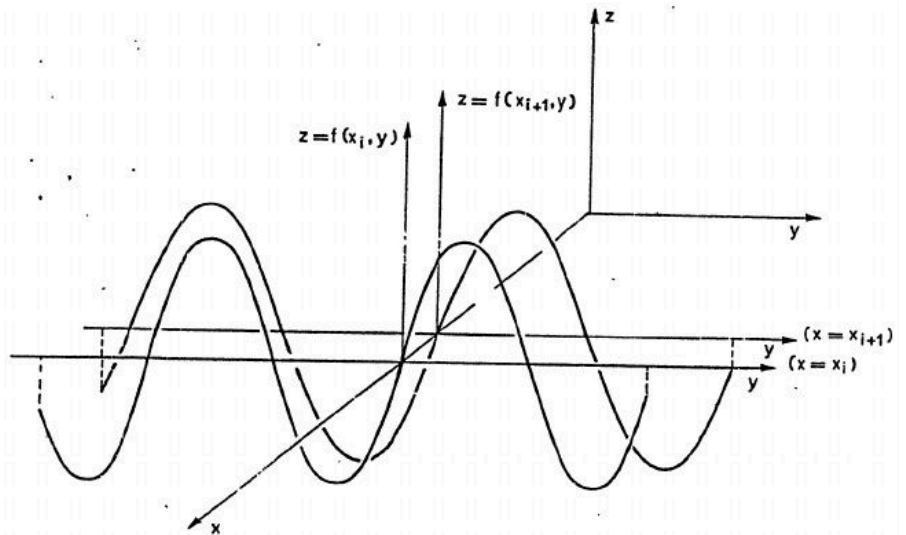


Figura 4

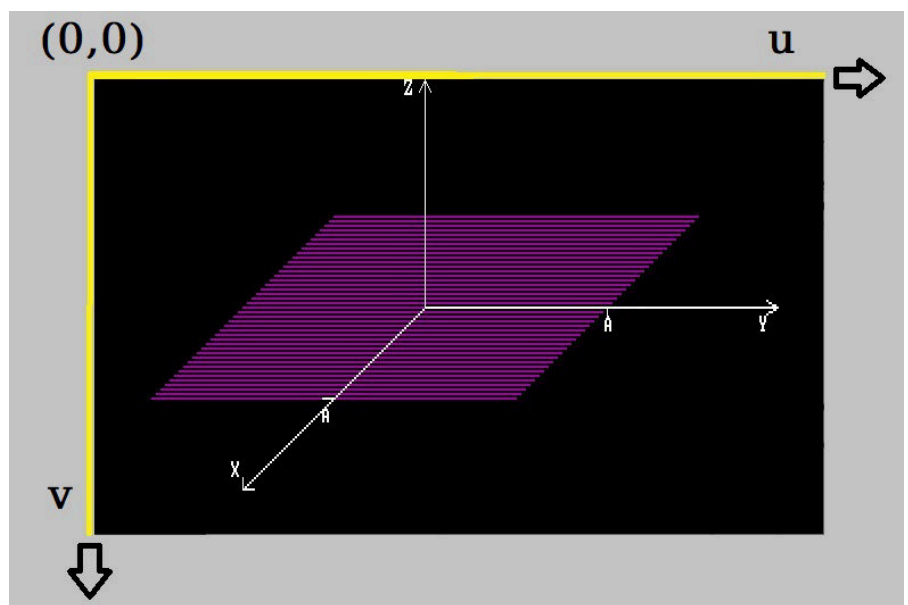


Figura 5

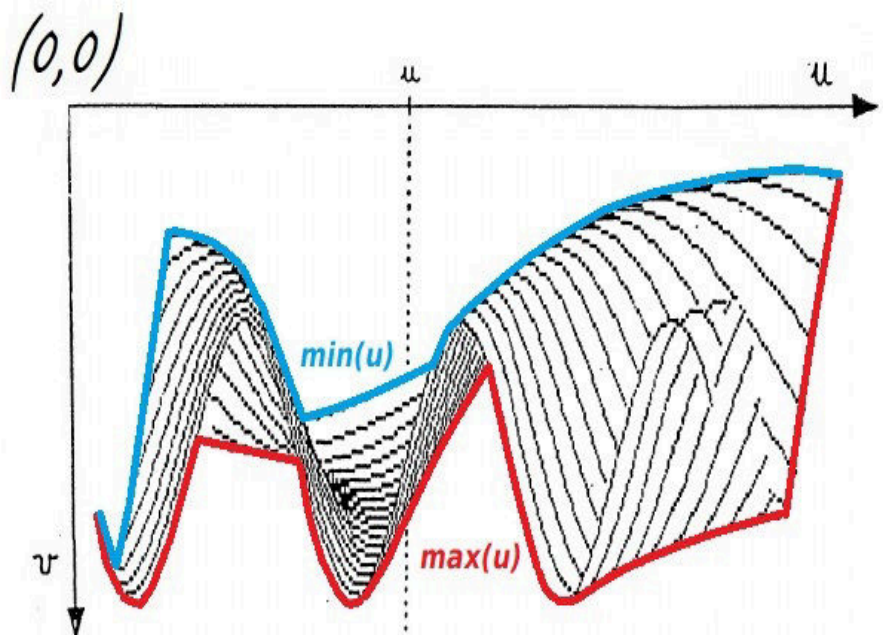


Figura 6





al dominio della curva $n+1$.

Il programma GRAF3D01.BAS in GWBASIC condensa tutte le considerazioni fatte fino ad ora (Figura 7). Di seguito sono descritte brevemente le linee del programma.

Linee 1000-1030:

dichiarazione di variabili.

Linee 1040-1080:

preparazione schermo e palette dei colori usati.

Linee 1090-1120:

inizializzazione dei parametri. La variabile A riguarda l'ampiezza del quadrato del piano xy sul quale rappresentare la funzione, K è un coefficiente di scala per l'asse z, CONSTRAINTS\$ ="Y" indica che ci sono condizioni poste sui valori di x e y (i valori di x e y non inclusi sono evidenziati in giallo), CHKONLYSSET\$ ="Y" indica che si vuole solo studiare l'esistenza della funzione (in questo caso si pone $z=0$, dove la funzione esiste il colore è viola altrimenti è verde).

Linee 1130-1180:

inizio elaborazione, dichiarazione dei cicli di iterativi (per ascisse e ordinate), dichiarazione linea di programma alla quale saltare in caso di errore nel calcolo della funzione, uso del colore viola per il grafico.

Linee 1190-1220:

zona di programma dove applicare le condizioni sui punti x e y. E' possibile inserire / cancellare / modificare istruzioni tra le linee 1201 e 1219.

Linee 1230-1250:

zona di programma dove calcolare la funzione $z=f(x,y)$. E' possibile inserire / cancellare / modificare istruzioni tra le linee 1231 e 1259.

Linee 1260-1270:

gestione studio di esistenza della funzione.

Linee 1280-1290:

gestione errore di calcolo della funzione. Per qualsiasi errore non si interrompe il programma ma si prosegue alle istruzioni successive.

Linee 1300-1330:

calcolo coordinate dello schermo.

Linee 1340-1370:

```

1000 REM GRAF3D01.BAS
1010 DEFINT C, I-J, U-V
1020 PI = 3.141592654#
1030 DIM MAX(640), MIN(640)
1040 REM SCREEN 16 COLORS 640 * 200 PIXELS
1050 REM 0=BLACK,5=PURPLE,14=YELLOW,15=WHITE,4=GREEN
1060 SCREEN 8: DATA 0,5,14,15,3: FOR I = 0 TO 4: READ C: PALETTE I, C: NEXT I
1070 REM CLEAR SCREEN
1080 KEY OFF: CLS
1090 REM PARAMETERS
1100 CONSTRAINTS$ = "N": CHKONLYSSET$ = "N"
1110 A = 6.29 : REM GRAPH OF F(X,Y) RESTRICTED TO SUBSET [-A,A] X [-A,A]
1120 K = 50: REM FACTOR FOR Z
1130 REM PROCESSING
1140 X = A: Y = -A
1150 FOR I = 0 TO 40
1160 FOR J = 0 TO 160
1170 ON ERROR GOTO 1280
1180 KOLOR = 1
1190 REM CONSTRAINTS ON F(X,Y)
1200 IF CONSTRAINTS$ <> "Y" THEN GOTO 1230
1210 IF ((X ^ 2 + Y ^ 2) <= 2) THEN GOTO 1230
1220 KOLOR = 2: Z = 0: GOTO 1290
1230 REM TYPE THE FUNCTION Z=F(X,Y) HERE
1240 T = X * X + Y * Y
1250 Z = EXP(-T/B)*COS(T)
1260 IF CHKONLYSSET$ = "Y" THEN Z = 0
1270 GOTO 1290
1280 IF ERR > 0 THEN Z = 0: KOLOR = 4: RESUME 1290
1290 ON ERROR GOTO 0
1300 REM SCREEN COORDINATES
1310 U1 = 290 + (-80 * X + 160 * Y) / A: V1 = 100 + (40 * X) / A
1320 IF ((V <> V1) OR (Z <> 0)) AND (J > 0) THEN KOLOR = 1
1330 V = V1 - Z * K: U = U1 : REM PROJECTION OF COORDINATE Z
1340 IF I = 0 OR J = 159 OR J = 160 THEN MAX(U) = V: MIN(U) = V: GOTO 1380
1350 IF V > MAX(U) THEN MAX(U) = V: GOTO 1380
1360 IF V < MIN(U) THEN MIN(U) = V: GOTO 1380
1370 C = 0: GOTO 1400
1380 IF C = 0 OR J = 0 THEN PSET (U, V), KOLOR
1390 LINE -(U, V), KOLOR: C = 1
1400 Y = Y + 2 * A / 160: NEXT J
1410 X = X - 2 * A / 40: Y = -A: NEXT I
1420 DATA 22,16,"X",14,74,"Y",1,35,"Z"
1430 FOR I = 1 TO 3: READ X, Y, A#: LOCATE X, Y: PRINT A#: NEXT I
1440 LOCATE 14, 57: PRINT A: LOCATE 19, 26: PRINT A
1450 DATA 290,100,0,-100,290,0,-5,5,290,0,5,5
1460 DATA 290,100,310,0,600,100,-10,-3,600,100,-10,3
1470 DATA 290,100,-160,80,130,180,0,-5,130,180,10,0
1480 DATA 210,140,-10,0,450,100,0,3
1490 FOR I = 1 TO 11: READ X, Y, U, V: PSET (X, Y), 3: LINE -STEP(U, V), 3
1500 NEXT I
1510 IF INKEY$ = "" THEN GOTO 1510
1520 END

```

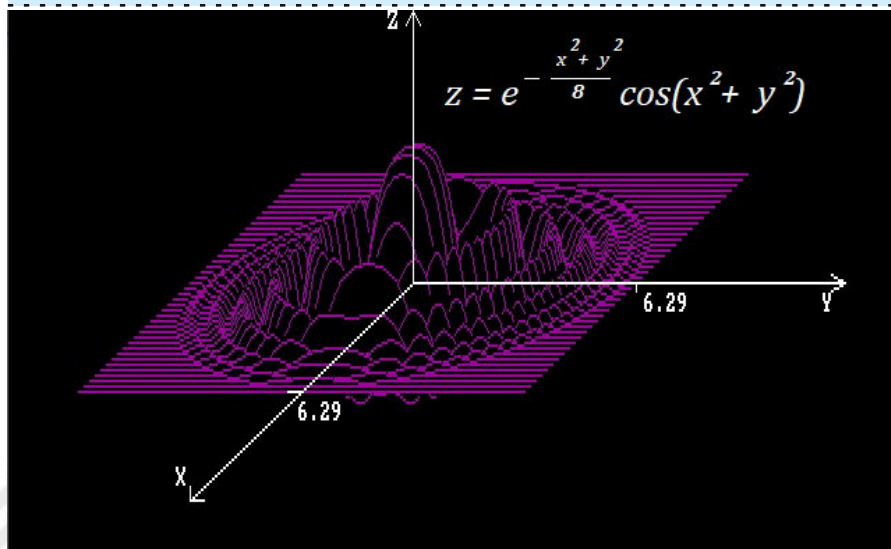


Figura 7





verifica se deve essere rappresentata o no una parte di curva.

Linee 1380-1390:

si traccia un punto e la linea retta che lo congiunge al punto precedente.

Linee 1400-1410:

incremento dei valori x e y e chiusura rispettivi blocchi iterativi.

Linee 1420-1500:

disegno degli assi cartesiani (in bianco) con ascissa e ordinata A.

Linee 1510-1520:

attesa infinita di un tasto premuto e fine programma.

In Figura 8 è riportato lo studio dell'esistenza di una funzione in una certa regione del piano xy. Modificando alcune righe del programma si può studiare $z = \text{sqr}(x*y)$ nei punti interni al cerchio di raggio 1 (bordo compreso).

Si conclude il paragrafo mostrando come "sezionare" il grafico. In Figura 9 è riportato il grafico della funzione rialzato di 1 (per osservare anche la parte negativa). Si impone la condizione $x \leq 0$ alla linea 1210 per avere la sezione con il piano $x = 0$.

Le sezioni migliori si ottengono per $x=0$ o con piani paralleli al piano $x=0$. Negli altri casi si ottengono degli effetti collaterali causati dal fatto che, per tracciare il grafico, si usano le curve $f(x_i,y)$ parallele al piano yz (Figura 10).

