

EDITORIALE: Buon Natale a tutti!

- Prova su strada dell'interfaccia Pi1541 Disk Drive Emulator
- MSX - The Dark Side of 8bit
- I purosangue Sinclair della serie ZX
- **ChimicaWare**: un software didattico per ZX Spectrum
- Ghost'n COBOL - parte 2 - I file sequenziali
- Un fantasmagORICo home computer - parte 2
- RetroPie Utilities
- **BENVENUTI IN UN MONDO FANTASTICO**: Introduzione al gioco di ruolo
- **RetroMath**: La matematica degli stormi di uccelli
- B.B.S. su C64? Si puo' fare!
- Esplorando l'Amiga - parte 3

GIOCHI

- Nova9: il ritorno di Gir Draxon
- Journey to the centre of the Earth
- Civilization
- Black Tiger
- Space Manbow

RetroGiochiAmo: The Official Father Christmas

RetroGiochiAmo: Gabriel Knight - Sins of the Fathers

Intervista a **Fabrizio Radica**: Cinema, VR, Amstrad, Sprint2076 - *terza puntata*

Resoconto evento: **Firenze Vintage Bit 2018**

Alla scoperta del "Museo dell'Informatica Funzionante"

MANUALE - INTRODUZIONE ALL'ARCHITETTURA DEL C64

RetroEvoluzione elettronica

DICEMBRE 2018 - WWW.RETROMAGAZINE.NET

RetroMagazine

Anno 2 - Numero 11

Hanno collaborato a questo numero:

- Giuseppe Friscaro
- Ermanno Betori
- Alberto Apostolo
- Attilio Capuozzo
- Daniele Brahimi
- Marco Pistorio
- Starfox Mulder
- Leonardo Giordani
- Michele Ugolini
- Giuseppe Fedele
- Dante Profeta
- David La Monaca (Cercamon)
- Giorgio Balestrieri
- Querino Ialongo
- Adriano Avecone
- Francesco Fiorentini

Immagine di copertina:
Flavio Soldani

IN EVIDENZA IN QUESTO NUMERO

Buon Natale a tutti!

di Francesco Fiorentini

Sembra che lo facciamo apposta, ogni uscita di RetroMagazine e' accomunata da una ricorrenza piu' o meno rilevante dell'anno, quasi a voler scandire il trascorrere inesorabile del tempo e della nostra esistenza.

Probabilmente non e' proprio un caso, tutti i redattori scrivono per diletto e, non essendo un lavoro vero e proprio anche se vi assicuro che l'impegno e' lo stesso, le festività ci aiutano, a trovare quel poco di tempo in piu' da dedicare a questa nostra passione.

Nel momento in cui scrivo l'editoriale, Halloween e' passato da poche settimane ed il Natale si profila all'orizzonte, ma ancora sufficientemente lontano per non pervadere l'editoriale con solo i canonici auguri di rito. ☺

Il Natale pero' e' anche quel periodo che ci avvicina alla fine dell'anno, generalmente tempo di bilanci e di promesse per l'anno nuovo. L'anno scorso ho potuto saltare questa 'tradizione', avendo RetroMagazine soltanto 2 numeri sul groppone (perdonate la mia toscanita'), ma quest'anno devo per forza di cose farlo per rendere giustizia al fantastico lavoro svolto da tutto il team redazionale.

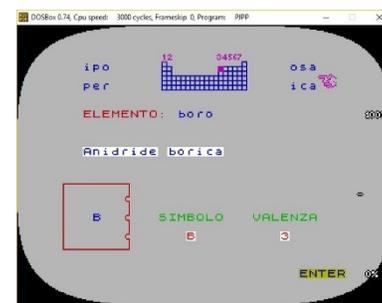
Quello che tenete tra le mani e' il numero 11, l'ultimo numero del 2018; un 2018 che ha visto ben 9 edizioni prendere corpo e trasformarsi da semplici idee in qualcosa di tangibile che ha fortunatamente riscosso il Vostro consenso. Centinaia di pagine sono state scritte, migliaia di parole sono state dette, ma sembra che la

vena artistica della nostra redazione sia inesauribile e sia sempre pronta a partorire qualcosa di sorprendente per il numero successivo!

Durante quest'anno avrete avuto modo di notare come la rivista si sia evoluta, cercando di trattare sempre piu' sistemi possibile ed aggiungendo ad ogni numero una piccola novita'; il sito web, il software allegato, le nuove rubriche e per ultima, ma solo in ordine temporale, la copertina.

Come per tutti progetti, la fase di crescita e' forse quella piu' semplice, il difficile e' riuscire a dare continuita' a quanto di buono e' stato fatto sinora. Adesso abbiamo di fronte una sfida per cercare di mantenere lo stesso livello qualitativo che ci ha contraddistinto sin qui. Personalmente non ho nessun dubbio che riusciremo nell'intento perche' le idee all'interno della redazione non mancano e per fortuna nemmeno gli argomenti da trattare. Inoltre le collaborazioni che abbiamo stretto con realta' affermate come **OGI** e nuove realta' come **8 Bit RetroProgramming Italia** non potranno che portare beneficio a noi, ma soprattutto ai nostri lettori, che potranno contare su un prodotto sempre piu' eterogeneo e professionale.

Adesso pero', come e' giusto che sia, e' venuto il momento degli auguri. Quindi **Buon Natale** e **Felice Anno Nuovo** a tutti i nostri lettori da parte della redazione di RetroMagazine.



ChimicaWare

Alberto Apostolo e' riuscito nel suo intento di intervistare il prof. **Cardellini**, autore di **ChimicaWare** per ZX Spectrum, ed ottenere anche la distribuzione di questo software introvabile in esclusiva per tutti i lettori di RetroMagazine.

Articolo a pagina 12



B.B.S. su C64? Si puo' fare!

Marco Pistorio non ha esitato a mettere in piedi una patch (non ufficiale) di **VICE** per far provare ai Suoi lettori "l'ebrezza" di usare un vecchio software per B.B.S. "old-style" sfruttando nel contempo le tecnologie attuali!

Articolo a pagina 34

Prova su strada dell'interfaccia Pi1541 Disk Drive Emulator

di Giuseppe Friscaro



Si tratta di un accessorio per il C64-C128 e anche Vic20-C16/Plus4, in grado di emulare un Drive 1541 con 100% di compatibilità rispetto agli altri dispositivi economici disponibili attualmente, che possono avere qualche limitazione sotto questo aspetto.

Il Pi1541, come si evince dal nome, è basato su Raspberry Pi 3 e consiste in un modulo aggiuntivo da inserire nell'apposito connettore di espansione presente sul Raspy.

Il presupposto per potere essere utilizzato è proprio quello di possedere un Raspberry Pi3, che inizialmente potrebbe sembrare un limite, ma in realtà questo SBC (Single Board Computer) si sta facendo strada nel mondo informatico e videoludico, grazie alle sue molteplici possibilità di impiego. Tuttavia la mia prova si limiterà al solo Pi1541.

Questo viene collegato alla porta IEC e alimentato ovviamente dal Raspberry. La versione in Test è quella equipaggiata con display OLED e firmware 1.12.

Da quello che ho potuto provare, devo dire che è un bell'accessorio, specie per chi vuole vedere su display quello che sta per essere letto dal C64 (come nel mio caso) e preparare eventualmente una lista di file .d64 da caricare, come una playlist.

Questa eventualità torna utile, nonché indispensabile, nel caso in cui si utilizza un gioco multi-disco: basta caricare i dischi in sequenza e successivamente selezionarli quando richiesti dal gioco. Forse un po' troppo

laboriosa e poco immediata (volendo prendere come riferimento una normale SD2iec, in cui bastava caricare direttamente il file browser presente nella Root e avviare quindi la selezione dei file) in quanto qui il file Browser va selezionato prima da una posizione ben precisa sulla SD, altrimenti il contenuto non risulta visibile dal C64.

Tuttavia, la comodità sta nel potere fare queste selezioni completamente dal display Oled, senza l'ausilio del computer. Questo avviene mediante i 5 pulsanti presenti sull'interfaccia che permettono la navigazione tra file e cartelle, nonché la selezione e la memorizzazione delle liste. Ogni file selezionato dal menù va caricato e da quel momento risulterà "inserito" come Disco nel Drive e pertanto visibile sul C64, un po' come accade sugli emulatori.