ROTAZIONE DEI GRAFICI

Il programma GRAF3F01.BAS offre sempre la stessa prospettiva. Si può aggiungere la possibilità di "girare" il grafico per avere un altro punto di vista e osservare particolari che potrebbero essere sfuggiti a una prima visualizzazione. Per ottenere quanto desiderato occorre applicare alle coordinate (x,y) le formule di rotazione degli assi per un certo angolo theta tale che angoli positivi corrispondano a rotazioni in senso anti-orario:

$$\begin{aligned} x_t &= +x \cos\theta + y \text{sen}\theta, \\ y_t &= -x \text{sen}\theta + y \cos\theta. \end{aligned}$$

```
1000 REM GRA3D01.BAS
...
1090 REM PARAMETERS
1100 CONSTRAINTS$ = "Y": CHKONLYSET$ = "Y"
1110 A = 1 : REM GRAPH OF F(X,Y) RESTRICTED TO SUBSET [-A,A] X [-A,A]
...
1190 REM CONSTRAINTS ON F(X,Y)
...
1210 IF ((X ^ 2 + Y ^ 2) <= A) THEN GOTO 1230
...
1230 REM TYPE THE FUNCTION Z=F(X,Y) HERE
1240 Z = SQR(X*Y)
...
1520 END
```

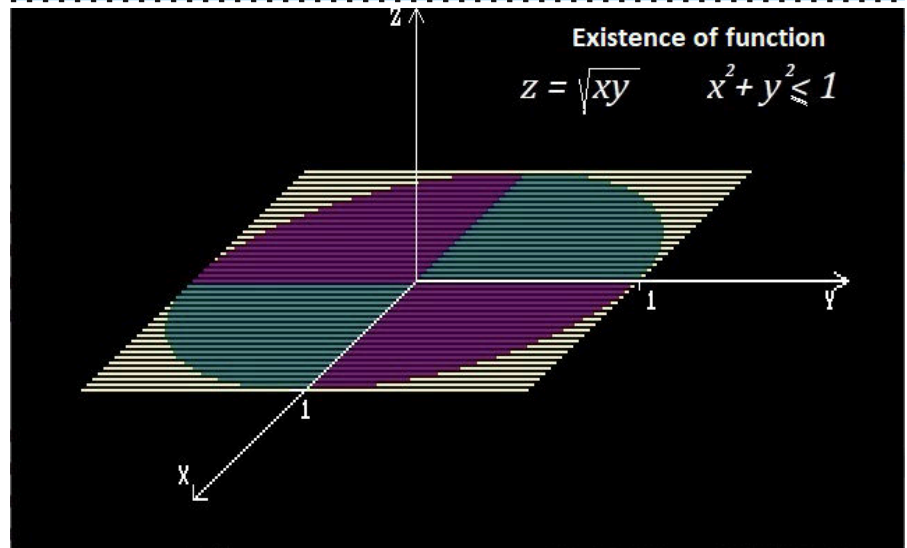


Figura 8

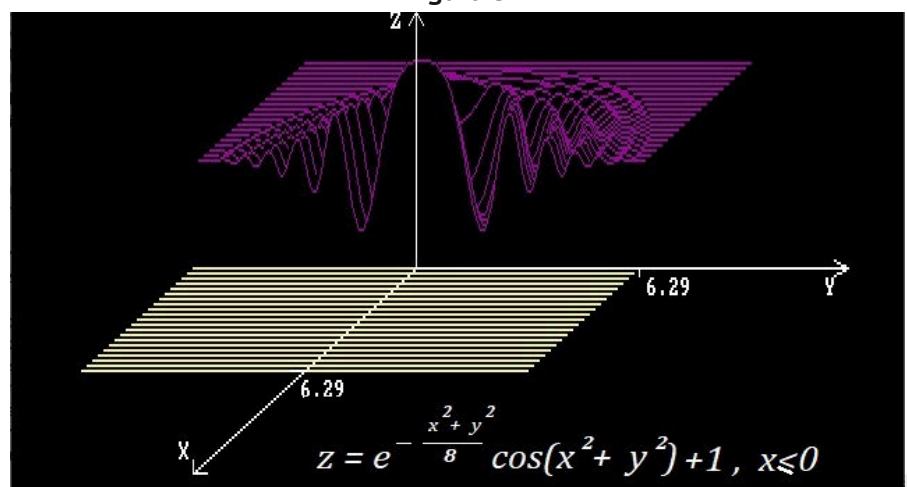


Figura 9

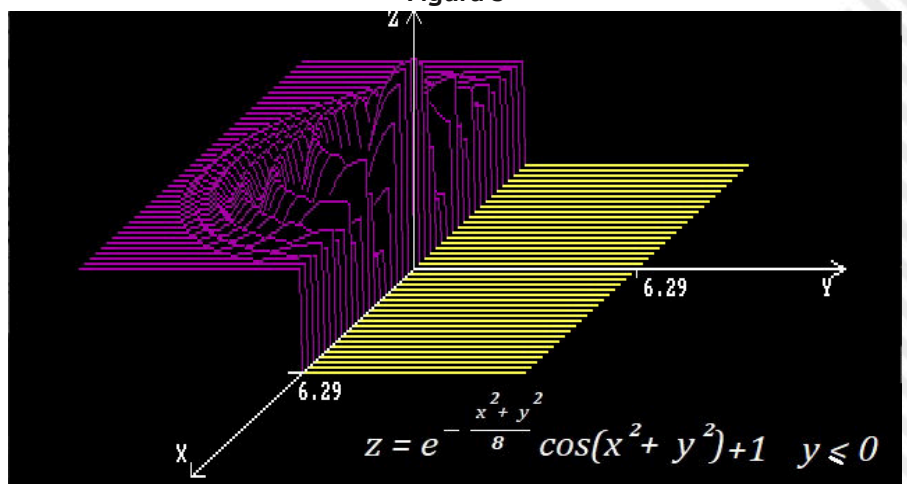


Figura 10



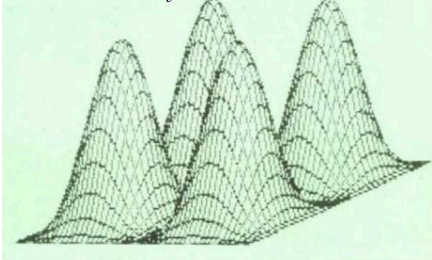


Poi si riportano sullo schermo le terne $(x,y,f(xt,yt))$ per dare l'effetto di rotazione del grafico. Anche sui vincoli occorre applicare le nuove coordinate (xt,yt) . Otteniamo così il programma GRAF3D02.BAS (Figura 11, dove in grassetto-corsivo sono evidenziate le differenze rispetto a GRAF3D01.BAS).

CONCLUSIONI

Il metodo trattato in questo articolo non è l'unico. Un altro approccio è stato usato da Belloni e Giacinti [BG84] per il loro programma su ZX Spectrum, il quale sfrutta la versatilità delle istruzioni grafiche dello ZX Spectrum e implementa un algoritmo capace di cancellare righe di superficie $f(*,y)$ e $f(x,*)$ già disegnate (per sostituirle con quelle delle linee in primo piano) senza tenere conto di cavità e intersezioni (come nella figura in basso).

$$z=(1-\sin x)*(1-\cos y)$$



Bibliografia

[BG91] A. Belloni, E. Giacinti, "TREDIM: grafica a tre dimensioni", MC Microcomputer n.29, Aprile 1984, pag.118-120. <https://issuu.com/adpware/docs/mc029>

[MS91] P. Marcellini, C. Sbordone, "Esercitazioni di Matematica", Vol. 2, Parte 2, Liguori Editore, 1991.

```

1000 REM GRA3D02.BAS
1010 DEFINT C, I-J, U-V
1020 PI = 3.141592654#
1030 DIM MAX(640), MIN(640)
1040 REM SCREEN 16 COLORS 640 * 200 PIXELS
1050 REM 0=BLACK,5=PURPLE,14=YELLOW,15=WHITE,4=GREEN
1060 SCREEN 8: DATA 0,5,14,15,3: FOR I = 0 TO 4: READ C: PALETTE I, C: NEXT I
1070 REM CLEAR SCREEN
1080 KEY OFF: CLS
1090 REM PARAMETERS
1100 CONSTRAINTS$ = "Y": CHKONLYSSET$ = "N"
1110 A = 4 : REM GRAPH OF F(X,Y) RESTRICTED TO SUBSET [-A,A] X [-A,A]
1120 K = 40: REM COEFFICIENT FOR Z
1125 THETA=-PI/4 : ROTATION OF AXIS X,Y
1126 DEF FNXT(X,Y,THETA)=X * COS(THETA) + Y * SIN(THETA)
1127 DEF FNYT(X,Y,THETA)=-X * SIN(THETA) + Y * COS(THETA)
1130 REM PROCESSING
1140 X = A: Y = -A
1150 FOR I = 0 TO 40
1160 FOR J = 0 TO 160
1170 ON ERROR GOTO 1280
1180 KOLOR = 1
1185 XT = FNXT(X,Y,THETA):YT = FNYT(X,Y,THETA)
1190 REM CONSTRAINTS ON F(XT,YT)
1200 IF CONSTRAINTS$ <> "Y" THEN GOTO 1230
1210 IF (YT<=0) THEN GOTO 1230
1220 KOLOR = 2: Z = 0: GOTO 1290
1230 REM TYPE THE FUNCTION Z=F(XT,YT) HERE
1250 Z = SQR(1-XT*XT/4 -YT*YT/9)
1260 IF CHKONLYSSET$ = "Y" THEN Z = 0
1270 GOTO 1290
1280 IF ERR > 0 THEN Z = 0: KOLOR = 4: RESUME 1290
1290 ON ERROR GOTO 0
1300 REM SCREEN COORDINATES
1310 U1 = 290 + (-80 * X + 160 * Y) / A: V1 = 100 + (40 * X) / A
1320 IF ((V <> V1) OR (Z <> 0)) AND (J > 0) THEN KOLOR = 1
1330 V = V1 - Z * K: U = U1 : REM PROJECTION OF COORDINATE Z
1340 IF I = 0 OR J = 159 OR J = 160 THEN MAX(U) = V: MIN(U) = V: GOTO 1380
1350 IF V > MAX(U) THEN MAX(U) = V: GOTO 1380
1360 IF V < MIN(U) THEN MIN(U) = V: GOTO 1380
1370 C = 0: GOTO 1400
1380 IF C = 0 OR J = 0 THEN PSET (U, V), KOLOR
1390 LINE -(U, V), KOLOR: C = 1
1400 Y = Y + 2 * A / 160: NEXT J
1410 X = X - 2 * A / 40: Y = -A: NEXT I
1420 DATA 22,16,"X",14,74,"Y",1,35,"Z"
1430 FOR I = 1 TO 3: READ X, Y, A$: LOCATE X, Y: PRINT A$: NEXT I
1440 LOCATE 14, 57: PRINT A: LOCATE 19, 26: PRINT A
1450 DATA 290,100,0,-100,290,0,-5,5,290,0,5,5
1460 DATA 290,100,310,0,600,100,-10,-3,600,100,-10,3
1470 DATA 290,100,-160,80,130,180,0,-5,130,180,10,0
1480 DATA 210,140,-10,0,450,100,0,3
1490 FOR I = 1 TO 11: READ X, Y, U, V: PSET (X, Y), 3: LINE -STEP(U, V), 3
1500 NEXT I
1510 IF INKEY$ = "" THEN GOTO 1510
1520 END

```

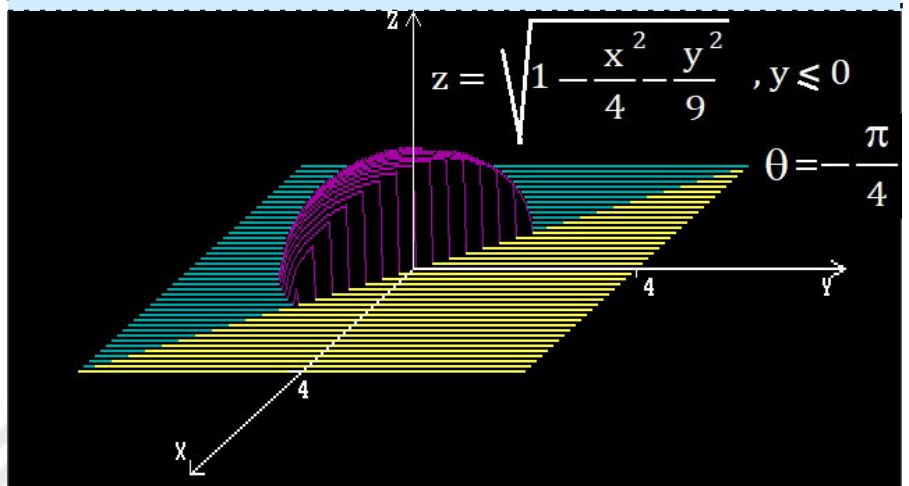


Figura 11





Giappone 11^ puntata: Un nuovo Game & Watch?

di Michele Ugolini

Bentornati cari lettori, ecco a voi la recensione del nuovo Game & Watch di Nintendo. Il titolo è corretto, questo amatissimo prodigio elettronico ha contagiato così tanti clienti che non poteva assolutamente passare inosservato durante l'ultima decade, che ricordiamo, è stata realmente prolifica nel riprodurre mini remake, soprammobili non sempre coerenti ed accrocchi acchiappa polvere più o meno riusciti.

Risorgendo dalle proprie ceneri, risvegliato nel 2009 da un fugace pisolino di un terzo di secolo, riecco a voi Ball, una versione tra le più amate al mondo dei Game&Watch di Nintendo. (cfr. figura 1)

Il 2009 è stato un anno veramente interessante in casa Nintendo. L'allora presidente era Iwata (Sapporo, 6 dicembre 1959 – Kyoto, 11 luglio 2015) ed è stato un autore di videogiochi molto produttivo nonché imprenditore giapponese. Quarto presidente della società Nintendo, ha ricoperto il ruolo dal 2002 al 2015, succedendo al longevo predecessore Hiroshi Yamauchi (una nostra conoscenza accennata nella decima puntata).

Possiamo leggere una sua interessante intervista in questo link:

<https://www.nintendo.it/Iwata-Chiede/Iwata-Chiede-Game-Watch/Iwata-Chiede-Game-Watch/5-Ricreare-Ball-30-anni-dopo/5-Ricreare-Ball-30-anni-dopo-223202.html>

E' stato veramente divertente leggere le parole di Iwata e del suo staff. Quanti stratagemmi curiosi sono stati fusi insieme, in circostanze

roccambolistiche, per poter produrre nuovamente questo oggetto da collezione.

Prendiamo il Presidente Iwata e l'intero staff: Izushi, Kano, Yamamoto.

Mettiamoli a discutere attorno ad un tavolo ed ascoltiamo questa tipologia così rara di cervelli che può far tremare la Terra: genialità ed inventiva allo stato dell'arte! (cfr. figura 2)

Per esempio, nell'intervista, si parla del nuovo Ball del 2009, lontano ben 30 anni dal Ball originale, realizzato come regalo in edizione limitata per i membri Platino 2009 del Club Nintendo.

Il Club Nintendo è un programma fedeltà di Nintendo, partito nel 2003 in Giappone. Nel Regno Unito il Club Nintendo è stato inaugurato ufficialmente nel dicembre 2007. I membri sono stati premiati con punti chiamati Stelle al momento della registrazione delle console e software Nintendo acquistati. Le Stelle si possono scambiare con i prodotti originali Nintendo disponibili nello Star Catalogo del Club Nintendo.

Perché è stato scelto proprio Ball e non un altro remake dei Game&Watch? Nell'intervista viene definito "il gioco preferito del nostro amatissimo Yokoi" (altra nostra conoscenza raccontata nella decima puntata).

Principali problemi per ricreare Ball?

L'idea per realizzarlo era chiara nelle testa di tutto lo staff, ma, al contrario della bizzarra situazione avvenuta a Yokoi descritta nella precedente puntata,



Figura 1





questa volta bisognava adattare la logica costruttiva prelevando da un'altra dimensione tutto lo scibile elettronico per poi riadattarlo ad un presente lontano anni luce dagli albori dell'elettronica degli anni 80.

Quindi regnavano due sostanziosi problemi, recuperare sia l'hardware che il software: essendo un gioco di trent'anni fa, non c'era più la relativa documentazione. È stato difficile contattare tutte le persone interessate e raccogliere il materiale.