Molte opzioni si possono impostare dal file di configurazione (tramite normale Text Editor) presente nella Root della microSD. Una di queste è proprio il caricamento all'avvio del "disco" con il fileBrowser, in modo da averlo sempre pronto al boot del sistema.



Come accennato all'inizio, la Pi1541 emula il Drive 1541 al 100% in quanto rispetta i timing della CPU, garantendo così la totale compatibilità con i FastLoader presenti in molti giochi.

A prova della sua "fedeltà" nell'emulare il Drive, ho voluto appositamente eseguire un programma per il controllo dell'allineamento testina sui 1541 e devo dire che il C64 lo aveva considerato un vero Drive! Ovviamente le tracce risultavano impeccabili!

Anche la compatibilità con un eventuale JiffyDos è garantita dal fatto che incorpora già il codice per accelerare il protocollo IEC, per

cui anche in assenza di tale velocizzatore nel computer non fa alcuna differenza.

In sostanza, tralasciando il fatto che dobbiamo essere possessori di Raspberry 3b o in procinto di acquistarne uno anche per altri scopi, lo trovo una buona alternativa per chi non dovesse possedere una SD2iec o accessori analoghi, ma non lo ritengo un suo sostituto, anche per il fatto che si tratta di un progetto piuttosto recente, per cui in continua evoluzione ed aggiornamento.

Questo fa del Pi1541 un accessorio interessante ma non proprio indispensabile, almeno fino a nuove release, magari un po' più "stand-alone", considerando il prezzo totale per tale attrezzatura (55€ circa per un kit Raspberry 3b completo + 30-40€ per la Pi1541), indirizzato forse più ad un pubblico smanettone.

Link sull'argomento:

<https://cbm-pi1541.firebaseio.com/>

Gruppi Facebook correlati

Retrocomputer World:

<https://www.facebook.com/groups/239822176722864/>

Pianeta Commodore:

<https://www.facebook.com/groups/1548452001848020/>

Blog personale:

<https://www.facebook.com/GiuMacGyver/>



MSX - The Dark Side of 8bit

di Ermanno Betori

Nei numeri antecedenti abbiamo trattato il computer TI99/4A creato dalla ditta Texas Instruments (aka T.I.) e raccontato la sua storia, l'hardware, il software ma abbiamo omesso di dire che la T.I. fu anche una ditta che vendeva componentistica militare e civile, pertanto i chip che produsse e con i quali assemblò il TI99/4A, furono venduti a qualunque azienda che ne facesse richiesta, localizzata in qualsiasi parte del mondo. Questo permise a varie ditte di computer di avere a disposizione già pronti i chip video, sonori, di logica ad esclusione delle Grom che contenevano il linguaggio GPL in quanto erano un componente specifico ed essenziale per i computer Mini/Home creati dalla T.I.

Ciò permise di avere sul mercato una serie di computer molto simili nelle caratteristiche tecniche e nelle capacità grafiche/sonore che di fatto crearono una famiglia di macchine le quali ebbero successo in varie parti del mondo ma in Italia furono misconosciute ad esclusione di qualche eccezione.

Tali macchine furono costruite di massima usando come base tre chip fondamentali. - Lo Z80 come CPU a 8 bit in quanto più economica rispetto al TMS9900 (16bit)

- Il processore Video: TMS9918A/9928A-29A

- Il processore audio: Texas Instruments SN76489A e in altri modelli il General Instrument AY-3-8910.

Tra i componenti di questa "famiglia" troviamo vari sistemi quali console o ibridi console/home computer come il ColecoVision, il CreatiVision, il Sega SG-1000 e computer veri e propri come il TI-99/4, il Sega SC-3000, il Sord M5, il Tatung Einstein, il Memotech MTX, i computer SVI318/328 della Spectravideo, ed infine i computer MSX1 ed i loro cloni. Abbiamo già trattato il TI99/4 che in ordine temporale lo possiamo definire il nonno di questa famiglia perciò ci dedicheremo al nipotino il computer MSX che è tra tutti quello più conosciuto. I genitori, zii, cugini vari saranno argomento di future letture.

Nipoti di un dio minore .. i computer dello standard (mancato) MSX.

Il computer MSX chi era costui? Di Carneadine memoria in buona parte dell'Europa, ma specialmente in Italia ne sappiamo ancora

oggi veramente poco.. ad esempio il sottoscritto, conoscitore di sistemi 16 Bit come Amiga - Atari ST - TI99, oppure di 8 bit Commodore - Spectrum - Amstrad CPC, era fino al 2008 un completo ignavo dei computer aderenti allo standard MSX.

Si' uno standard! Io erroneamente credevo che il computer MSX fosse di una sola casa costruttrice e gli altri modelli dei cloni, vedi la storia del PC IBM, invece come scoprii, agli inizi degli anni 80 un consorzio di ditte di varia nazionalità si accordarono per creare dei computer a 8 bit che dovevano aderire allo standard MSX (acronimo di Computer con Software Interscambiabile).

Tra loro vi era la europea Philips/Phonola, la statunitense Spectravideo, la brasiliana Gradiente/National, la Argentina Talent ed una moltitudine di ditte dell'estremo oriente giapponesi, coreane, taiwanesi, tipo Sony, Panasonic, Canon, Goldstar, Toshiba, Casio, Pioneer, Sanyo, JVC, Daewoo, Mitsubishi, Hitachi, Sharp, Yamaha, Fujitsu, Sharp, Samsung. Alcune aziende come la Sony o Philips lanciarono addirittura circa dieci modelli.



Cosa accumulava tali macchine? Dovevano avere tutte uguali quasi la stessa componentistica che gli permettesse di far funzionare lo stesso software.

Nel leggere tali informazioni una domanda che mi sorse spontanea... perché molte case costruttrici aderirono a tale standard.. dove era il loro guadagno dato che le macchine erano tutte uguali? La risposta era nel fatto che lo standard dava dei paletti non definiti, cioè i computer dovevano avere tale componentistica X e poter far girare il software Y, ma nulla toglieva alla ditta di customizzare la macchina aggiungendo ad esempio una periferica Z. A tal proposito scoprii che la ditta Yamaha trasformò il suo computer MSX il CX5M in una stazione musicale che era interfacciabile con la tastiera musicale (sempre Yamaha) DX7 molto in voga negli anni 80.



Hic sunt leones.. mi si era aperto un mondo sconosciuto e misterioso...

Fortunosamente, tramite un mercatino sotto casa nel 2010 venni in possesso di un MSX il Philips VG-8020. Questo modello è quello che ebbe una buona vendita in Italia (infatti si trova a prezzi accessibili) ma come quantità fu inferiore rispetto al computer Commodore 64 e ZX Spectrum.

Spinto dalla necessità di usarlo ed anche dalla curiosità di avere a che fare con un computer

all'epoca semiconosciuto, cominciai a documentarmi tramite la ricerca di libri, riviste ecc...



Primo problema... I modelli

Sembrava una cosa facile da comprendere ma di computer MSX ne furono costruiti a iosa e li possiamo dividere in quattro grandi gamme con caratteristiche diverse:

- il MSX1 come abbiamo già accennato aveva come CPU uno Zilog Z80 a 3,5 Mhz con memoria Ram che variava da 16K a 64K; Un processore video TMS9918A ed uno sonoro il AY-3-8910.

- il MSX2 che era un MSX1 potenziato nella grafica in quanto usava il nuovo chip grafico Yamaha V9938 (tale chip era stato sviluppato dalla T.I. e poi venduto alla Yamaha) retro compatibile al 100% con il TMS9918A, aveva una memoria RAM che variava di serie da 64K a 256K, ma poteva gestirne fino ad 1Mbyte. In più il 95% dei modelli MSX2 avevano di serie il floppy drive.

- Il MSX2+ invece fu uno standard seguito solamente da tre case costruttrici giapponesi, Panasonic, Sanyo e Sony. Aveva un chip grafico ancora più evoluto il Yamaha V9958 che era il successore del V9938, una maggiore possibilità di gestione della ram fino a 4 Mbyte, e avere di serie due chip sonori, uno era il PSG e l'altro il YM2413 che aderiva allo standard MSX-Music.