E' stato difficile anche recuperare tutti i singoli pezzi dei Game&Watch. Non esistevano più da tantissimi anni. Quindi è stato necessario recuperare tutti gli schemi e le forme. Bisognava ricreare tutto.

Il signor Kano, in merito al software, ha dovuto recuperare tutto il know-how di tanti anni fa e relativamente all'hardware ha dovuto plasmare il feeling di pressione dei tasti e del gameplay... tutto basato su lontani ricordi!

Quando i collaboratori di Mr Kano gli chiedevano "Perché questo è così?", lui stesso, con grande umiltà, poteva solo rispondere: "Non ne ho idea!".

La sua memoria dopo trenta anni si era naturalmente affievolita.

Mr. Yamamoto, anch'esso nel gruppo addetto alla rinascita di Ball, riguardo a Mr. Kano, ha ironicamente confessato: "Meno male che lo ha fatto quando ancora se lo ricordava. Da un momento all'altro poteva dimenticare tutto, anche mentre lo stava facendo!"

Un aneddoto veramente buffo? Avete presente i due pilastri neri ai lati del personaggio Mr Game & Watch? Quelle due strisce in realtà nascondono qualcosa. Nel gioco originale di Ball, ai margini dello

schermo, c'erano dei circuiti elettrici non necessari che sono stati coperti per renderli invisibili. E' stato deciso di nasconderli con un filtro stampato. Negli LCD di oggi, non si vedono i circuiti elettrici e quindi nella riproduzione non c'è stato bisogno di nasconderli. Così è stato ricreato il più fedelmente possibile l'originale: mettendo lo stesso i pilastri. (cfr. figura 3)

Una miglioria? Nella scorsa puntata avevo spiegato quanto sia estremamente importante il silenzio in treno e soprattutto perché gli uomini d'affari degli anni ottanta non potevano maneggiare un videogioco. Ebbene, l'unica miglioria apportata in Ball del 2009 è stata la possibilità di disattivare l'audio. Niente di più!

In questa recensione voglio onorare al massimo livello l'adorabile riproduzione datata 2009. Descriverò esattamente come funziona, anche perché, lo ammetto, ogni tanto rispolvero il mio Ball remake con una rapida partita. Non parlerò invece del prezzo. E' un oggetto ormai ambito, sebbene non rarissimo e purtroppo ad oggi, 2020, il prezzo non rispecchia più alcuna logica coerente.

In Ball, un giocoliere di palline lanciate in aria deve recuperarle senza farle cadere al suolo, in progressione sempre più rapida di gameplay. Fine! Qui sta il divertimento!

Ci sono solo due pulsanti che vengono utilizzati durante il gioco. Il pulsante sinistro, posizionato sul lato sinistro dello schermo, sposta entrambe le braccia del personaggio a sinistra, mentre al contrario il pulsante destro, posizionato sul lato destro dello schermo, sposta entrambe le braccia del personaggio a destra. L'obiettivo del gioco è di destreggiarsi tra le palle e assicurarsi che non



Figura 2





colpiscono il terreno. Questo può essere fatto posizionando le mani del personaggio sotto le palle. Una volta che la mano e la palla si collegano, la palla verrà lanciata verso l'alto e curverà verso l'altro lato del corpo del personaggio dove l'utente deve posizionare nuovamente la mano in modo che i due si scontrino con successo. Quando una singola palla cade, il gioco termina e l'utente deve riavviarlo. Nel gioco A il giocatore si destreggia tra due palle e riceve un punto per ogni cattura riuscita mentre nel gioco B sono tenuti a gestire tre palle e ricevono dieci punti per ogni cattura. Il punteggio più alto possibile che un utente può ottenere è 9.999 per la partita A e 9.990 per la partita B, poiché 9.999 non è un multiplo di 10.

Il giocatore può facilmente determinare dove cadrà la palla a causa della sua velocità iniziale, che è piuttosto lenta. Man mano che il gioco procede, la palla inizierà ad accelerare ma il gioco rallenta automaticamente dopo ogni 100 punti ottenuti dal giocatore.

Ritornando sul discorso degli utenti Nintendo platino, vorrei approfondire una questione.

Nel novembre del 2009, Nintendo del Giappone ha annunciato che i membri del Club Nintendo avrebbero avuto diritto a ricevere Ball come premio. L'unità sembra quasi identica all'originale, anche se presenta il logo Club Nintendo e presenta un interruttore che consente all'utente di aumentare o diminuire il volume. Questo premio era disponibile solo per i membri del Club Nintendo in Giappone ed è stato inviato nell'aprile del 2010 a coloro che hanno accumulato abbastanza punti per raggiungere lo status

Platinum. A differenza di Nintendo of America, Nintendo ha inviato automaticamente queste unità ai giocatori che erano idonei senza la necessità per l'utente di registrarsi online. Ball è stato successivamente rilasciato sul servizio Club Nintendo di Nintendo of America, rendendolo disponibile per i fan negli Stati Uniti. Al suo rilascio, furono necessarie più monete di qualsiasi altro prodotto: 1200. Ralleghiamoci del fatto che, almeno, veniva fornita una batteria preinstallata con ogni unità.

Per gli amanti dei particolari ecco una carrellata di dettagli tecnici di Ball remake 2009:

:: Dimensioni

Larghezza:95mm, Altezza:63mm, Profondità:11,5mm

:: Peso

51 grammi (con le batterie)

:: Batteria

Due batterie (LR43 o SR43)

:: Durata della batteria

Con l'orologio, la batteria dura circa 6 mesi con batterie LR43 e 12 mesi con batterie SR43. Se l'utente gioca solo un'ora al giorno, il sistema durerà per sei mesi con LR43 (la durata della batteria con SR43 non è stata rilasciata). L'utente può notare quando la durata della batteria è esaurita : il display dello schermo diventa difficile da distinguere.

:: Input

Pulsante sinistro (controlla le braccia del personaggio e le ore dell'orologio),
 pulsante destro (controlla il braccio del personaggio così come i minuti dell'orologio),
 pulsante Gioco A,
 pulsante Gioco B,
 pulsante orario (visualizza l'ora),
 ACL (consente all'utente di modificare l'ora

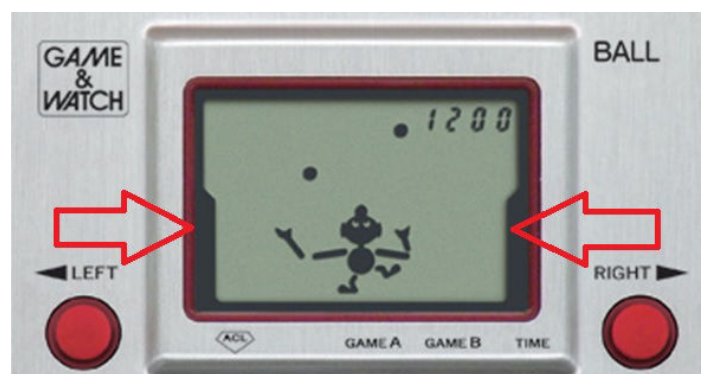


Figura 3





Figura 4

visualizzata).

:: Temperatura

Ball funziona correttamente a temperature comprese tra 50 ° e 104 ° Fahrenheit (10 ° -40 ° Celsius).

:: Precisione dell'orologio

La differenza media giornaliera è stimata in 3 secondi a temperature adeguate (vedere sopra).

:: Numero modello

AC-01.

:: Prezzo

Meglio non parlarne. Questione spinosa. (cfr. figura 4)

Anche per questa puntata è tutto cari lettori,

avrei voluto parlarvi di un altro progetto bellissimo. Purtroppo in tempo di Covid19 è stato momentaneamente parcheggiato con le quattro frecce. Peccato. Comunque, speriamo riparta tutto a gonfie vele. E' rinato Ball dopo un terzo di secolo, con grande impegno potremo ritornare alla luce anche noi, uniti in un enorme abbraccio virtuale.

Nella prossima puntata, spero tantissimo di poter parlare di questo secondo progetto, ma visti i tempi ahimè difficili, più probabilmente vi parlerò di un altro recente remake, realmente adorabile, dei Game&Watch (cfr. figura 5). A presto!



Figura 5





CHUBBY GRISTLE

Anno: 1988
Sviluppatore: Teque Software Development Ltd.
Publisher: Grandslam Interactive
Piattaforma: Commodore 64
Genere: Platform

Ci stiamo avviando verso la fine della pandemia, la primavera inizia a farsi sentire per quello che è, ossia con piogge e tempo pazzarello. Molti di noi come sempre ne approfittano per rispolverare giochi mai terminati. So di essere un po' ripetitivo ma vi racconto come è andata durante il periodo di reclusione forzata e come sta andando tuttora ai primi assaggi della meritata libertà dopo quasi tre mesi di lockdown.

In una nottata piovosa a Milano con tuoni, fulmini e allagamenti (nella speranza di aver evitato un black-out) ho cercato con gli occhi nella mia libreria di giochi un piccolo titolo per C64, uno tra i tanti che non sono passati sotto i riflettori dell'epoca: Chubby Gristle!

Già il titolo potrebbe risultare poco invitante, anzi sembrerebbe il titolo di un film comico (NdR: 'chubby gristle' lett. si traduce in 'paffuta cartilagine'). E infatti un pochino lo è, basta guardare il simpatico protagonista, un omino un po' troppo sovrappeso che si aggira per vari livelli e schermate alla ricerca del suo amato cibo.

All'epoca non lo sapevo e solo più tardi scoprii che il nome del gioco indicava appunto una persona un po' cicciottella (abbiamo imparato qualcosa di nuovo in inglese).

In pratica lo scopo del gioco è quello di attraversare le varie schermate in tre diverse ambientazioni: un supermarket, un pub ed infine il nostro salone bello imbandito di ghiottonerie.

Il gameplay ricalca il classico gioco di piattaforme in cui bisogna fare razzia di oggetti sparsi per la schermata, evitare i nemici con precisione millimetrica e fare attenzione ai salti visto che cadendo anche da un'altezza minima si perde

una delle tre vite a disposizione.

Inutile dirlo, ma a tenerci compagnia e a motivarci nel proseguire il gioco nonostante l'alta difficoltà (e quale gioco era facile all'epoca?) sarà la simpatica musichetta che difficilmente riuscirete a togliervi dalla testa.

Per poter proseguire e arrivare fino in fondo avrete bisogno di molte ore di allenamento (cosa che il protagonista evidentemente non fa!) per poterlo muovere senza cadere, evitando i nemici nei punti più complessi e prendendo tutti gli oggetti. Quindi fate molta attenzione!

Alcuni oggetti in realtà non vanno raccolti, altrimenti risulteranno nocivi più avanti, compromettendo l'intera partita! Ultima, ma non per importanza, è la grafica.

Ricorda molto i giochi dello ZX Spectrum, per via di alcune aree monocromatiche. In effetti il gioco uscì su diversi home computer, compreso il sopra citato 8-bit della Sinclair, per cui la grafica non differiva molto da quella del C64. Ma comunque vale sempre la pena di provarlo almeno una volta e perché no?, rivalutarlo.

Ricordo che era presente persino nel mio negozio di fiducia tra i titoloni che hanno fatto la storia del C64, anche se non l'ho mai acquistato.

Siamo giunti al termine (forse) della pandemia e ci stiamo preparando per uscire, vedere i nostri amici e giocare insieme a questo simpatico gioco!

Auguro a tutti un buon rientro alla vita normale e spero che usciate vittoriosi da entrambe le sfide!

di **Daniele Brahimi**



GIUDIZIO FINALE

» Giocabilità 70%

Si muore facilmente e serve parecchia pratica.

» Longevità 80%

Non molto lungo, ma impegnativo.





KNIGHTMARE SAGA

Data di pubblicazione: 1986

Pubblicazione: Konami

Piattaforma: MSX

Genere: RPG

Parlare di Knightmare (Majou Densetsu , Demon Castle Legend) non è affatto facile.

Fu prodotto dalla Konami software nel lontano 1986 per il solo standard MSX, sfruttava al massimo le capacità grafiche del processore video TMS9918/29 creando uno scrolling verticale perfetto e un uso sapiente dei 16 colori a disposizione, mentre sul lato audio usarono sì, il processore sonoro standard dei modelli MSX1 il General Instruments AY-3-8910 ma con un risultato sorprendente. La colonna sonora fu scritta e realizzata da Miki Higashino musicista di altri giochi che divennero pietre miliari come Salamander, Gadius, Ye ar Kung fu, Suidoken 1 e 2 ecc...

Il gioco ci vede nei panni di un simpatico vichingo blu dal nome "popolon" il cui compito è quello di salvare la nostra dolce principessa Aphrodite rinchiusa in un castello dal nostro acerrimo nemico. Per fare ciò bisogna attraversare ben 8 livelli pieni zeppi di mostri, scheletri, pipistrelli aventi il boss di fine livello come ciliegina.

Tecnicamente il gioco non è nient'altro che uno shoot'em up a scorrimento verticale maledettamente fluido, gli sprite dei nostri nemici sono stati disegnati con una cura a dir poco maniacale e la difficoltà calibrata in maniera perfetta anche se debbo ammettere che è davvero dura andare oltre il quarto livello. A condire il tutto i programmatori inserirono la possibilità di trovare dei bonus che ci permettevano di cambiare arma, diventare invisibili, fermare il tempo ecc... Questi comparivano rompendo delle lastre di pietra grigie disseminate un poco ovunque, alcune delle quali nascoste che comparivano dal nulla sparandovi sopra.

Il gioco ebbe un successo tale che l'anno seguente la konami produsse ben 2 seguiti: "THE MAZE OF GALIOUS" e "SHALOM-KNIGHTMARE 3 FINAL BATTLE".



Le versioni ufficiali di Knightmare ad oggi sono quella per MSX del 1986, la versione freeware in coreano realizzata dal programmatore indipendente Kim Sungsik nel 1990 per PC MS-Dos (il primo gioco coreano a supportare la grafica EGA), il Remake per MS-DOS del 1992 (realizzato dalla friends software, ovvero un gruppo di Hacker che nel 2000 hanno realizzato anche una versione di Ye ar Kung fu per PC), Wii (2009), WiiU (2014) e tutti i cellulari abilitati nel 2003.

Adirittura è uscita nel 2006 la Gold edition per MSX2, nel 2012 la versione per ZX Spectrum prodotta da climacus che ha spremuto lo ZX come un limone (<https://www.youtube.com/watch?v=MIOxC3Xjelg>) ed infine il remake nel 2014 di Nerlaska Studio vedi il video da youtube: (collage con una splendida colonna sonora in chitarra elettrica <http://it.youtube.com/watch?v=vvV2v3cpAnE>).

Curiosità sulla versione GOLD:

Questa versione è nata grazie al duro impegno di Daniel J. Caetano e dei suoi amici programmatori brasiliani della software house amusement factory.