- Il MSX Turbo R fu prodotto solo dalla ditta Panasonic e rappresentò il canto del cigno dei sistemi MSX, venne commercializzato solo in Giappone come i modelli MSX2+ ma a differenza aveva delle interessanti novità tra le quali ricordiamo:

- Il doppio processore: R800 a 16 bit, architettura RISC e lo Z80A con frequenza di 3,576 MHz (per compatibilità verso il basso)
- Tre chip sonori: il General Instruments AY-3-8910 ("PSG"), il Yamaha YM2413 (OPLL) ("MSX Music") ed il PCM.

- Un Microfono incorporato, e delle porte MIDI in/out (solo nel modello FS-A1GT)

Secondo problema... i libri

In quel periodo non riuscivo a trovare documentazione scritta in italiano, ne in Internet come PDF ne su siti di e-commerce. Oggi tale problema è in parte risolto grazie al sito <http://www.microatena.it> che mette a disposizione di chiunque i quattro libri indispensabili al neofita sull'uso dei sistemi MSX.

Perché in parte? Il motivo consiste nel fatto che per questi computer esistono un mare di libri pure molto tecnici e didatticamente evoluti ma ahimè scritti prevalentemente in Giapponese, Brasiliano, Spagnolo, Russo, Olandese. In lingua inglese sono molto pochi ed in quella italiana non superano la decina di unità. In pratica i libri furono prodotti dove vi fu una forte vendita.

Terzo problema... le riviste

Stando alle informazioni ricavate in internet nel 2010 risultava che i vari modelli di computer MSX venduti in Italia fossero arrivati come volume di vendita quasi al terzo posto come unità presenti pertanto logica presupponeva che vi fossero un elevato numero di riviste dedicate a queste macchine. Ma nulla si trovava... nessun riferimento ad eventuali club MSX italiani ancora attivi e l'unico gruppo italiano che trovai presente era su Yahoo ma con post veramente rari, al contrario dei gruppi esteri attivissimi, come ad esempio spagnoli o olandesi.

Deciso ad approfondire questa inspiegabile mancanza di riviste cercai durante i vari raduni di appassionati di retro computer tipo Brusaporto, Firenze Vintage Bit, ecc. appassionati del mondo MSX che mi sapessero delucidare in merito, ma trovai esperti utenti dei computer, tecnicamente preparati ma nessun gruppo! Non esisteva un vero club attivo di appassionati. Da tutti loro in questi ultimi otto anni sono riuscito a scoprire che oltre a alle testate generaliste multi-computer tipo M&P Computer o MC Computer ecc... dove vi erano spazi dedicati al mondo MSX, esistevano altre 16 riviste italiane dedicate quasi esclusivamente ai sistemi MSX ma queste non erano mai state molto pubblicizzate, in quanto tali riviste si vendevano solo con le unità a cassetta contenenti i programmi (la maggior parte giochi) e la stragrande maggioranza degli utenti dell'epoca non ne diedero la giusta importanza derubricandole a corredo della cassetta.

Oggi grazie al lavoro di scansione delle riviste e dump delle cassette effettuato dal Sig. **Giuseppe di Lillo**, in arte **Sovox**, <http://specialprogramsipe.altervista.org/riviste.php> finalmente tali riviste sono a disposizione di tutti.

Quarto problema... le periferiche

Già le periferiche.. un vero mondo a parte! Ricordate le customizzazioni che lo standard permetteva? Ebbene non furono poche, anzi, una vera pletora! Per avere una visione di insieme di quante furono e del loro uso è stato compiuto un grande sforzo in quanto vi sono un serie di programmi dedicati al loro uso che non sono pochi.

Cominciamo dal comparto audio. Ad esempio se il Commodore 64 ha come chip audio il SID e basta, perciò qualsiasi programma usava quel chip chi usava i sistemi MSX aveva la possibilità di usare ben cinque chip audio per creare i suoi programmi.

Cominciamo dal **PSG (Programmable Sound Generator)** che è il chip inserito di default in tutti i computer MSX. Il chip originale è il modello **General Instrument AY-3-8910** ma vi furono una serie di cloni usati dalle svariate ditte, ecco in elenco una piccola lista:

- File KC89C72 (usato nei computer MSX Arabi)
- Microchip Technology AY-3-8910A (non è veramente un clone in quanto la Microchip Technology è una vecchia divisione della General Instrument)
- OY-2-8910AC (usato nei computer MSX Arabi)
- Toshiba T7766A (clone ufficiale)
- WB5300 (raro)
- Winbond WF19054 (raramente usato nei sistemi MSX)
- Yamaha YM2149 (clone ufficiale spesso usato nei sistemi MSX)

I produttori degli MSX capirono che i computer aderenti allo standard avevano potenzialità nei paesi dove non vi era stato ancora un boom dell'informatica ma dovendosi scontrare nel mercato Americano ed Europeo già saturo, queste macchine venivano soffocate negli U.S.A. dalle aziende presenti sul posto come Commodore - Atari e in parte dal residuo di user del TI99/4A, mentre in Europa e specialmente in Italia dal duopolio Commodore-Spectrum. Per ovviare

a ciò pensarono di migliorare il comparto musicale creando di fatto due standard musicali il MSX-Audio e il MSX-Music.

Il MSX-AUDIO era uno standard per FM sound synthesizer cartridges, che contenevano un chip OPL-compatibile costruito dalla Yamaha il Y8950, la possibilità di avere porte MIDI In/Out e il connettore per inserire una tastiera musicale. Questo chip fu introdotto nel 1984 ma le cartucce che lo usavano cominciarono ad apparire nel 1986.

Queste furono le cartucce che adottarono lo standard MSX-Audio:

IL Philips NMS 1205 (Music Module) con porte midi e annessa tastiera musicale.



Il Panasonic FS-CA1 (MSX-AUDIO cartridge)



Ed infine il Toshiba HX-MU900 (FM-synthesizer unit)



Ora ci dedicheremo al chip audio Yamaha YM2413. Questo chip fu scelto per costruire le cartucce da usare che aderivano allo standard MSX Music. La più famosa FM sound synthesizer cartridge che fu venduta ed aderiva a tale standard era la cartuccia FM-PAC creata dalla Panasonic. Il nome completo

era FM Pana Amusement Cartridge. E' la sola cartuccia che aderì ufficialmente allo standard MSX-Music le altre aziende che crearono delle cartucce MSX-Music furono la Zemina con il Music Box e la Checkmark con il FM-Stereo-Pak. Queste cartucce furono create e molto usate nei giochi per aumentarne gli effetti sonori da renderli a livello musicale competitivi con i computer a 16 bit Amiga / Atari ST ma essendo stato integrato come nuovo standard solo nei computer MSX2+ e MSX TurboR non ebbe una grande diffusione se non in Giappone.



SCC - Sound Creative Chip

SCC è l'acronimo di Sound Creative Chip, venne menzionato nella Konami Software Club newsletter del 3 Novembre 1987. Questo chip sonoro venne sviluppato dalla Konami, in collaborazione con la Toshiba dal febbraio del 1986, ed era stato progettato per essere inserito nei nuovi giochi in formato cartuccia o per essere inserito in una specifica cartuccia necessaria a dei game creati appositamente su floppy (in pratica giochi prettamente MSX2 potenziati da questo chip sonoro dedicato). In più questo chip fu usato in vari Konami arcade systems (cabinati).



Sopra il PCB e visione completa della Konami Sound Cartridge, e sotto il pcb di un gioco konami contenente il chip sonoro aggiuntivo SCC.



MoonSound e suoi cloni

La MoonSound è una scheda audio FM a 16 bit creata esclusivamente su cartuccia, venne rilasciata nel 1995 alla fiera dei computer di Tilburg (Olanda). Tale cartuccia venne concepita dall'ingegnere elettronico Henrik Gilvad e venne prodotta dalla ditta svizzera Sunrise. Basata sullo Yamaha YMF278 OPL4 sound chip, è capace di 18 canali FM sound e 24 canali a 12 / 16 bit sample. Questo progetto arrivò a disposizione della comunità MSX dopo che la divisione US di Microsoft abbandonò i sistemi MSX focalizzandosi sul IBM PC, ma passarono molti anni prima che Henrik riuscisse a concretizzarlo.

Praticamente questa scheda dovrebbe essere compatibile con le schede audio aderenti allo standard MSX-Audio e MSX-Music ma i programmi dovrebbero essere ricompilati per far riconoscere la scheda audio, inoltre permettere di suonare perfettamente senza gravare il lavoro sul processore i file .mod (amiga), Midi, wave.



Riassumendo in questa breve carrellata abbiamo visto ben sei tipi di schede sonore che si interfacciano con i computer MSX.

Il seguito e la conclusione al prossimo mese!

I purosangue Sinclair della serie ZX

di Alberto Apostolo

Prosegue con questo articolo, il percorso nella storia delle macchine di Clive Sinclair iniziato con il Science Of Cambridge MK 14 (RM n.6) e poi trattando alcuni modelli di calcolatrici prodotte negli anni '70 del XX secolo (RM n.9).

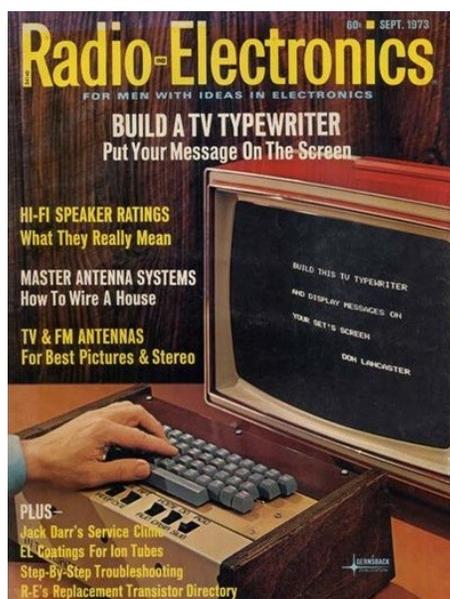
Parlare dei computer della serie ZX è arduo a causa della vastità della letteratura sull'argomento e per il fortissimo rischio di finire a scrivere l'ennesima inutile (pallosa) recensione, elencando dati tecnici ormai noti ai più.

Spero di riuscire lo stesso a dare un tocco di originalità e, magari, far esclamare a qualcuno "Ah, però! Non lo sapevo!", perché non si può mancare di dare un accenno riguardo a queste macchine da me considerate alla stregua di purosangue della scuderia di Clive Sinclair (insignito del titolo di Sir per i successi ottenuti con le vendite degli ZX).

Il Sinclair ZX 80

Nel 1973 apparve il TV Typewriter (non è un dispositivo Sinclair, n.d.A.). Progettato da Don Lancaster, i fanatici dell'elettronica potevano costruirsi una macchina da scrivere in grado di mostrare caratteri alfa-numeric su un normale televisore da salotto [Rya11].

I componenti elettronici necessari per realizzarlo costavano 120 Sterline dell'epoca, come riportato dalla rivista "Radio Electronics" nel numero di Settembre del 1973 (fig.1).



Il progetto originale includeva 2 schede di memoria e poteva generare e memorizzare 512 caratteri suddivisi in 16 righe di 32 caratteri ciascuna.

Una musicassetta di 90 minuti poteva servire da memoria di massa per circa 100 pagine di testo (bastava connettere un registratore qualsiasi al Typewriter).

Questo progetto usava un hardware ridotto all'osso per generare la temporizzazione dei diversi segnali necessari per produrre un segnale TV.

Clive Sinclair utilizzò lo stesso approccio per costruire il leggendario Sinclair ZX 80 (fig.2).



In commercio dal Febbraio 1980, il nome doveva suggerire che il micro-processore utilizzato fosse lo Z80 (la "X" era una trovata per creare una sorta di aura "magica" [Sta15]). In realtà i primi esemplari montavano un "clone" NEC μ PD780C-1, i successivi un vero Zilog Z80.

Il progetto era stato affidato a Jim Westwood e fu completato all'interno del periodo fissato di 6 mesi [LA12].

Una curiosità: lo ZX80 fu progettato quando ancora esisteva la Science Of Cambridge, rinominata Sinclair Computer nel Novembre 1979 che poi divenne Sinclair Computer nel Marzo 1981: (<http://rk.nvg.ntnu.no/sinclair/sinclair/corphist.htm>, consultata il 27 ott. 2018).

Siccome la produzione terminò nell'Agosto 1981, allora esistono ZX80 marcati "Sinclair Computer" e altri marcati "Sinclair Research" (

<http://rk.nvg.ntnu.no/sinclair/computers/zx80/zx80.htm>, consultata il 27 ott. 2018).

L'aspetto esteriore era il frutto del lavoro di Rick Dickinson. Venduto con appena 1KB di RAM, costringeva gli acquirenti a procurarsi l'espansione da 3KB.

Lo ZX80 si collegava a un televisore sintonizzandolo sul canale UHF 36, freq. 591,25 MHz. La generazione del segnale video era in carico al microprocessore

Z80, il quale doveva anche gestire (a turno) il software, la tastiera e il resto dell'hardware [Dil16].

Come effetto collaterale di tutto ciò, risultava uno sfarfallio della immagine ("flickering"). Ogni volta che si premeva un tasto l'immagine sullo schermo scompariva per un istante brevissimo ma percepibile.

I programmi e i dati si potevano registrare sulle musicassette di un normalissimo registratore. Alla fine degli anni '70, usare una musicassetta per salvare dati e programmi era ormai una prassi consolidata presso i costruttori di home-computer di tutto il mondo. E pensare che solo nel 1971 era ritenuta fantascienza oppure un gadget alla James Bond. In "Diamonds are forever" ("007 - Una cascata di diamanti"), una parte consistente del film era incentrata sulla ricerca di una musicassetta nella quale era registrato il programma di controllo di una potente arma satellitare posseduta dal cattivo Blofeld (fig.3).



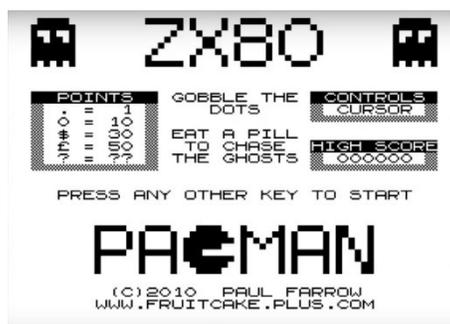
A causa della scomoda tastiera a membrana, per alleviare la difficoltà di digitare i programmi, i comandi in linguaggio Basic si inserivano premendo un singolo tasto. Per esempio, premendo il tasto G, appariva sullo schermo il comando GOTO.

Nel Basic, compresso in 4 KB di ROM, era disponibile solo l'aritmetica intera (le variabili avevano un range tra -32768 e +32767). Non era prevista una libreria di funzioni

matematiche e nemmeno l'aritmetica in virgola mobile.

Merril Rick Chapman nel suo libro "Alla ricerca della stupidità" [Chao4], descrive così lo ZX80: "...gli analisti lo definirono il primo vero computer da armadio: lo acquistavi, ci giocavi un po' e un attimo dopo lo chiudevi in un armadio e andavi a comprare un vero computer".

Di sicuro le limitazioni non hanno scoraggiato Paul Farrow, il quale con molto senno di poi ha realizzato nel 2010 una versione per ZX80 del celeberrimo gioco PAC-MAN (<https://www.youtube.com/watch?v=pdSo3E7FrPg>, fig.4)



Successivamente furono venduti kit di aggiornamento per potenziare lo ZX80 fino a renderlo simile a uno ZX81, suo immediato successore [Bon82].

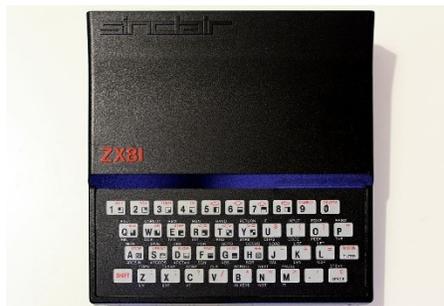
Ma erano comunque presenti delle limitazioni come l'impossibilità di vedere formarsi un disegno e quindi di effettuare animazioni grafiche [Mar82].

Questi modelli aggiornati di ZX80 sono noti con il nome di "ZX80 Nuova ROM" (fig.5) e disponevano di una espansione di memoria da 16 KB.