Questo gruppo di appassionati brasiliani ha pensato proprio a tutto, infatti la versione GOLD di Knightmare è stata rilasciata in formato CD-ROM, hanno rivisto e migliorato il codice adattandolo al dos dell'MSX2, hanno pensato alla confezione, al manuale d'uso, alla grafica del cd, ecc.. insomma un gran bel lavoro.

Il gioco contiene tutti i livelli dell'originale ma con una grafica molto più curata e delle schermate aggiuntive (come quella iniziale), e persino le colonne sonore sono state totalmente rifatte e re-mixate con risultati fantastici. Dal sito ufficiale è possibile scaricare tutte le colonne





sonore in versione mp3, il manuale in pdf, la copertina del cd e l'intero gioco per MSX2 <http://www.caetano.eng.br/MSXPage/kmg/?p=5&l=en> PS: il sito è stato tradotto anche in inglese.

Come cloni che tratteremo dettagliatamente nei prossimi articoli abbiamo:

Da parte della Casio il gioco Leonidas di cui allego un video: <https://www.youtube.com/watch?v=iw2DkKWIVSI> , e da parte della Zemina ben due giochi Cyborg-Z, e WON-SI-IN (che tradotto significa uomo preistorico) questo ultimo veramente difficile, tutti per computer MSX.

Un altro gioco palesemente ispirato a Knightmare fu King's Knight della Square nel 1989 per NES <https://www.youtube.com/watch?v=D7MbzoSqqcI>

Abbiamo ancora un altro clone storico ed è quello più spudoratamente simile a Knightmare, al limite del plagio, il suo nome era THE THREE DRAGON STORY di Zemina del 1989 per Master System....date un'occhiata <https://www.youtube.com/watch?v=zovFVa-YWsk>



KNIGHTMARE REMAKE:

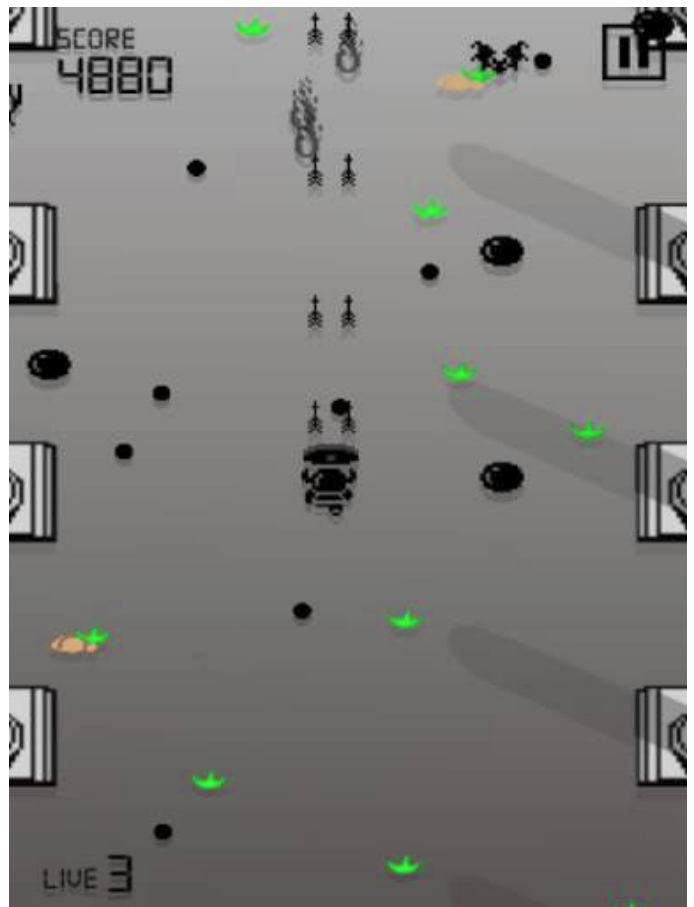
Ad occuparsi del remake di Knightmare è stato un gruppo spagnolo dal nome NERLASKA STUDIO, loro si occupano della realizzazione di videogames per PC e Smartphone.

Il gioco porta il nome di KNIGHTMARE TRIBUTE ed è un vero spettacolo, rispettoso al massimo del gioco originale, ha una grafica ed una fluidità davvero incredibili ma il punto forte è la giocabilità, praticamente la stessa presente nei computer Msx.

Inoltre i Nerlaska tempo fa avevano anche realizzato un gioco per Iphone, Ipad e Android dal titolo GAZZEL QUEST. Anche questo gioco è un clone di Knightmare realizzato molto bene, pertanto i Nerlaska dovrebbero entrare nel Guinness dei primati per essere stata l'unica software house che ha realizzato un remake ed un clone dello stesso gioco.

Adesso la Nerlaska ufficialmente non si occupa più di videogame indie, ma dalle sue ceneri è nata la <http://demonvideogames.blogspot.com/> dove trovate in vendita a prezzi popolari i suoi giochi con altri sviluppati ultimamente.

Una vera curiosità è che prima che uscisse ufficialmente il remake dei Nerlaska ora DemonVideoGames, in giro ci sono state una infinità di tentativi di remake, tantissime Beta version e un mare di realizzazioni artigianali indoor ma mai una vera e propria versione ufficiale.



Uno dei remake meglio riusciti aveva il nome di IMPETUTH; realizzato in Giappone il 29 dicembre 2008 dalla software house O-Games il gioco si sviluppa in soli 5 livelli, ma con una colonna sonora impressionante, composta da Shinji Hosoe, che prima aveva lavorato per la Namco su giochi come Dragon Spirit e Ridge Racer. Possiamo comunque ammirare su youtube un video completo, il gioco è davvero bello e in una veste inedita dove abbandona il tipico stile fumettoso del gioco MSX per apparire in maniera più accattivante e seria con marcate sfumature manga. Un altro remake indie amatoriale è KNIGHTMARE 3D, invece il progetto di un remake in 3D di Knightmare da parte della BRAIN GAMES (un gruppo di appassionati spagnoli del computer MSX che hanno realizzato i remake di tutti i migliori giochi msx, da Goonies a Maze of Galious) , non ha mai visto la luce il che è un vero peccato.

VIDEO DI IMPETUTH:

<https://www.youtube.com/watch?v=UAqPtMDAYuo>

VIDEO DI KNIGHTMARE 3D:

<https://www.youtube.com/watch?v=IX4b5C2YrY4>

Qui al momento ci fermiamo... Nei prossimi articoli continueremo a descrivervi i giochi che nacquero dopo questo capostipite e che divennero storia ludica.

di **Ermanno Betori e Alessandro Tiso**





CITY BOMBER 4K

Data di pubblicazione: 2020
Pubblicazione: Megastyle
Piattaforma: Commodore 64
Versione recensita:
 Commodore 64
Genere: Azione

Beh, chi di voi ha iniziato la sua carriera videoludica con il Commodore Vic 20 non può non ricordare quel semplice ma ipnotico gioco arcade dove a bordo di un aereo della seconda guerra mondiale si doveva far piazza pulita di una intera città pur di assicurarci la possibilità di atterrare incolumi senza schiantarci contro i palazzi.

Era il famoso BLITZ, uno dei primi sparatutto per il piccolo gioiello di casa Commodore conosciuto anche come "grattacieli" nelle compilation su cassetta disponibili in edicola in quegli anni.

L'aereo, ormai senza carburante non poteva far altro che abbassarsi gradualmente ad ogni passaggio sullo schermo con però la fortuna di poter sganciare bombe infinite attraverso le quali demolire interi palazzi. Non si poteva rallentare né virare ma solo sganciare bombe ovviamente senza nessuno strumento elettronico in grado di dirci il momento giusto. Le bombe però

non potevano essere sparate in rapida successione, il giocatore doveva attendere la detonazione di quella appena sganciata prima di poterne sganciare un'altra.

Per giocare a City Bomber 4K avremo però bisogno di un Commodore 64 o di un suo emulatore gratuito come ad esempio il VICE e oltre che molto divertente si è dimostrato godibilissimo anche dal punto di vista grafico/sonoro visto anche il suo peso di soli 4K.

Il gioco può essere scaricato gratuitamente in formato d64 ed è disponibile anche in versione fisica su cartuccia insieme a diversi altri giochi di diversi autori tutti giocabili premendo un solo tasto.

Versione digitale gratuita:
<https://megastyle.it.ch.io/citybomber-4k>

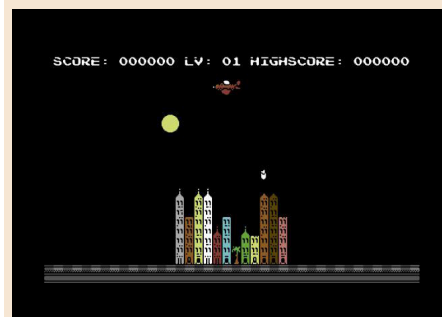
Single Button Games – Versione fisica su cartuccia:

<https://shorturl.at/hJMW2>

di **Flavio Soldani**



Blitz - versione originale VIC20



City Bomber 4K - versione C64

GIUDIZIO FINALE

» Giocabilità 89%

Un titolo per Commodore VIC 20 rivisitato egregiamente nel 2020 per Commodore 64 a dimostrazione che i veri classici non muoiono mai. Il gioco prevede anche il settaggio della difficoltà in modo da renderlo adatto a tutti i palati.

» Longevità 90%

Ovviamente adatto per partite brevi e intense come del resto quasi tutti i buoni arcade anno '80. Se però il divertimento di giocare a Blitz è rimasto invariato in tutti questi anni (l'originale è del 1982) direi che la prova longevità è stata ampiamente superata.





CLOUDY MOUNTAIN - Boardgame per un giocatore



di Starfox Mulder

Cloudy Mountain è un boardgame da giocare in solitario, ispirato ad uno dei miei videogame preferiti di sempre (Advanced Dungeons & Dragons per Mattel Intellivision). Vi servono solo 8 dadi a sei facce, una stampante, un foglio e una matita. Se aveste delle matite colorate sarebbe meglio ma non è necessario.

La scheda del personaggio

In questo gioco interpretate un arciere elfo, intento nella sua ricerca della corona del Re. Inizierete il gioco con 6 punti ferita, una faretra di 6 frecce ed una scorta di cibo per 6 giorni. Per indicare ciò posizionate 3 dadi, con la faccia in alto sul numero 6, negli appositi spazi.

- I Punti Ferita: all'interno dei dungeons potreste incontrare diversi tipi di mostri che saranno in grado di ferirvi. Se questo avverrà ruotate il dado di una posizione. In caso arrivaste a 0 punti ferita il gioco sarà terminato e voi avrete perso.

- Farette e frecce: Iniziate il gioco con 6 frecce e ne consumerete nei dungeons affrontando i mostri e ruotando il dado in una posizione inferiore per tenerne nota. Nei dungeons potreste trovare una nuova faretra (mai più di una per dungeon) e questa verrà indicata con un ulteriore dado posizionato sul 6 a fianco del precedente. Nell'inventario abbiamo al massimo spazio per 3 farette dopo di che, se ne trovassimo altre, ci limiteremo a ricaricare al massimo le scorte.

- Scorte di cibo: Ogni passo che facciamo sulla mappa di gioco ci obbliga a consumare almeno una risorsa di cibo, ruotando il dado in una posizione inferiore per tenerne nota. Nei dungeons potremmo trovare delle ulteriori scorte di cibo, cosa che ci porterà a riempire le riserve e riportarle al massimo (mai oltre il 6). Spostarsi dentro un dungeon non consuma scorte di cibo mentre all'esterno, nel caso ci trovassimo senza scorte di cibo, saremo costretti a cacciare. La caccia comporta un consumo di frecce in un rapporto 1 ad 1, pari quindi ad una freccia ogni risorsa cibo richiesta dalla casella percorsa senza possederne.

Riassumendo:

Ogni volta che è richiesta 1 risorsa cibo ma non la si possiede, si consuma un equivalente in frecce.

Ogni volta che è richiesta 1 freccia ma non la si possiede, si consuma un equivalente in punti vita.

Esplorare la mappa

Alcune parti di mappa sono già svelate e uguali in ogni partita. Il giocatore inizia dalla sua abitazione (all'estremo sinistro del foglio) e deve arrivare alla montagna nebbiosa (estremo destro). Le sessioni gialle rappresentano pianure, attraversarle comporta il consumo di una risorsa cibo.

Le sessioni blu rappresentano il fiume e possono essere attraversate solo una volta trovata la barca all'interno di un dungeon blu.

Le sessioni "cancello" sono invece invalicabili, a meno che non si possieda la chiave (recuperabile nei dungeon viola) e passando da una delle due aperture presenti.

Buona parte del gioco viene creato sul momento dal giocatore stesso quando il suo personaggio si sarà spostato su una casella che confina con zone ancora inesplorate. Spostarsi in una casella che confina con caselle inesplorate obbliga il giocatore ad esplorarle immediatamente, effettuando una serie di tiri di 2D6 per ogni casella di cui si vuol svelare il contenuto.

Un ulteriore tiro diventa obbligatorio in caso di Dungeon. Nella tabella che segue sono elencati i risultati dei dadi e quel che dovrete indicare sulla casella svelata in base alla vostra disponibilità di colori o meno. Nessuno vi vieta ovviamente di disegnare la foresta o la montagna di tutto se siete abili nel farlo.

Risultato 2D6	Mappa del mondo
2-6	Pianura
7	Foresta
8	Montagna
9-12	Dungeon

Risultato 2D6	Dungeon
2-6	Bianco
7-8	Blu
9-10	Rosso
11-12	Viola

- Pianura : Scrivete P con la matita o colorate la casella di giallo. Le pianure sono liberamente percorribili senza rischi, consumando una risorsa cibo.

- Foresta : Scrivete la lettera F o colorate la casella di verde. Le foreste possono essere percorse consumando una risorsa cibo se si possiede l'accetta. In caso contrario attraversarne una consuma un punto ferita e due risorse cibo.

- Montagna: Scrivete la lettera M o colorate la casella di marrone. Le montagne sono luoghi impervi e pericolosissimi che sarebbe meglio evitare. Se non è possibile si potranno oltrepassare solo consumando un punto ferita e tre risorse cibo.

- Dungeon: Disegnate un triangolo ed al suo interno aggiungete la lettera B (Blu), R (Rosso), V (Viola) o lasciate vuoto in caso sia bianco, anche se, potendo, colorarlo sarebbe preferibile. Entrare in un dungeon consuma una risorsa cibo.



Cloudy Mountain

	2 - 6
	7
	8
	9 - 12

	2 - 6
	7
	8
	9 - 12

		= - 1
		= - 1
		= - 1
		= - 1
		= - 1
		= + 1
		= PIENO
		= NO DANNI PROSSIMO
		= + 1P
		= USCITA
		= - 1

		= + 1P
		= - 1
		= - 1
		= - 1

	= - 1
	= - 2
	= - 3
	= - 1

= -1
 = -2
 = -3
 = -1

Cibo

Punti Ferita

Stanza Dungeon

attraversa il fiume
 attraversa il cancello
 foreste come pianure





DUNGEONS

Un Dungeon è composto da 6 stanze generate sul momento proceduralmente tirando 2d6. Per tener traccia di quale stanza stiate esplorando usate un dado e giratelo di numero di volta in volta.

Affrontare un nemico non comporta il tiro di ulteriori dadi ma direttamente le conseguenze elencate in tabella.

Nota: in caso si ottenga lo stesso risultato in più stanze, le ripetute si considerano come vuote.

Risultato Dado	Incontro-Evento	Conseguenza/Danno
Doppio 1	Mostro: Abominio	-1 punto ferita
Doppio 2	Mostro: Pipistrello	Perdi le tracce (se ne hai)
Doppio 3	Mostro: Topo	-1 freccia
Doppio 4	Mostro: Ragno	-2 frecce
Doppio 5	Mostro: Boss	-1 punto ferita; -2 frecce; tesoro
Doppio 6	Pozione di Cura	+1 punto ferita
Scarto di 1 tra i due risultati	Scorta di Cibo	Riempi la tua scorta di cibo
Scarto di 2 tra i due risultati	Tracce	Nessuna ferita dal prossimo Boss
Scarto di 3 tra i due risultati	Faretra	+1 faretra
Scarto di 4 tra i due risultati	Uscita	Trovi l'uscita
Scarto di 5 tra i due risultati	Mostro: Boss	-1 punto ferita; -2 frecce; tesoro

Boss e Tesoro: In ogni dungeon c'è un Boss e se non lo incontrate nelle prime quattro stanze occuperà necessariamente la quinta. Boss e tesoro rilasciato sono però differenti in base al colore del Dungeon.

Colore Dungeon	Boss	Tesoro
Bianco	Nessuna modifica	Faretra (mai più di una a dungeon)
Blu	Richiede una freccia in più	Barca (permette di attraversare i fiumi)
Rosso	Richiede una freccia in più	Accetta (permette di attraversare le foreste)
Viola	Fa una ferita extra e richiede una freccia in più	Chiave (permette di attraversare il "cancello")

Uscita: In ogni dungeon c'è un uscita e se non lo incontrate nelle prime cinque stanze occuperà necessariamente la sesta. Trovare l'uscita non significa per forza che la vogliate prendere, potete lasciarla per dopo e continuare ad esplorare le restanti stanze rimaste.

IL MERCANTE

Sulla mappa di gioco standard sono presenti due caselle mercante. Finendo su di esso (in termini di spostamenti si considera come una pianura) il personaggio può scegliere di vendere una o più delle sue risorse raccolte nei dungeons per ottenere in cambio una pozione di cura e recuperare uno o più punti ferita persi. Le risorse vendibili sono: una faretra (con 6 frecce, non una di meno), la barca, l'accetta e la chiave.

LA MONTAGNA NEBBIOSA

La montagna nebbiosa è a tutti gli effetti un dungeon ma contiene il doppio delle stanze (arrivati alla sesta fate ricominciare il conteggio col dado) e necessita di affrontare due Boss per raccogliere le due parti in cui è divisa la corona. I boss sono draghi a due teste e richiedono 4 frecce per essere sconfitti, inoltre infliggono 2 ferite.

La regola per cui risultati identici nella scoperta delle stanze van calcolati come "stanza vuota" non vale nella montagna nebbiosa, ragion per cui potreste ad esempio trovare due volte le tracce dei mostri e uscire illesi dai successivi due scontri.

Se non compaiono Boss nelle stanze precedenti, la undicesima e la dodicesima li contengono.

VITTORIA E SCONFITTA

Per trionfare è necessario raggiungere la montagna nebbiosa e sconfiggere i due Boss al suo interno, raccogliendo così le due parti della corona del Re. Fatto ciò il gioco termina e voi trionfate.

In qualsiasi momento i vostri punti ferita dovessero arrivare a 0, il gioco finirà con la vostra sconfitta.

CONSIGLI SU COME UTILIZZARE QUESTO GIOCO:

- Stampate a colori la mappa di gioco e inseritela in una busta trasparente in plastica, sopra cui potrete scrivere liberamente con pennarelli ad acqua e cancellare alla fine della partita.
- Se non disponete di colori appositi, stampate in bianco e nero la mappa di gioco e scriveteci sopra con una penna, una matita o delle matite colorate.
- Con la penna o un pennarello nero registrate il percorso del vostro eroe all'interno della mappa, così da ricordarvi dov'è già passato. Se si torna in un dungeon che si ha già esplorato questo verrà considerato come una pianura.
- Solo nei dungeons si trovano le scorte di cibo necessarie alla sopravvivenza, ragion per cui vi consiglio di esplorarne in buon numero.





GRAND THEFT AUTO I

In questa mia nuova serie di articoli, ci troveremo davanti a quei videogiochi che non hanno bisogno di presentazione.

Almeno una volta nella propria vita infatti, ognuno di noi ha provato cosa vuol dire emozionarsi giocando ad un titolo di questa serie targata Rockstar Games. Ma è proprio di fronte ad essa che il mondo dei videogiocatori (e non solo) si spacca a metà.

C'è chi come me che ha trascorso l'infanzia e l'adolescenza (e ancora oggi lo fa) scorrazzando per le strade di Liberty City, Vice City o per l'immenso stato di San Andreas, arrivando a fare di questa serie videoludica, un importantissimo pilastro su cui sfogare il trascorrere della vita. Altri invece hanno sempre odiato questi videogiochi, a causa della loro violenza gratuita scatenata verso ignari cittadini.

E' sicuramente risaputo il problema in cui alcuni gamers

hanno commesso omicidi, affermando di essere giocatori fanatici di questa serie (Manhunt compreso). E' ovvio però che questa è e sarà sempre un'enorme scusa per giustificare il gesto compiuto. Ebbene si... Stiamo parlando proprio di GTA!

"Grand Theft Auto" è il nome del primo titolo di questa serie, pubblicato dalla DMA Design (oggi Rockstar North) il 14 dicembre 1997 per MS-DOS, PlayStation 1 e GameBoy. L'ambientazione oscilla tra tre città diverse: Liberty City, Vice City e San Andreas, durante la quale ricopriremo il ruolo di un criminale che sarà contattato da vari boss locali per compiere missioni e lavoretti. Nulla ci vieta però di girovagare liberamente per le tre città, anche se la vista aerea sicuramente non è delle più performanti.

Grand Theft Auto è un gioco ricco di curiosità... tra le tante sicuramente spicca il fatto che, i

Anno: 1997

Sviluppatore: DMA Design

Publisher: BMG Interactive

Piattaforma: MS DOS

Genere: Action/Adventure



Fansites:

GTA Expert: www.gta-expert.it

GTA Series: www.gta-series.com

GTA Vision: www.gtavision.com

GTA Forum: www.gtaforums.com





nomi delle città in cui è ambientato, le possiamo ritrovare anche nei titoli successivi. Anche un boss della città lo ritroveremo nel capolavoro GTA 3, che è El Burro.

Troveremo anche i tanto amati Pay 'n Spray che ci aiutano tantissimo durante un inseguimento, grazie ai quali è possibile riverniciare la nostra auto. E non finisce qui...Sapevate che la stazione Head Radio fa la comparsa proprio in questo primo capitolo di GTA? E che un Donald Love fortemente invecchiato lo ritroverete proprio qui?

A differenza dei moderni GTA, in Grand Theft Auto per completare una missione, occorre ottenere un certo punteggio, utilizzabile come denaro. Le armi disponibili nel gioco sono più o meno quelle che ancora oggi utilizziamo, tra cui pistole, mitra, lanciafiamme e lanciafiamme.

La meccanica di gioco invece, è la stessa di quella utilizzata per esempio in GTA V, ovvero la presenza della polizia. Finché ci comporteremo da bravi cittadini, nessun poliziotto sarà alle nostre costole. Man mano che cominciamo ad infrangere la legge, ci ritroveremo a girovagare tra strade invase da poliziotti e poi da militari se il nostro livello di sospetto arriva a 5 o 6 stelle.

Una delle grandi pecche (oggi diremo questo) del videogioco è sicuramente la grafica scarnissima con tanto di visuale dall'alto (vista aerea), ma d'altronde cosa ci si poteva aspettare da un videogioco rilasciato oltre 20 anni fa. Non era

presente nemmeno una mappa con la quale ci si poteva orientare tra le città.

Grand Theft Auto è stato rilasciato più volte come free download dalla Rockstar Games ma la più grande notizia è che una pagina dedicata è nata sullo store più famoso di videogiochi...Parlo di Steam! Su questa piattaforma è presente proprio questo capitolo, seguito da GTA 2 e GTA London... Non è che Steam sta valutando l'idea di poter mettere in vendita questi titoli intramontabili?

Se così fosse, la prima copia scaricata sarà senza ombra di dubbio quella mia ;) E' impossibile lasciarsi scappare un'occasione del genere. Come dicevo nell'articolo su RM22 di Atari 80 Games Collection, all'epoca possedere questi videogiochi non significava nulla. Oggi invece hanno un valore inestimabile...probabilmente non in termini monetari, ma in senso affettivo sicuro!

Se possedete una copia, allora approfittate per reinstallarlo sui vostri pc. Magari sull'emulazione di Windows 98 in Virtualbox (articolo su RM22) nel caso in cui non abbiate un computer abbastanza vecchiotto.

Sicuro di aver sollevato in tutti i lettori di RM un bel polverone di nostalgia con questi miei primi tre articoli scritti per il magazine, vi auguro di poter vivere ore e ore di infinito divertimento e di ricordi rigiocando con questi videogiochi installati addirittura su un vecchio sistema operativo.

di **Marco Fiaschi**



GIUDIZIO FINALE



» Giocabilità 60%

Questo primo capitolo della serie prevede l'utilizzo di una vista aerea, la quale rende abbastanza complesso il movimento all'interno di Liberty City. Per questo motivo attribuisco al videogioco un timido 6 che, nella sua versione per pc, complicava ancor di più la perlustrazione dell'ambiente circostante. Per quanto riguarda invece la grafica e la meccanica di gioco nulla da dire. Stiamo pur sempre parlando di un videogioco del '97!

» Longevità 80%

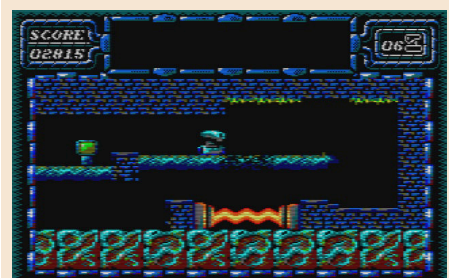
Grand Theft Auto I, secondo i puristi della serie videoludica, rappresenta un pilastro con delle fondamenta talmente possenti che è al limite dell'impossibilità creare oggi un prodotto "più contemporaneo", ma che riprenda le meccaniche originali. Non a caso, in molte classifiche di videogame dell'epoca, GTA I è sempre sul podio, oscillando tra la seconda e la terza posizione (meritatissime a mio parere), scavalcato solo da GTA 3 e GTA Vice City. Considerando che la Rockstar Games sia ancora molto affezionata a questo titolo, e probabilmente anche molti di noi, una votazione pari a 8/10 sia più che meritata!





TRAKERS

Sviluppatore: Gary e Scott Kennedy
Anno: 1991
Piattaforma: Amstrad CPC
Genere: Esplorazione/Enigma



Nel 1986 uscì al botteghino un film intitolato Corto Circuito (Short Circuit) il cui protagonista era un robottino dai tratti semi-umanoidi. Il film riscosse un discreto successo in Italia e nel mondo, grazie all'interprete principale ed al suo comportamento, simpatico ed amichevole.

La trama del film è semplice, ma allo stesso tempo intrigante; durante un'esibizione, uno dei 5 robot creati dalla NOVA Robotics, a causa di un corto circuito generato da un fulmine, si ritrova a vagabondare libero nel mondo e piano piano ad acquistare coscienza di se stesso...

Solo a me il robottino di Trakers ricorda **Numero 5**, il protagonista di Corto Circuito? Probabilmente no, perché anche la storia è in qualche modo simile.

Durante una missione esplorativa, Trakers, questo il nome del protagonista del nostro gioco, a causa di un disguido, viene dimenticato dall'equipaggio dell'astronave Starl da solo sul pianeta Magellanx.

Trakers, proprio come Numero 5, è dotato di cingoli con i quali può

muoversi su qualsiasi terreno e compiere lunghi salti per evitare eventuali ostacoli. La fortuna di Trakers è che su Magellanx è presente un sistema automatico che potrà riportarlo sull'astronave Starl, ma per attivarlo ha bisogno di 10000 crediti! Oh, persino su uno sperduto pianeta ai limiti della galassia i soldi fanno la differenza...

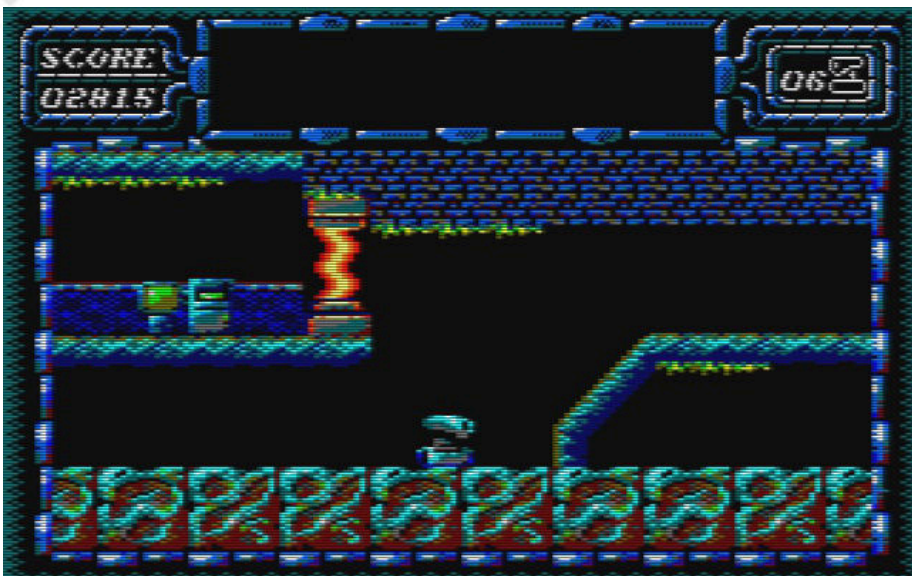
Così il nostro robot si mette alla ricerca di questa somma per potersi ricongiungere con i suoi amici. Io sarei rimasto su Magellanx a fare la bella vita, ma Trakers sembra essere di diversa opinione e quindi il nostro compito è quello di aiutarlo in quest'impresa.

Il gioco

Trakers è un gioco di esplorazione a schermata fissa, dove dobbiamo raccogliere oggetti per risolvere gli enigmi e proseguire nell'avventura.