Il Sinclair ZX 81

Con lo ZX81 (fig.6), Clive Sinclair proseguì il tentativo di produrre un computer nazionale popolare.



Presentato nel Marzo 1981, il design portava ancora la firma di Rick Dickinson. L'hardware (fig.7) si basava sempre sul microprocessore Z80 e rappresentava una ottimizzazione dello ZX80.



Molti componenti discreti presenti sulla scheda dello ZX80 erano stati incorporati in una U.L.A. (Unità Logico Aritmetica), un chip custom realizzato dalla Ferranti. Aveva un solo KB di RAM, rendendo di fatto obbligatorio l'acquisto della espansione da 16 KB.

Il case era per l'epoca moderno e minimalista, le connessioni di input/output erano poste con discrezione da un lato e sul retro. All'inizio i primi prototipi erano fatti di acciaio stampato e alluminio, poi di plastica stampata per non avere problemi con l'alimentazione.

Le piccole dimensioni (in cm: 16.7 x 17.5 x 4, 350 gr di peso), la forma elegante e ben rifinita, il basso prezzo al dettaglio, lo rendevano il "vero" computer da 100 Dollari [LM11].

La tastiera a membrana permetteva di ridurre il numero di componenti meccaniche e abbassare i costi di produzione ma era piccola e poco ergonomica (nel senso classico della dattilografia). Tuttavia, per dare tutte le indicazioni possibili senza creare confusione, era stata fatta molta attenzione nel disporre funzioni e comandi e al contrasto dei colori usati per numeri, lettere e simboli grafici.

Jim Westwood aveva apportato delle migliorie rispetto allo ZX80, riuscendo anche

ad eliminare quel fastidioso sfarfallio dello schermo [Laio4].

In 8 KB di ROM, Steve Vickers della Nine Tiles aveva esteso il Basic [Laio4], implementando l'aritmetica dei numeri in virgola mobile e una libreria di funzioni matematiche non esente da bug (il più famoso è la radice quadrata di 0.25 che restituiva 1.3591409 anziché 0.5!).

Tuttavia alcune istruzioni erano ancora assenti come (ad esempio) le READ, DATA, RESTORE per gestire i dati statici all'interno di un programma. I possessori dello ZX81 potevano contare sulla rapida diffusione di libri e riviste specializzate che gareggiavano nel proporre trucchi di programmazione e routine in linguaggio macchina per superare tali mancanze.

Anche il suono era assente ma sui libri e sulle riviste di Elettronica si proponevano progetti di circuiti aggiuntivi da comprare oppure da costruire.

I programmi e i dati venivano salvati su musicassette collegando opportunamente un registratore qualsiasi (velocità di trasferimento: 250 baud).

Nello stesso periodo, la Sinclair aveva messo in commercio anche la ZX Printer, una piccola stampante su carta chimica (fig.8).



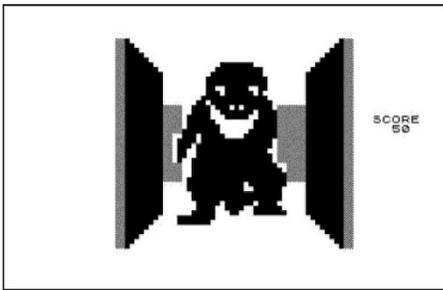
Un monitor dedicato non era necessario perché anche lo ZX81 si poteva collegare a un televisore qualunque dopo averlo sintonizzato sulla frequenza UHF 36.

Se si desiderava collegare un monitor occorreva una manovra da esperto di elettronica: dopo avere aperto lo ZX81 (e rimosso dalla scheda i cavi flat della tastiera senza romperli) si dovevano saldare, a monte del modulatore del segnale TV, i contatti del cavo per il monitor al fine di avere accesso al segnale video composito [Ber13].

Lo ZX81 aveva due istruzioni per la modalità di esecuzione dei programmi Basic: SLOW e FAST. In SLOW, lo ZX81 eseguiva calcoli e li visualizzava in contemporanea mentre in

FAST era più veloce nei calcoli ma la visualizzazione veniva ritardata (a causa della mancata sincronizzazione tra microprocessore e ULA).

La grafica in bianco e nero, basata su caratteri alfanumerici e simboli speciali che permettevano di visualizzare grafici in risoluzione di 64 x 44 punti, non impediva ai programmatori professionisti di realizzare videogiochi come "3D Monster Maze" di Malcom Evans (apparso nel 1982, [Sta15], fig.9), mentre un dilettante esperto poteva realizzare una versione semplificata di "Space Invaders" (in fig.10 è riportato il programma apparso sulla rivista italiana "LIST" n.2, Maggio-Giugno 1984 - "Marzianetti").



Il Sinclair ZX Spectrum

Clive Sinclair ampliò i concetti sviluppati con lo ZX81 e nell'Aprile 1982 presentò lo ZX Spectrum (fig.11).

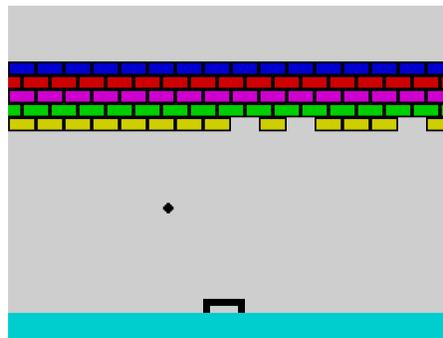


Nelle intenzioni di Sinclair, lo ZX Spectrum doveva distinguersi dal diretto concorrente BBC Micro, costare almeno la metà, essere piccolo ed elegante.

Richard Altwasser fu incaricato del progetto hardware chiamato ZX82 e cominciò a scrivere le specifiche. In linea di massima, bisognava prendere uno ZX81, migliorarne la grafica aggiungendo i colori (da cui il nome Spectrum). Una volta definito il sistema grafico e le parti restanti dell'hardware, lavorò duramente con Steve Vickers della Nine Tiles per estendere il Basic Sinclair.

Il lavoro si svolse piuttosto in fretta e ne uscì un computer dotato di grafica a colori, maggiore capacità di memoria, un Basic più esteso e una tastiera migliore. Il caratteristico involucro nero era una nuova idea stilistica di Rick Dickinson. La versione con 16 KB di RAM era venduta a 125 Sterline, quella con 48 KB a 175 Sterline.

Inclusa nella confezione vi era una cassetta contenente "HORIZONS", un insieme di programmi dimostrativi realizzati dalla PSION (fig.12) tra cui "THRO' THE WALL" (fig.13, il mio primo video-gioco, n.d.a.).



Lo ZX Spectrum si poteva collegare a una TV a colori e con un buon registratore si potevano caricare/salvare i programmi su nastro senza incorrere nell'odioso messaggio "R Tape

loading error" (la "R" stava per "Recording", n.d.a.).

Per collegare un monitor, valeva lo stesso intervento menzionato in precedenza per lo ZX81.

Rispetto allo ZX81, lo ZX Spectrum si comportava come se girasse automaticamente in modo FAST durante i calcoli e in modo SLOW durante la visualizzazione dei dati [Bon83]. Lo scrolling del video era gestito in modo interattivo durante l'esecuzione di un programma (premendo un tasto dopo la domanda "scroll?") oppure assegnando valori con la POKE alla variabile di sistema "SCR CT" posta nel byte di indirizzo 23692 in RAM.

Il comando UNPLOT era stato soppresso e sostituito da PLOT OVER 1. I programmi registrati per lo ZX81 non potevano essere letti dallo ZX Spectrum. Era possibile solo tradurli e reinserirli (tenendo conto della diversità dell'hardware, dell'organizzazione della memoria e delle variabili di sistema).

La ZX Printer era compatibile.

Paragonato alla concorrenza, il Basic risultava a volte più lento ma alcune caratteristiche interessanti lo rendevano superiore.

Per esempio, come sullo ZX81, la funzione VAL era implementata per effettuare il parsing di una espressione matematica all'interno di una stringa (ciò facilitava la scrittura di programmi didattici in Basic come lo studio delle funzioni, senza modificare i programmi ogni volta che si desiderava calcolare una nuova funzione).