Ovviamente il percorso esplorativo è reso più difficoltoso dalla presenza di svariati alieni il cui solo tocco risulta letale per Trakers; ma non solo, il pianeta è costellato di sabbie mobili, liquami corrosivi e trabocchetti vari. Insomma, Trakers non ha vita facile





su Magellanx... Per fortuna però ha dalla sua una scorta iniziale di 9 vite con le quali può affrontare tutti i pericoli con relativa sicurezza.

Tutto sommato il gioco è accattivante, la grafica non è niente di che, ma è sufficientemente dettagliata da invogliare il giocatore a vedere cosa l'attende nella schermata successiva.

I colori sono usati sapientemente nelle diverse ambientazioni, interno, esterno, caverne... E l'effetto è decisamente realistico.

Il gioco inoltre ha una particolarità: l'area riguardante il punteggio ed i messaggi dei computer che via via esploreremo, è realizzata in MODE 1, l'area di gioco vera e propria, quella dove si muove Trakers, è in MODE 0.

Il sonoro è completamente assente, fatto salvo per alcuni effetti speciali presenti qua e là nelle diverse locazioni. Personalmente lo apprezzo!

Dopo i primi enigmi facilmente risolvibili ed aver visitato le prime schermate senza perdere nessuna vita, vi ritroverete a perdere diverse vite soltanto per capire come affrontare un determinato passaggio. Inoltre dopo i primi oggetti facilmente identificabili, vi troverete in mano degli utensili e non saprete cosa farne. Vi occorrerà del tempo per capire cosa usare e dove, senza però mai avvertire quel senso di frustrazione che vi farebbe smettere di giocare.

Conclusioni

Se siete amanti dei giochi di esplorazione in cui dovete raccogliere oggetti e capire dove usarli e non vi spaventa visitare

più di una volta le stesse schermate, troverete Trakers decisamente attraente. Se invece preferite menare le mani o blastare alieni a tutto spiano, statene alla larga, qui non c'è ombra di violenza.

Homebrew

Trakers nasce nel 1991 come un gioco homebrew realizzato dai fratelli Gary e Scott Kennedy. Inizialmente i due avevano suscitato l'interesse di alcune software house, ma alla fine decisero di vendere il gioco direttamente, senza intermediari, per massimizzare i guadagni. Vista la qualità del gioco non credo che abbiano fatto fatica a venderne diverse copie.

Dove recuperarlo

Il gioco può essere scaricato da qui: <https://www.cpc-power.com/index.php?page=detail&onglet=pub&num=22>

Come proseguire nel gioco

Per proseguire nel gioco avrete bisogno di una buona memoria o di disegnare una mappa!

Che ne dite di farne una ed inviarcela per aiutare gli altri giocatori?

La pubblicheremo volentieri, come abbiamo fatto con quella dell'Abbaye des Morts.

Serve aiuto?

Se invece rimarrete bloccati da qualche parte, qui potete trovare un longplay di Trakers: <https://www.youtube.com/watch?v=Z2K3NurOyy4>

Più di un ora di gioco! Ce n'è da stare occupati per un po'.

Buon divertimento!

di **Francesco Fiorentini**

GIUDIZIO FINALE

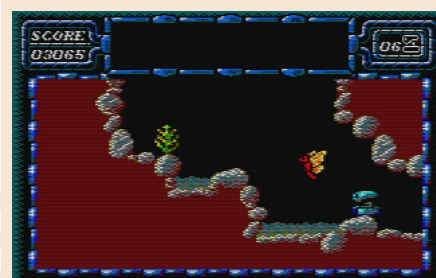
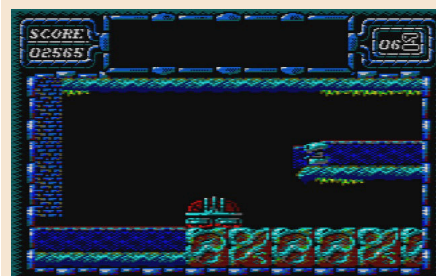


» Giocabilità 80%

Un gioco onesto, ben realizzato dal punto di vista tecnico. Se amate i giochi di esplorazione lo adorerete.

» Longevità 90%

Inizialmente semplice, la difficoltà aumenta dopo i primi 2 o 3 enigmi... Più di 100 schermate da visitare e circa un ora effettiva di gioco necessaria per completarlo. Serve altro?





CRYSTAL CASTLES

Publisher: Psytronik
Anno: 2020
Piattaforma: Commodore64
Genere: Action/Adventure

Parlare di collezionismo oggi giorno è diventata un'attività "fantasma". Sfiderei chiunque di voi lettori per sapere se anche voi siete mai stati presi per matti quando avete cominciato a collezionare oggetti appartenenti al vostro passato, o addirittura al periodo antecedente la vostra nascita.

Da alcuni anni anch'io ho cominciato a fare collezionismo, in particolar modo quello riguardante cd musicali e videogiochi per pc. Per fortuna non è il caso mio quello riportato qualche riga più su, ma c'è mancato poco... D'altronde agli occhi delle altre persone appariamo come dei matti, poiché pensano che non è normale collezionare vecchi oggetti. Ma è proprio qui che si sbagliano. Questi articoli che sto scrivendo (e che spero di scriverne in futuro per altre centinaia di volte) per RetroMagazine, riguardano grosso modo il mio passato.



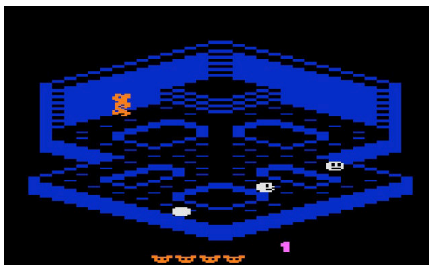
E certamente non potevo farmi mancare l'articolo sul tanto amato orsacchiotto Bentley di Crystal Castles. Ecco, prendiamo in esame proprio questo videogioco... Molti sicuramente penseranno "Ma questo ragazzo a 22 anni ancora gioca con Bentley?" e tante altre cose. Beh, la risposta è SÌ.

Credo che per noi collezionisti, sia un'emozione sempre più forte poter rivivere quegli attimi vissuti in particolari momenti della nostra vita. Tra tutti questi, c'è

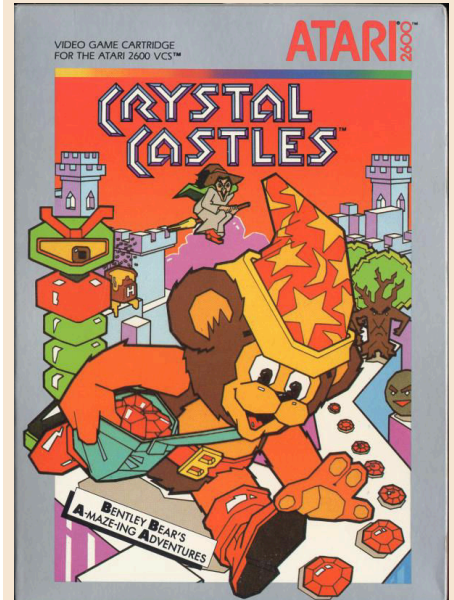
sicuramente l'avventura di Franz X Lanzinger. Ovviamente non ero nato quando uscì il cabinato di Crystal Castles, ma ho potuto giocarci per la mia prima volta sul disco Atari 80 Classic Games In One (Articolo su RM22).

Lasciato in un cassetto per anni, e ritrovato poche settimane fa, questo disco mi ha permesso di rivivere gli attimi in cui ero solo un bambino spensierato e un accanito gamer primordiale di videogiochi per pc.

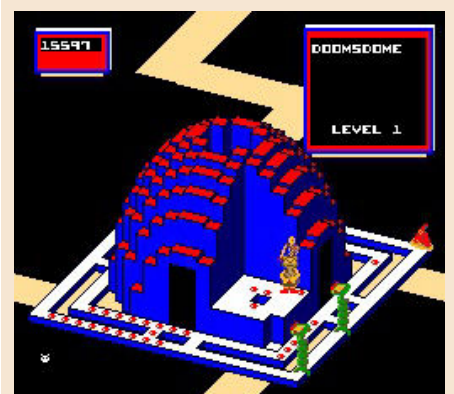
L'avventura che più mi ha colpito è proprio quella di Bentley, l'orsacchiotto che doveva sconfiggere la strega Berthilda attraversando 9 livelli in 4 castelli diversi, caratterizzati dalla presenza di un numero sempre crescente di nemici, e di una difficoltà sempre maggiore. Nel gioco infatti era possibile incontrare mostri di ogni tipologia: Crystal Balls, ovvero palline che rotolando sul terreno di gioco, mangiano le gemme di Bentley e... anche lui se non si presta abbastanza attenzione, streghe, scheletri danzanti, alberi, sciame di vespe ecc... Insomma, il gioco non si fa mancare proprio nulla!



Crystal Castles fu rilasciato anche per Atari 2600 e C64 ma, per ovvie ragioni, era caratterizzato da una grafica molto scarna. Infatti il livello era caratterizzato dalla presenza di figure geometriche astratte nelle quali dovevamo



Cover ATARI e cabinato





muoverci per prendere le gemme. Il suo successo però fu scaturito dalla vendita di ben 5380 cabinati, considerati i migliori mai realizzati da Atari. Presentavano una trackball al centro e i pulsanti Start e Fuoco su entrambi i lati.

Per quanto riguarda le curiosità di questo gioco, possiamo trovarne davvero tante. Quella che sicuramente spicca maggiormente sulle altre, è la particolare forma della struttura dei castelli che, se osservate attentamente, notiamo che sono in realtà le iniziali dei



nomi e dei cognomi degli amici dello stesso Lanzinger.

Con il tempo però si è venuti a sapere che le iniziali in realtà si riferivano ai nomi e ai cognomi dei giocatori più leggendari del gioco stesso.

Nel 2012 è iniziata la progettazione di un videogioco a scorrimento laterale (stile Super Mario) con i personaggi di Crystal Castles, durante i quali Bentley dovrà recuperare 5 gioielli preziosissimi che ha rubato la strega Berthilda.

Ultima, ma non per questo meno importante delle altre, troviamo la curiosità riguardante la leggenda di Crystal Castles, ovvero Frank Seay, il giocatore che nel 1998 raggiunse il record planetario di 910722 punti, e mai più nessuno riuscì a batterlo.

Forse sarà grazie a lui che



sappiamo come termina in realtà il gioco. C'è chi sosteneva che il gioco fosse in realtà infinito, e chi invece credeva che il gioco fosse basato su un unico livello che mutava ogni volta che si raccoglievano tutte le gemme. In realtà il gioco una fine ce l'aveva eccome.

Forse dopo tutta quella faticaccia ci saremo aspettati qualche riconoscimento in più, ma dipende soprattutto dal numero di vite a disposizione durante il completamento del livello 10. Una cosa è sicura... la frase "I GIVE UP: YOU WIN" non ce la toglie nessuno!

Ed arrivare ad ottenere questa schermata vuol dire essere entrati di diritto nella Hall of Fame di Crystal Castles!

di **Marco Fiaschi**

GIUDIZIO FINALE

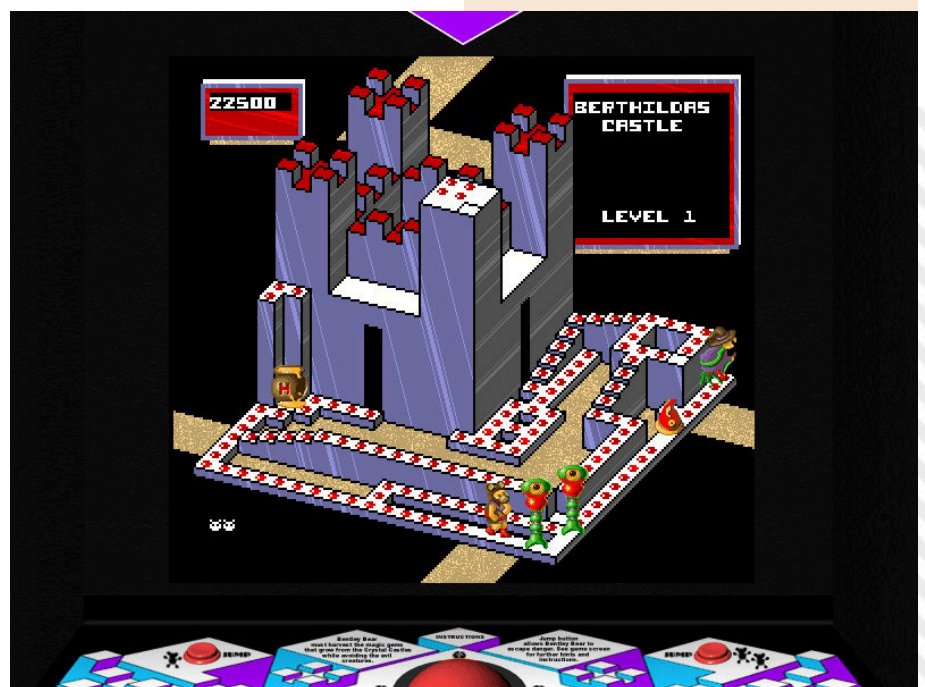


» Giocabilità 70%

Ricordo che in questo articolo è stato fatto affidamento alla versione per PC contenuta in Atari 80 Games Collection In One (purtroppo non abbiamo a disposizione un cabinato :). E' possibile procedere tra i vari livelli utilizzando esclusivamente un mouse, il che rende Bentley abbastanza controllabile. Se invece parlassimo della versione originale per cabinato, non arriverebbe nemmeno alla sufficienza, a causa della presenza di una trackball estremamente sensibile e deteriorabile. Molti giocatori hanno trovato Crystal Castles difficilissimo da giocare sul cabinato.

» Longevità 60%

Anche in termini di longevità Crystal Castles purtroppo non eccelle. Stiamo parlando di un videogioco ormai caduto nel dimenticatoio. Sebbene il titolo fosse presente in ogni collezione di Atari, nel XXI secolo purtroppo Crystal Castles non è entrato in modo trionfante (nonostante oggi è ancora considerato uno dei migliori cabinati mai realizzati). invito tutti i lettori di RM a documentarsi su questo videogioco e a farsi una bella partita. Altro che giochi moderni! Attenzione però... le avventure di Bentley provocano dipendenza!





L'ANGOLO DELLE OSCURITA' SEPTERRA CORE

Sviluppatore: Valkyrie Studios
Anno: 1999
Piattaforma: PC CD-ROM
Genere: RPG

Bentornati alla rubrica dedicata ai giochi "particolari"; in questa puntata parliamo di Septerra Core, un titolo piuttosto singolare, frutto di una lunga e difficoltosa gestazione. Tutto ha inizio negli studi di Rabid Entertainment, software house in precedenza conosciuta come ICOM Simulations (autori di Shadowgate, i due Dèjà Vu e la serie Sherlock Holmes Consulting Detective). A differenza dei precedenti titoli, Septerra Core è un RPG occidentale, ma realizzato con uno stile ispirato ai classici nipponici del genere disponibili su console. Rabid Entertainment però è in difficoltà finanziarie ed è costretta a chiudere i battenti; parte del team fonda allora Valkyrie Studios e, grazie al supporto della mitica Monolith Productions, porta a completamento il gioco. Purtroppo il titolo si rivela un insuccesso commerciale e causa la prematura dipartita dello studio.

Septerra Core è ambientato su un pianeta dalla forma unica, con sette continenti che ruotano attorno ad un nucleo centrale, collegati tra loro tramite una sorta di colonna vertebrale bio organica, che ne regola il movimento. Il Creatore (la divinità che ha forgiato il mondo) ha lasciato due chiavi per poter accedere al nucleo, che ogni cento anni viene attivato dalla luce del sole che filtra attraverso un varco temporaneo tra i continenti. La leggenda dice che chi sarà in grado di scoprire i segreti celati all'interno, riceverà il Dono del Creatore e la conoscenza per accedere al Regno del Cielo.

In passato un angelo decaduto tentò di entrare nel nucleo, rubando le chiavi; il Creatore fu allora costretto a far scendere su Septerra il proprio figlio Marduk. Dopo una lunga battaglia, il giovane uccise il demone con l'aiuto di un'armata composta

dai guerrieri dei sette continenti, rimanendo poi a vivere sul pianeta e dando luce alla stirpe dei Prescelti. Sono passati molti secoli da allora, ma il desiderio di scoprire il segreto del Nucleo affascina i discendenti di Marduk. Doskias, generale dei Prescelti, decide allora di recuperare le chiavi e forzare l'apertura, incurante dei pericoli che porterà al mondo intero. Nei panni di Maya, una giovane orfana che vive nella città di Oasis, nel secondo continente (posto proprio sotto a quello dei Prescelti), dovremo cercare di sconfiggere il tiranno, che scopriremo essere anche stato la causa della morte dei nostri genitori.

Graficamente Septerra Core presenta un mondo di gioco vasto ed accattivante, totalmente realizzato in grafica bidimensionale pre-renderizzata. I fondali sono estremamente curati, ricchi di dettagli e particolari in movimento. I personaggi sono anch'essi bidimensionali e, sebbene non siano animati in maniera eccelsa, fanno il loro dovere a livello estetico, presentando anche volti dalla buona caratterizzazione durante le fasi di dialogo. Il mondo di gioco è inoltre arricchito da un sistema di luci ed ombre dinamico di buona fattura, oltre che da un buon comparto di effetti durante le battaglie. A questo si aggiunge la presenza di coreografiche sequenze in computer grafica, che sottolineano i momenti salienti della storia.

Il comparto audio si avvale di una colonna sonora di qualità discreta composta da Martin O'Donnell (responsabile delle musiche di molti titoli di casa Bungie), che alterna temi di ottima fattura per le fasi di esplorazione della mappa principale a monotone composizioni per gli scontri. Gli effetti sono di buon livello, pur non raggiungendo picchi di





GIUDIZIO FINALE



» Giocabilità 80%

Grazie ad uno stile ispirato ai classici RPG nipponici, il gioco offre una riuscita alchimia tra esplorazione e scontri a turni.

» Longevità 90%

Il titolo ha una durata assolutamente consistente, come da tradizione del genere. Se siete pronti a dedicargli il tempo che merita, vi aspetta un'esperienza lunga e coinvolgente.



eccellenza. Quello che invece sorprende è il parlato: ogni singolo personaggio è infatti doppiato, con una qualità che varia dal buono all'eccellente. Questo aiuta ad avere una caratterizzazione più efficace e rappresenta un grande passo avanti rispetto alla concorrenza dell'epoca.

Septerra Core è un gioco di ruolo in stile giapponese che alterna fasi di esplorazione, dialogo e risoluzione di enigmi, a sequenze di combattimento a turni. Possiamo dire che, con le dovute differenze, se avete giocato ad un episodio classico di Final Fantasy, non ci metterete molto ad ambientarvi. Sono presenti tutti gli stilemi classici del genere, dall'equipaggiamento alle magie, qui rappresentate da carte simili a tarocchi, che traggono energia dal nucleo del pianeta e che si possono persino combinare per effetti più devastanti.

Il sistema di combattimento è interessante e, nonostante la lentezza degli scontri (dovuta all'impossibilità di saltare le animazioni), risulta piuttosto immediato e divertente, grazie alla possibilità di poter portare tre tipi diversi di attacco, a seconda del tempo di attesa.

E' interessante notare inoltre che i nemici sono sempre visibili su schermo nella fase di esplorazione, nonché costantemente in movimento. Questo ci permetterà di poter evitare qualche scontro di troppo, se lo vorremo. Altra caratteristica di rilievo è la presenza di una vera e propria mappa in tempo reale della zona dove ci troviamo, richiamabile in qualunque momento con il tasto TAB. L'interfaccia non è comodissima, soprattutto nel menù di

gestione dell'inventario e dell'equipaggiamento, ma alla fine ci si fa l'abitudine.

Il titolo è interamente giocabile con il mouse, ma in caso di preferenza possiamo utilizzare anche la tastiera per muovere direttamente il nostro personaggio. La durata del gioco è assolutamente ragguardevole; si parla in totale di circa 55-60 ore di gioco, che senza dubbio rappresentano un buon traguardo per una produzione di questo tipo. Il livello di difficoltà rientra nella media del genere, anche se presenta alcuni picchi più complicati del previsto. Fortunatamente si può salvare in ogni situazione e questo aiuta ad evitare il senso di frustrazione.

Nonostante sia oggi disponibile su Steam e Good Old Games, Septerra Core rimane sostanzialmente sconosciuto al grande pubblico, anche se nel tempo ha trovato riscontri positivi in una piccola cerchia di appassionati. Se vi piacciono i classici RPG a turni in stile giapponese, con una narrativa solida ed un buon impianto estetico, potrebbe fare al caso vostro, a patto che conosciate l'inglese, visto che purtroppo non esiste una versione italiana. Non dimenticate, inoltre, di aggiornare (nel caso abbiate il cd originale) il titolo con la patch 1.04, in modo da evitare fastidiosi bug che potrebbero compromettere l'esperienza.

È arrivato il momento di salutarci, ma ricordate il nostro motto: "Stay Hungry, stay Obscure!"

Alla prossima!

di **Federico "Arzak1" Gori**





SONIC THE HEDGEHOG

Editore: Sega
Sviluppatore: Sega Sonic team
Piattaforma: Sega Megadrive, Sega Master System, Game Gear, Virtual console Nintendo
Versione recensita: Sega Megadrive
Genere: Platform

È ancora divertente correre nei panni di Sonic, anche dopo 29 anni?

Sono passati veloci questi anni, dal 1991 al 2020. Veloci e sfocati come l'eroe blue di Sega. E' difficile credere che sia passato così tanto tempo dalla sua uscita (sembra ieri, ndr) da quando il riccio creato da Yuji Naka e il Sonic Team, si è lanciato per la prima volta sulla scena videoludica.

Sonic the Hedgehog, conosciuto anche come Sonic 1, è proprio il primo prodotto dove la mascotte Sega fece il suo debutto. Arrivo dentro i nostri megadrive/genesis nel caldo giugno del 1991 accompagnato da una aggressivissima campagna di marketing e creando una leggendaria rivalità con Mario di Nintendo aprendo le danze alla console wars degli anni 90.

Il suo gameplay fresco e, soprattutto, il suo incredibile senso di velocità hanno lo hanno reso un classico e uno standard per i giochi di piattaforme del periodo.

Proposto inizialmente per il 16bit Sega, venne subito convertito con una versione totalmente diversa ma davvero avvincente anche per il Master System e per il portatile Game Gear. Successivamente anche su Gameboy advance e sulle console virtuali di Wii e WII U. Ha generato decine di sequel e di spin off, ma la magia dei primi tre capitoli e la loro innovatività.

Sonic, il personaggio, è nato per necessità. Sega vide che Nintendo stava uccidendo con il NES e poi con il Super Nintendo il mercato attraverso la gamma di platform con protagonista Mario, e decise di sostituire le sue mascotte del tempo (chi si ricorda di Alex Kidd? Ndr) con qualcosa di diverso. Un eroe diverso da Mario, più

dinamico, sfrontato e decisamente più veloce.

Mentre lo sviluppo da parte del team si spostava verso un gioco incentrato esclusivamente sulla velocità, i personaggi candidati erano ridotti a una manciata di disegni di animali; animali che potevano potenzialmente correre molto velocemente e attaccare senza perdere il loro slancio. La squadra si focalizzò su due tipi di creature che potevano far rotolare i loro corpi in una palla: un armadillo e un riccio.

L'idea era che la forma della palla potesse servire da attacco rotolante e saltante senza rallentare il senso di velocità del gioco. Il design dell'armadillo non fu completamente abbandonato, poiché in seguito divenne la base del personaggio dell'universo di Sonic, Mighty the Armadillo. Ma l'idea del riccio vinse, dando origine allo stesso Sonic.

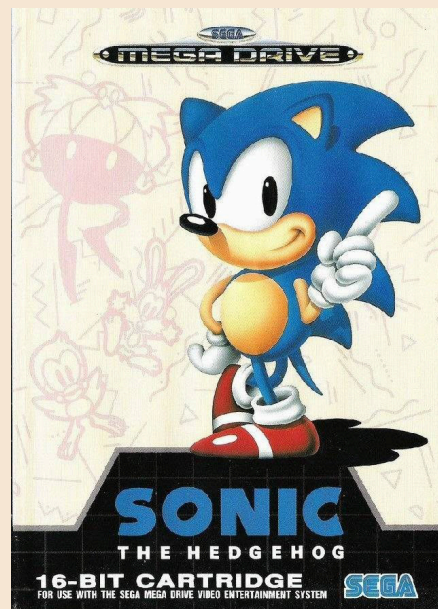
In questa prima uscita, Sonic si trova ad opporsi alle forze meccanizzate del Dr. Ivo Robotnik.

Lo scienziato pazzo ha conquistato l'Isola del Sud, racchiudendo la sua popolazione di animali in gusci cibernetici mentre cerca di recuperare i leggendari Smeraldi del caos.

Guidando Sonic, tocca al giocatore farsi strada attraverso l'esercito di animali robotici, raccogliere gli smeraldi per te stesso e fermare gli schemi di Robotnik una volta per tutte.

Tuttavia, Sonic non era e non è tutt'ora un semplice gioco di piattaforme salta-raccogli-calibra-fai attenzione ma è il primo esempio di running platform.

La maggior parte delle fasi del gioco sono costruite per mostrare la velocità





del personaggio e sono piene di molle, pendenze e loop per inviare Sonic a velocità sempre più elevate. Lungo la strada raccoglierai anelli sparsi che fungeranno da "energia vitale". Fintanto si sta tenendo almeno un anello, una collisione con un nemico o un rischio ambientale non ucciderà il nostro eroe.

Ci sono anche altre cose da afferrare: scudi protettivi, invincibilità temporanea. Anche scarpe da corsa ultrarapide che fanno precipitare Sonic a un ritmo ancora più implacabile. Ogni mondo è diviso in tre atti al cui termine si affronterà il malefico scienziato pazzo e una delle sue macchine di morte.

Non è un numero eccessivamente elevato di fasi, ma il gioco nel suo insieme è altamente rigiocabile - anche dopo 29 anni.

Come è invecchiato? Bene!

La grafica è ancora una gioia per gli occhi sia nella versione 16bit che nella versione 8bit soprattutto se giocato su un vecchio TV CRT.

Poche persone si rendono conto di quanto sia stato difficile creare un motore grafico come quello del gioco, che ha permesso di avere un'incredibile velocità e una grande varietà di animazioni, oggetti, sprites e effetti di parallasse. Il risultato era stupendo nel '91 e fa ancora una grande figura oggi.

Verdetto

Sonic the Hedgehog è un meraviglioso gioco classico che deve essere provato dalle nuove generazioni e rigiocabile da noi appassionati con la stessa passione del 1991.

Chiudo citando la bellissima versione per Master System, i programmatori hanno riscritto completamente il gioco e quindi la versione per il piccolo 8bit di Sega è un altro gioco. Incredibile!

Questa recensione è stata fatta con test su real hardware, ma vi consiglio di provarlo su tutte le forme possibili. Emulazione, Collection su console (Ps2, Xbox, Wii) e persino su Gameboy advance.

Alla domanda iniziale come posso rispondere??? Sì.. è ancora divertente correre nei panni di Sonic, anche dopo 29 anni.

di **Carlo N. Del Mar Pirazzini**

GIUDIZIO FINALE

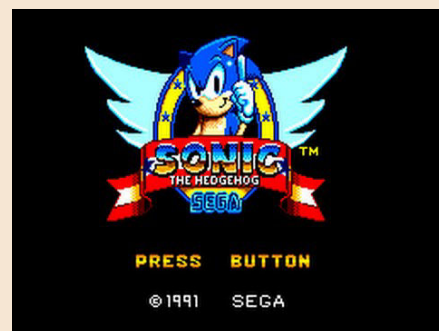


» Giocabilità 95%

Veloce, velocissimo, ipercinetico. Tutto si fa usando la croce direzionale e un pulsante. Semplice e intuitivo.

» Longevità 99%

Ogni livello presenta passaggi segreti, oggetti nascosti, piattaforme quasi irraggiungibili. Esplorarlo tutto anche dopo 29 anni è davvero difficile. Provateci.





ASTAL

Editore: Sega
Sviluppatore: Sega
Piattaforma: Sega Saturn
Genere: Platform

"Astal, il miglior platform... che nessuno ha mai giocato!"

Già perché quando è stato sviluppato, con il suo magnifico 2D ai livelli massimi, con i suoi sprite coloratissimi, complessi e numerosi su schermo il mondo delle console era nel pieno della guerra 32bit che si combatteva a suon di giochi 3D. Tutti volevamo giocare con Super Mario 64, Tomb Raider o Resident Evil! Il 3d aveva preso d'assalto l'industria dei videogiochi.

Sega rimase all'angolo col suo Saturn cercando di recuperare con giochi in 3d non tutti perfettamente riusciti, ma realizzando piccoli gioielli grafici come Astal, snobbati da giocatori e critica.

Astal è un gioco di piattaforma 2D piuttosto semplice, non c'è dubbio. Ma ciò che spicca è la sua grafica straordinaria, la sua bellissima colonna sonora, un gameplay divertente e una storia semi-interessante.

Il salvataggio del mondo da parte di Astal, eroe inizialmente burbero e irascibile che dovrà salvare il suo mondo e la sua bella dai piani di conquista di un demone e del suo servitore.

Una storia semplice, vista e rivista (e per fortuna! ndr) nei videogiochi, ed è proprio così che inizia questo Astal, uno dei primi giochi che ha davvero mostrato le capacità nell'uso della grafica 2d su Saturn.