In aggiunta rispetto allo ZX81 era disponibile la funzione VAL\$ che effettuava il parsing di

Marzianetti

...ancora una volta dobbiamo difenderci dall'avanzata inesorabile degli invasori

ISTRUZIONI

TASTI:l=sinistra
3=destra
0=laser

```

5 REM 81
10 REM "E"
15 LET S=PI-PI
20 LET A=VAL "12"
25 LET B=PI-PI
30 LET C=VAL "3"
35 CLS
40 LET T=5
45 LET B=B+VAL "2"
50 IF B=VAL "16" THEN PRINT S.
" HITS";X
55 LET A$=""
60 LET B$=""
65 PRINT AT VAL "16",A-VAL "2"

```

```

70 PRINT AT B,C:A$
75 LET T=T+SGN PI
80 IF T=VAL "50" THEN GOTO VAL
"35"
85 IF INKEY$="" THEN GOTO VAL
"115"
90 LET G=RND*VAL "3"
95 LET A=A+(INKEY$="3")-(INKEY
$="1")
100 PRINT AT B,C:B$
105 LET C=C+(I6)=1.5) AND C<VAL
"7")-(I6<1.5) AND C>SGN PI)
110 GOTO VAL "65"
115 PRINT AT B,A:" "
120 IF A$((A-C)+SGN PI)(<>"") TH
EN GOTO VAL "65"
125 LET A$(A-C TO A-C+2)=" "
130 LET B$(A-C TO A-C+2)=" "
135 PRINT AT B,A-SGN PI;"*"*
140 LET S=S+SGN PI
145 IF A$=""
" THEN GOTO VAL "55"
150 GOTO VAL "65"

```

espressioni stringa (in fig.14 è riportato un programmino dimostrativo, n.d.A.).

```

1000 CLS
1010 LIST
1015 PRINT
1020 LET x=0.5
1030 LET a#="SIN x+cos x"
1040 PRINT VAL a#
1045 PRINT
1050 LET b#="*****ABCDE"
1060 LET c#="a#+b$(1 TO 5)"
1070 PRINT VAL# c#
1080 STOP

1.3570081
SIN x+cos x*****

```

Il set standard di caratteri dello ZX Spectrum si poteva sostituire con un altro set caricato in RAM (cambiando il contenuto del puntatore di sistema "CHARS" con un paio di POKE sugli indirizzi 23606 e 23607) oppure (tramite Basic) modificare alcuni caratteri speciali detti U.D.G. (User-Defined Graphics).

Il dispositivo per il suono era un semplice "buzzer" o cicalino comandato tramite l'istruzione BEEP.

Per la grafica (e altre funzioni) era stata richiesta alla Ferranti una ULA di nuova concezione.

All'area video erano assegnati 6912 bytes di RAM così suddivisi:

- 1) 6144 bytes per avere 256 x 192 punti corrispondenti a 24 righe x 32 colonne di caratteri rappresentati in una matrice 8 x 8 punti,
- 2) 768 bytes (= 24 righe x 32 colonne) per gli attributi video dove ciascun byte memorizzava: 3 bit per il colore di background (sfondo) gestito con PAPER, 3 bit per il colore di foreground (primo piano) gestito con INK, bit di luminosità e bit di lampeggiamento gestiti con BRIGHT e FLASH rispettivamente.

Normalmente per una immagine a colori, si memorizzano in RAM le informazioni sui colori per ciascun punto o pixel ("picture element", n.d.A.).

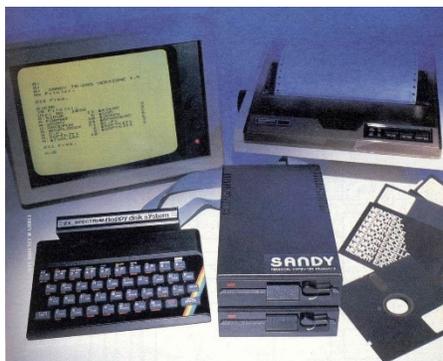
Ma per risparmiare memoria, sullo ZX Spectrum i colori non erano gestiti a livello di pixel ma a blocchi delle dimensioni di un carattere. Ne conseguiva, durante le fasi di una animazione o di un videogioco, il verificarsi del fenomeno detto "colour clash" ("conflitto cromatico", n.d.A.) ossia repentini cambi di colore nel momento in cui due figure colorate diversamente si sovrapponevano sullo schermo. Per aggirare il problema, chi realizzava videogiochi dovette esibirsi in veri e propri virtuosismi di programmazione in linguaggio macchina.

La tastiera consisteva in un foglio di gomma "molliccia" con i tasti in rilievo, sovrapposto alla solita membrana con i contatti. A causa dell'aumentato numero di istruzioni, erano stati previsti 6 differenti livelli di funzioni accessibili con diverse combinazioni di tasti (tra cui gli indispensabili "CAPS SHIFT" e "SYMBOL SHIFT").

Le recensioni sullo ZX Spectrum furono tutto sommato positive. David Tebbutt di "Computer World" dichiarava che era un esempio di come si poteva produrre un computer decente a un prezzo basso, mentre Tim Hartnell di "Your Computer" sosteneva che: "Nonostante qualche difettuccio, lo Spectrum era avanti ai suoi rivali".



Lo ZX Spectrum aveva la stessa interfaccia con l'esterno dello ZX81 (cfr. "edge connector"). Il fatto che non fosse standard, non aveva impedito ad altre aziende di costruire circuiti esterni per ZX Spectrum come sintetizzatori vocali, stampanti, interfacce joystick, interfacce per floppy disc (cfr. OPUS Discovery in fig.15 e SANDY in fig.16).



La stessa Sinclair produsse la ZX Interface 1 per connettere i Microdrive (dispositivi che usavano cartucce contenenti una microbobina di nastro magnetico, fig.17) e la ZX Interface 2 (che accettava cartridge elettronici e gestiva 2 joystick, fig.18). Anche se non ebbero grande successo, testimoniavano la volontà di Sinclair di espandere il sistema.



Oltre a una utilissima connessione RS-232, nella ZX Interface 1 si trovava anche una ROM "fantasma" di 8KB. L'accesso alla ROM "fantasma" avveniva disattivando la ROM principale in due casi: con l'esecuzione di una istruzione assembler "RST 8" oppure quando lo Z80 cercava di eseguire l'istruzione nella locazione 1708Hex della ROM principale.

Leggendo il manuale della ZX Interface 1, uno "smanettone" poteva aggiungere temporaneamente nuove istruzioni Basic sullo ZX Spectrum (con tanto di token richiamato dalla tastiera!). In un articolo apparso sul numero 7 di "Sinclair Computer", Gianluca Carri spiegava l'implementazione della istruzione DRAW! inventata da lui stesso, ricorrendo al linguaggio macchina e valorizzando opportunamente la variabile di sistema "VECTOR" (costituita dagli indirizzi 23735 e 23736).

Chi aveva progettato lo Spectrum non aveva solo in mente di supportare giochi, ma si era sforzato di renderlo attraente anche per eventuali applicazioni professionali. Purtroppo i ritardi nello sviluppo della tecnologia dei Microdrive fecero perdere alla Sinclair la battaglia contro il BBC Micro per la conquista del mercato degli strumenti didattici.

Ad ogni modo il successo mondiale dello ZX Spectrum fu grazie alla sterminata libreria di giochi (https://en.wikipedia.org/wiki/List_of_ZX_Spectrum_games), senza contare le innumerevoli versioni pirata del software "ufficiale" che circolavano allegramente tra i possessori dello ZX Spectrum.

Lo ZX Spectrum fu prodotto in più di 5 milioni di esemplari tra il 1982 e 1988, seguito dallo

ZX Spectrum+ (1984, fig.19) e dallo ZX Spectrum+ 128 (1986, fig.20).



In quest'ultimo modello (prodotto per il mercato spagnolo) la memoria totale (ROM + RAM) era stata portata a 128KB, ben oltre le capacità di indirizzamento dello Zilog Z80. Per gestirla si faceva ricorso a tecniche di paginazione.

Dopo la cessione di Sinclair ad Amstrad (avvenuta nel 1986) furono messe in commercio le versioni ZX Spectrum +2, +2A, +3, +3A, +3B. La produzione fu interrotta definitivamente nel 1992 [BBC07].

Appendice

La produzione dei computer ZX è stata accompagnata da quella di "cloni" più o meno autorizzati costruiti in diverse nazioni.

Sulla pagina Web https://en.wikipedia.org/wiki/List_of_ZX80_and_ZX81_clones si trova l'elenco dei "cloni" per ZX81.

Sulla pagina Web https://en.wikipedia.org/wiki/List_of_ZX_Spectrum_clones si trova l'elenco dei "cloni" per ZX Spectrum.

Alcuni emulatori per ZX81 si trovano sulla pagina web http://www.zx81.it/ZX81_emulator.html, mentre un compendio sugli emulatori per ZX Spectrum è "The ZX Spectrum on your PC" di Colin Woodcock.

L'edizione 2012 si trova liberamente accessibile sulla pagina web https://issuu.com/ccwoodcock/docs/the_zx_spectrum_on_your_pc_second_edition_210420160

Altri siti e pagine Web da segnalare sono:

- 1) un sito per ZX80/ZX81 <https://sinclairzxworld.com/>
- 2) tutto per lo ZX Spectrum <https://www.worldofspectrum.org/>
- 3) l'Italian Fan Club dello ZX81: <http://www.zx81.it>
- 4) un elenco di bug dello ZX Spectrum (completi di spiegazione): <http://skoolkit.ca/disassemblies/rom/reference/bugs.html>
- 5) la realizzazione di un clone "moderno" dello ZX80 <http://blog.tynemouthsoftware.co.uk/2016/12/minstrel-zx80-clone.html>
- 6) la versione 2 del clone Tynemouth <http://blog.tynemouthsoftware.co.uk/2017/01/minstrel-zx80-clone-issue-2.html>
- 7) parti di ZX80/ZX81 in vendita <https://www.tindie.com/products/tynemouthsw/keyboard-overlays-for-ts1000zx80zx81/>
- 8) vendita del clone ZX80 su <https://www.tindie.com/products/tynemouthsw/minstrel-zx80-clone/>
- 9) il manuale per ZX81 digitalizzato http://zxnext.narod.ru/manuals/ZX81_Manual.pdf

Bibliografia

- [BBC07] AA.VV., "How the Spectrum began a revolution", <http://news.bbc.co.uk/1/hi/technology/6572711.stm>, ultima consultazione 2018/10/25.
- [Bon82] R. Bonelli, "Guida al Sinclair ZX81 ZX80 e Nuova ROM", Gruppo Editoriale Jackson, 1982.
- [Bon83] R. Bonelli, "Alla scoperta dello ZX Spectrum", Gruppo Editoriale Jackson, 1983.
- [Ber13] G. Bernardo, "Modifica video composito su Sinclair ZX81", <https://www.associazione64.it/home/tag/zx81/>, ultima consultazione 2018/09/19.
- [Car84] G. Carri, "Una funzione in più", Sinclair Computer n.7, pag. 48, 1984.
- [Chao4] M.R. Chapman, "Alla ricerca della stupidità", Arnoldo Mondadori Editore, 2004.

[CN18] AA.VV., "Tracing the History of the Computer - Sinclair ZX81", <http://www.computernostalgia.net/articles/zx81.htm>, ultima consultazione 2018/09/18.

[Com18] AA.VV., "Sinclair ZX Spectrum 128", <http://www.computinghistory.org.uk/det/2584/Sinclair-ZX-Spectrum-128/>, ultima consultazione 2018/10/25.

[Dil16] R. Dillon, "The Golden Age of Video Games: The Birth of a Multibillion Dollar Industry", CRC Press, 2016.

[LA12] R. Levene, M. Anderson, "Grand Thieves & Tomb Raiders", Aurum Press, 2012.

[Lai04] G. Laing, "Digital retro", Sybex, 2004.

[Lea16] T. Lean, "Electronic Dreams: How 1980s Britain Learned to Love the Computer", Bloomsbury Publishing, 2016.

[LM11] W. Lidwell, G. Manacsa, "Deconstructing Product Design", Rockport Publishers, 2011.

[Mar82] G. Marano, "66 prograami per ZX81 e ZX80 con nuova ROM + hardware", Gruppo Editoriale Jackson, 1982.

[Pur14] B.I. Purcaru, "Games vs. Hardware. The History of PC video games: The 80's", Purcaru Ion Bogdan, 2014.

[Rya11] D. Ryan, "History of Computer Graphics", Author House, 2011.

[Reg16] G. O'Regan, "Introduction to the History of Computing: A Computing History Primer", Springer, 2016.

[Sta15] R. Stanton, "A Brief History Of Video Games: From Atari to Virtual Reality", Hachette UK, 2015.

ChimicaWare: un software didattico per ZX Spectrum

di Alberto Apostolo

Esclusiva RetroMagazine

A metà degli anni '80 del XX secolo, tra le matricole di Ingegneria dell'Università di Firenze che seguivano i corsi di Chimica, circolavano copie di una musicassetta contenente ChimicaWare, un insieme di programmi didattici per esercitarsi con i concetti appresi durante le lezioni.



Il software girava sul Sinclair ZX Spectrum e poteva essere utilizzato con profitto anche da parte degli studenti delle scuole medie superiori.

Il pacchetto fu realizzato ad Ancona nel 1985 da **Liberato Cardellini** (classe 1949), oggi docente di Chimica presso la Facoltà di Ingegneria dell'Università Politecnica delle Marche. Tra le sue numerose pubblicazioni si trova "Strategie per il problem solving in chimica".

Chiamato ad assolvere diversi incarichi, si occupa anche di didattica della Chimica ed è coordinatore del progetto europeo **PROFILES**: <http://www.profiles.univpm.it>.

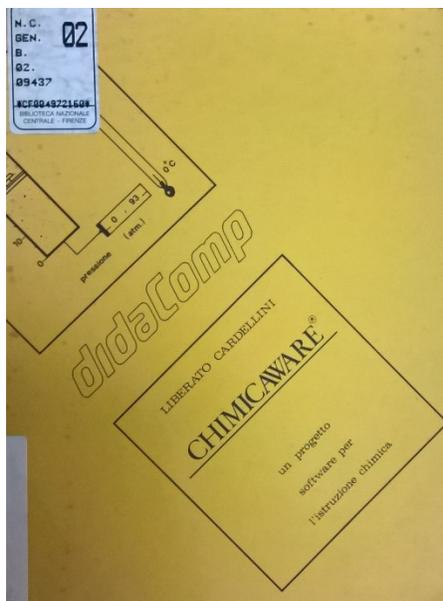
PROFILES è l'acronimo di "Professional Reflection-Oriented Focus on Inquiry Learning and Education through Science", e ha il fine di promuovere l'educazione scientifica attraverso l'approccio IBSE (Inquiry-based science education).

Per chi desidera sapere qualcosa di più su tale progetto, oltre al sito, può seguire su YouTube il discorso tenuto dal prof. Cardellini alla conferenza "Fondamenti educativi per l'educazione scientifica" che si è svolta a Perugia il 18 settembre 2014 (<https://www.youtube.com/watch?v=4z7MdYphtzA>).

Al momento ChimicaWare è introvabile in Rete. Facendo ricerche per verificare l'esistenza di citazioni, ho trovato su OPAC SBN (Catalogo del Servizio Bibliotecario Nazionale) un riferimento a una monografia

di 222 pagine intitolata "Chimicaware® : un progetto software per l'istruzione chimica".

Due copie si trovano rispettivamente presso la Biblioteca Nazionale Centrale di Firenze e di Roma (nel paragrafo di Appendice si troveranno ulteriori informazioni).



Grazie alla autorizzazione concessa gentilmente dal Prof. Cardellini, dal sito di RetroMagazine è scaricabile (gratuitamente) l'intero pacchetto costituito da file in formato SNA ("snapshot") perchè circa vent'anni fa, i programmi memorizzati su nastro furono "migrati" su un Pentium 166 MMX usando alcuni emulatori per ZX Spectrum in grado di interpretare i segnali sonori campionati su file audio WAV (il formato audio richiesto e i parametri di campionamento dipendono dall'emulatore scelto e di solito sono documentati).

Come si può facilmente intuire dalla estensione, un file ".SNA" contiene una "fotografia" dello stato della macchina emulata (nel nostro caso, 27 bytes di header seguiti dai 49152 bytes della intera RAM di uno ZX Spectrum 48K).