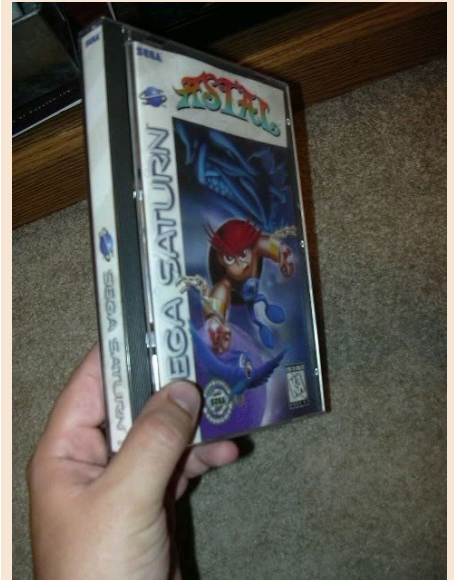
Il protagonista si muove su 16 livelli, interagendo con l'ambiente circostante nel più classico modo dei platform. Può colpire i nemici afferrandoli e lanciandoli oppure colpendoli con una devastante "capocciata" o con il colpo a doppio pugno (stile Bud

Spencer). Può anche aspirare e soffiare via i nemici con una forte raffica di vento.

Nel gioco saremo aiutati da una specie di corvo che ci aiuterà in battaglia con alcuni attacchi speciali utilissimi in alcune situazioni.

La grafica da sola vale l'esperienza di gioco. Il mondo di gioco con i suoi livelli è magnificamente realizzato stupendo.

Sfondi dettagliatissimi, animazioni fluide, sprite dei personaggi disegnati meravigliosamente con Boss di fine livello animati davvero bene.



La colonna sonora è bella come la grafica. Ogni livello ha il suo tema unico, dalla melodia sinfonica dal suono epico del livello 1, alla melodia rilassante e new age nei livelli della foresta. La colonna sonora del gioco utilizza strumenti reali, dandogli una sensazione sinfonica.

C'è anche una canzone pop giapponese nascosta che è stata tolta dal gioco





GIUDIZIO FINALE



» Giocabilità 90%

E' ben strutturato, il personaggio ben caratterizzato e si guida con estrema semplicità. I comandi sono facili e ci si prende la mano subito. La difficoltà non è mai troppo elevata...

» Longevità 65%

...e per questo si termina troppo facilmente. I sedici livelli sono belli ma troppo semplici e strutturati in una unica via. Questo difetto compromette il livello di sfida, soprattutto per chi ha dimestichezza con questo genere di giochi.



nella versione per gli Stati Uniti a cui puoi accedere se metti il CD di gioco nel tuo stereo.

L'unico difetto che posso trovare con la musica è che a volte sembra prevalere sul dialogo espresso durante le sequenze di intermezzo e nel gioco stesso, quindi a volte può essere difficile sentire cosa dicono i personaggi, ma forse è buono in un certo senso.

Il doppiaggio del gioco lascia sicuramente spazio a miglioramenti. Il narratore che muove la storia ha un tono morbido e rassicurante appropriato per la sua voce, ma spesso manca di emozione e sembra leggere i dialoghi dal giornale. Inoltre non mi piace la voce di Astal. Sembra una donna adulta che si sforza troppo di imitare la voce di un ragazzino burbero. Il demone, Jerado, possiede un tono imponente e propizio alla sua voce, ed è probabilmente il migliore degli attori.

Arriviamo alle parti dolenti. C'è un piccolo difetto di gameplay. Il gioco è terribilmente lineare. Il livello si attraversa solo in modo, non ci sono percorsi nascosti, livelli segreti o stanze bonus.

Nonostante la splendida grafica e gli alti valori di produzione, Astal vacilla leggermente a causa di uno dei principali fattori: la relativa facilità di gioco. Anche se il gioco vanta 16 livelli, questi sono abbastanza brevi e alcuni di loro sono



dedicati solo alle battaglie contro i boss. La difficoltà non è mai esagerata e mai punitiva.

Tirando le somme è un gioco divertente e bellissimo da vedere. Un gioiello nascosto del Sega Saturn che proprio quest'anno compie 25 dalla sua uscita in occidente. Un gioco che onora i bei platform in 2d ma che forse risulterà ai più "sgamati" troppo facile e che purtroppo non ha avuto una serie di seguiti e degli sviluppi futuri.

Se valutassi la sola grafica sarebbe un assoluto capolavoro, ma non solo di bellezza si può vivere...

di Carlo N. Del Mar Pirazzini





POWER SPIKES

Sviluppatore: V-System/Data East
Anno: 1989
Piattaforma: Arcade
Genere: Simulazione sportiva

Tra la fine degli anni '80 e la prima metà degli anni '90, la software house giapponese Video System realizzò una trilogia dedicata alla pallavolo. E se Super Volleyball nel 1989 ha riscosso un gran successo, Power Spikes – lo diciamo fin da ora – può essere considerato a buon diritto il suo validissimo erede naturale.

Dal suo illustre predecessore, infatti, Power Spikes prende a piene mani l'ossatura: l'insolita – per il genere di sport – visuale laterale a scorrimento e, sostanzialmente, le stesse dinamiche. Ma aggiunge un gameplay decisamente più raffinato grazie ad una fisica molto più “possente”.

Questo sequel è decisamente interessante in quanto, oggettivamente, migliora quel Super Volleyball che fece tanto scalpore e che è rimasto sempre nel cuore di moltissimi appassionati. Anche se, forse, almeno per chi vi scrive, difetta leggermente nel fascino. Questa però non è una colpa: del resto la prima volta non si scorda mai e Super Volleyball ebbe il merito di far conoscere a molti videogiocatori questo splendido sport che in Italia ha sempre avuto un grandissimo seguito ma che proprio negli anni '90 visse il maggior splendore visti – tra i numerosi trofei internazionali conquistati dalla nazionale maschile – i tre mondiali consecutivi portati a casa nelle edizioni del '90, '94 e '98.

Power Spikes offre maggiori contenuti. È possibile scegliere tra le nazionali maschili e femminili. Ed è giusto ricordare come le diverse versioni del titolo, asiatica ed occidentale, permettessero di selezionare team diversi.

Nell'edizione occidentale, in campo maschile, si potevano scegliere le nazionali di Italia, Francia, Germania, Regno Unito, Spagna, Svezia, Norvegia e Brasile. Le squadre femminili (presenti solo nelle varie edizioni distribuite in Asia, ma che stranamente ricordiamo anche di aver giocato in Italia, ndr) sono Giappone, Corea del Sud e Cina.

Come sempre, l'obiettivo è quello di vincere tutte le partite per salire sul gradino più alto del podio indossando la fatidica medaglia d'oro e, come da copione, il titolo ci permette di giocare le ultime fasi del quinto set. Sei i match per la gloria da conquistare sempre in rimonta visto che in finale si partirà dal un notevole distacco di 8-12 per la CPU. Sei come in Super Volleyball ma con diverse squadre da affrontare.

Il punto forte di Power Spikes è, però, sicuramente il suo gameplay che è più raffinato e ricercato ma anche – paradossalmente – snellito nei controlli. Il titolo utilizza un pulsante per eseguire tutti i fondamentali. Muro compreso che in Super Volleyball era demandato ad un secondo tasto.

I colpi da effettuare sostanzialmente sono rimasti quelli del predecessore (del resto sempre di pallavolo stiamo parlando, ndr), ma ottimamente migliorati da una fisica più credibile. Ci spieghiamo: l'impatto del pallone con i giocatori ha molte più sfaccettature e la ricezione, apparentemente identica, può avere risultati diversi come ad esempio generare una traiettoria non idonea all'alzatore che si vede costretto – anche lui – ad utilizzare il bagher, tutto a discapito di chi deve attaccare. Ne nascono azioni anche





rocambolistiche che, però, ne aumentano la varietà del gameplay e le incognite. Nonostante il gioco offra l'indicazione con tanto di freccia sul dove potrebbe arrivare la sfera, infatti, la difficoltà si rivela insidiosa perché l'impatto col pallone, come detto, può essere sempre diverso. Spesso e volentieri anche un leggero movimento del nostro pallavolista può determinare una giocata pulita o sporca e non è detto che quest'ultima, soprattutto in attacco, non sia vincente.

È proprio questo che fa la differenza col passato: la possibilità di effettuare gli stessi colpi in modo assolutamente differente e con risultati diversi. Un ampliamento del ventaglio d'azione non indifferente che ne aumenta anche la soddisfazione in termini di gameplay. Per farvi intuire meglio, questa evoluzione del gameplay è paragonabile al passaggio che vissero gli utenti di Kick Off che giocarono a Kick Off 2 e successivamente a Goal.

In un gioco come Power Spikes, dove la fisica incide parecchio, la tempistica, come dicevamo, determina una gran percentuale di successo di ricezione, alzata e soluzione offensiva. Il titolo di Video System dà molta importanza al fondamentale della difesa permettendo salvataggi anche spettacolari in tuffo sia in avanti per recuperare, ad esempio, delle palle smorzate, che indietro per "arrampicarsi in aria" e prendere pallonetti infidi, ma anche l'attacco ha la sua parte ed anche la buona riuscita dei colpi è dovuta al premere il pulsante quando la palla diventa rossa. Anche in questo caso possiamo attaccare come vogliamo. Anche – se si è abbastanza spregiudicati – schiacciare dalla seconda linea. Spesso e volentieri può anche essere un vero e proprio jolly. Si possono fare muri, pallonetti velenosi, finte, attacchi anticipati e così via. Ma si possono anche fare tanti errori. In Power Spikes, molte volte, vincerà chi sbaglierà di meno dato che l'errore è sempre dietro l'angolo anche in palloni apparentemente innocui. Ed un

alzata scellerata o svogliata possono rovinare ricezioni strappa applausi. Mai dare nulla per scontato.

Ah, dimenticavamo: come da tradizione sono presenti anche i servizi con colpi speciali: Tower, Phantom e Towing Service.

Dal punto di vista tecnico, il gioco presenta sostanzialmente lo stesso impatto dell'illustre predecessore ma con colori meno sgargianti in favore di più animazioni.

Quest'ultima caratteristica è palese in tutto il gioco, a partire dal pubblico che è in movimento durante le azioni. La visuale laterale è fluida mentre i giocatori si muovono con più frames e varietà. Da segnalare come l'azione del gameplay sostanzialmente non abbia sbavature. Non mancano i soliti ritratti dei giocatori del nostro sestetto, sia in fase di presentazione che dopo i punti messi a terra. E, se si sceglie l'Italia, alcuni giocatori in campo fanno riferimento ad atleti reali come ad esempio Andrea Lucchetta e Luca Cantagalli riconoscibili dal taglio di capelli e dai lineamenti.

Il sonoro, seppur molto in tema, ed in "palla", e pur essendo ancora più nitido, non riesce ad essere memorabile come i motivetti di Super Volleyball. Ciò non toglie che sia di buonissima fattura. Ed i commenti al termine di ogni attacco vincente come "Nice Spike", "Nice Block" sono sempre presenti e, marchio di fabbrica della serie.

Insomma, Power Spikes è un signor gioco, probabilmente il migliore della trilogia firmata da Video System pur non essendo "affascinante" come il capostipite. Le migliorie grafiche, sebbene i colori non siano caldi, le animazioni molto belle ed un ampliamento colossale del gameplay che offre molte più incognite e situazioni diverse ad ogni partita, lo portano ad essere ricordato come uno dei migliori arcade dedicati alla pallavolo di sempre.

di **Edoardo Ullo**

GIUDIZIO FINALE



» Giocabilità 93%

Netto miglioramento del gameplay grazie ad una fisica applicata potenziata e credibile. La varietà di soluzioni offensive e di colpi difensivi è stata ampliata. Difficoltà più accentuata ma gameplay più soddisfacente.

» Longevità 90%

Rispetto al predecessore troviamo più squadre da scegliere e, nelle versioni asiatiche, anche le squadre femminili.



Sinergia significa che il comportamento dell'intero sistema non è predicibile dal comportamento delle sue parti -

Buckminster Fuller, inventore, designer, architetto

La definizione generica di "sinergia" è: il rapporto tra elementi o forze operanti al conseguimento di uno stesso fine. Semplice vero? La chiave è tutta qui! Quando elementi diversi collaborano al conseguimento di uno scopo comune, i risultati sono eccezionali.

Ho voluto prendere in prestito la frase di Buckminster Fuller del titolo per parlare nuovamente di un argomento che mi è sempre stato caro: la collaborazione tra noi appassionati per ottenere un risultato sempre migliore.

In tutti i campi del lavoro dell'uomo ne abbiamo avuto la dimostrazione, ma per ricondurre il discorso al nostro mondo, basta pensare all'opensource, ai progetti condivisi su GitHub o, per rammentare un paio di storie conosciute da tutti, a quelle aziende nate da una comunione di intenti di uno sparuto gruppo di persone che condividevano lo stesso sogno!

Noi di RetroMagazine World abbiamo sempre fatto della collaborazione la nostra bandiera. Nessuno di noi, per quanto preparato o dotato tecnicamente, riuscirebbe a portare avanti una rivista così variegata e tutti i progetti che ruotano intorno al mondo di RMW, senza venirne sopraffatto. E' per questo che siamo in tanti ed abbiamo sempre invitato tutti voi a partecipare ed a collaborare.

Un esempio che mi preme citarvi è l'articolo che questo mese ci ha inviato **Antonino Porcino**; dopo averlo letto ho sentito il dovere di dare il mio contributo e, non appena ne abbiamo parlato in redazione, anche **Ermanno** si è messo all'opera per contribuire. Un altro esempio che, grazie all'amicizia che mi lega con **Marco**, posso seguire da vicino, e' la programmazione del tool PlayEdicola. Nato come un progetto personale, grazie al contributo di altri volontari è cresciuto sensibilmente e presto potrà essere a disposizione di tutti. E come non menzionare il gioco che **Leonardo** sta sviluppando insieme ad un coder olandese...

La barriera dell'individualismo è stata di nuovo infranta... Per arrivare là dove nessun uomo è mai giunto prima! (Ah no, questa è un'altra storia...).

Voglio chiudere anticipandovi che nei prossimi mesi lanceremo una serie di progetti che necessiteranno la Vostra collaborazione. Non vi rivelo altro, perchè è effettivamente ancora presto e di carne al fuoco ce n'è già tanta, come avrete avuto modo di leggere nell'editoriale di **David**. Non vi resta altro da fare che continuare a seguirci e, se lo vorrete, quando sarà il momento, collaborare con noi.

Francesco Fiorentini

Disclaimer

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

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

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

Attribuzione

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

NonCommerciale

Non puoi utilizzare il materiale per scopi commerciali.

StessaLicenza

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

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

Divieto di restrizioni aggiuntive

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



RetroMagazine World
Anno 4 - Numero 23 - MAGGIO 2020

Direttore Responsabile

Francesco Fiorentini

Vice Direttore

Marco Pistorio

Coordinatore Redattori

David La Monaca

Responsabile Area Web

Giorgio Balestrieri