Per i particolari si può consultare la pagina web

<http://rk.nvg.ntnu.no/sinclair/faq/fileform.html> e se uno vuole "smanettare" per vedere cosa c'è dentro, esistono programmini

gratuiti per fare il dump esadecimale come HxD (un Hex-editor scritto da Maël Hörz).

Per fare girare i programmi di ChimicaWare è presente in Rete un'ampia scelta di emulatori. Alcuni di essi possono girare sui computer più nuovi, dotati dei sistemi operativi più aggiornati, e addirittura sugli smartphone.

Per la stesura di questo articolo è stata usata una vecchissima ma affidabile versione gratuita del PJPP di Arnt Gulbrandsen (lanciato sotto DOSBox), il quale però mostra in sovra-impressione la percentuale del volume audio.

Si ricorda che di solito, negli emulatori per ZX Spectrum, i tasti "Shift" del PC equivalgono a "CAPS SHIFT", il tasto "Alt" equivale a "SYMBOL SHIFT", il tasto Invio corrisponde a "ENTER" e che sono usate le file di tasti: 1-0, Q-P, A-L (con "Invio"), Z-M (con i tasti "Shift", "Alt", "Spazio").

Ciascun programma del pacchetto ChimicaWare fornisce, in esecuzione, una presentazione e le istruzioni per l'uso. Generalmente, si usano i tasti "5" (sinistra), "6" (giù), "7" (su), "8" (destra) ed "ENTER".

A volte è richiesta la pressione di tasti in maiuscolo mediante "CAPS SHIFT" (per es. "S" maiuscola ed "N" maiuscola per rispondere "Sì" o "No").

Ogni programma è indipendente l'uno dall'altro. Tuttavia è suggerito di rispettare la sequenza sottostante:

- 1) **PREIST**, è il programma che contiene la presentazione di ChimicaWare,
- 2) **CONFEL**, descrive la configurazione elettronica di ciascun elemento (il nome dell'elemento si indica in minuscolo, per es. "sodio" e non "Sodio" oppure "SODIO"),
- 3) **SIMVAL**, consente di esercitarsi con simboli e valenze. Occorre fare attenzione a digitare correttamente la sigla dell'elemento con maiuscole e minuscole (per es. "Na"),
- 4) **NOME**, nomenclatura inorganica di ossidi e anidridi secondo la nomenclatura IUPAC,
- 5) **PREFOR**, è il programma che contiene la presentazione del programma FORMULA,

6) **FORMULA**, consente di fare esercizi con le formule dei composti. Durante la scrittura delle formule occorre fare bene attenzione ai simboli, i numeri degli atomi e le parentesi (dove necessarie) perché il programma in caso di errore costringe a digitare tutto da capo,



7) **CALCFOR**, per il calcolo delle percentuali e la formula minima,

8) **SINCOM**, (sintesi di composti) dove si deve determinare il bilanciamento delle reazioni chimiche elementari,

9) **BILANCIA**, dove si deve determinare il bilanciamento delle reazioni di metatesi e di ossido-riduzione,

10) **REDOX**, per determinare il bilanciamento di reazioni di ossido-riduzione (attenzione a non esagerare con il tasto di Aiuto, perché dopo 2 volte si disattiva)

11) **METATESI**, permette di imparare se una reazione a doppio scambio può avvenire oppure no,

12) **REALIM**, per il calcolo del reagente limitante,

13) **GAS1**, nel quale sono simulate le leggi dei gas (occorre fornire 10 coordinate per portare a termine la simulazione),

14) **WAALS**, per fare esercizi sulle leggi di Van der Waals,

15) **EQUOGAS**, per gli esercizi sugli equilibri gassosi,

16) **CINETICA**, sulla cinetica delle reazioni (consecutive/opposte),

17) **PREION**, presentazione del successivo gruppo di programmi inerente agli equilibri ionici (ZERO, UNO, DUE, TRE, KPS123),

18) **ZERO** (acidi e basi forti),

19) **UNO** (acidi deboli monoprotici e basi deboli),

20) **DUE** (acidi deboli biprotici),

21) **TRE** (acidi deboli triprotici),

22) **KPS123**, per i prodotti di solubilità nelle tre casistiche per sali derivati da acidi deboli: monoprotici, biprotici, triprotici. Bisogna avere pazienza con i tempi di calcolo che

possono essere un po' lunghi (circa 30 secondi). Per questo motivo è meglio provare il programma inserendo dati pertinenti,



23) **TITOLA**, simula una titolazione tra un acido monoprotico forte (HCl) e una base forte (NaOH). Bisogna attendere con pazienza circa 20-25 secondi prima di accedere al menù. Si usano i tasti "p" ed "l" minuscoli. Al termine della titolazione occorre indicare le tre cifre che rappresentano il volume di acido presente (si sceglie una cifra con i tasti cursore "5" e "8" e poi si preme "ENTER").



Questo software così ben curato, è ancora valido e utilizzabile al giorno d'oggi sui nostri portatili (ammesso di procurarsi un emulatore per ZX Spectrum).

Siccome meritava qualcosa di più di una sintetica recensione da parte mia, ho provato a rivolgermi al Prof. Cardellini (tramite il suo indirizzo mail accademico) per fare alcune domande e conoscere qualche particolare in più sulla realizzazione del suo progetto didattico.

Ebbene, ho potuto contare sulla sua disponibilità a dedicarmi parte del suo prezioso tempo. Per queste ragioni, desidero porgere i miei più sinceri ringraziamenti.

L'intervista al Prof. Liberato Cardellini

Come e quando è nata l'esigenza di avere a disposizione un software didattico per esercitarsi con la Chimica?

Più che esigenza, per il desiderio di aiutare gli studenti ad imparare in modo più divertente. Avevo già realizzato molti programmi di calcolo per la calcolatrice programmabile HP41C.

La pianificazione e la realizzazione di ChimicaWare è legata a un preciso incarico accademico oppure è il frutto di una iniziativa personale?

La seconda.

E' stato influenzato da progetti simili, già realizzati presso altre sedi?

No; ho seguito l'ispirazione.

Ricorda qualcuno in particolare (tra familiari, amici, colleghi) dal quale ha ricevuto ispirazione e sostegno in questo progetto?

È una domanda da non fare, nel senso che ho spesso avuto le giuste lamentele della moglie.

ChimicaWare è stato pianificato fin dall'inizio per ripercorrere un programma ministeriale sull'insegnamento della Chimica oppure i moduli software si sono accumulati con il passare del tempo?

Pianificato fin dall'inizio.

Per sviluppare ChimicaWare si è avvalso anche dell'aiuto di qualche collaboratore specializzato in Informatica?

Purtroppo, no.

Perché è stato scelto proprio il Sinclair ZX Spectrum e non un altro modello di computer?

Per il costo; era l'unico che all'epoca potevo permettermi.

Quali accorgimenti sono stati adottati per superare le difficoltà legate alle numerose casistiche presenti negli argomenti trattati in Chimica (per es. nomenclatura, simbologie, formule)?

Numerosi, anche se ora non ricordo i particolari.

Quanto tempo è stato impiegato per realizzare programmi funzionanti?

Lunghissimo.

Ci sono stati momenti nei quali ha pensato di non riuscire a realizzare il progetto come desiderava?

Direi di no; ero determinato ad arrivare alla fine.

Ha avuto la possibilità di misurare quanto ChimicaWare si sia dimostrato uno strumento didattico utile?

118 Raccolte di una versione con 9 programmi sono state distribuite gratuitamente prima della conclusione del progetto e la valutazione (Livello scientifico, Approccio educativo e Operatività dei tasti) è stata positiva: da 7.4 a 9.2 su 10. (n.d.R.: nel valutare ciascuna delle tre caratteristiche summenzionate è stata fatta l'assunzione: Scadente=2, Insufficiente=4, Medio=6, Buono=8, Eccellente=10. Con tale conversione numerica è stato possibile calcolare delle medie aritmetiche. Considerando i dati in monografia relativi ai 9 programmi esaminati si hanno Livello scientifico, Approccio educativo, Operatività con una media di 8.2, 8.5, 8.1 rispettivamente).

Con il senno di poi, ci sono cose che rifarebbe diversamente o non rifarebbe affatto?

Non saprei.

Se le chiedessero di riprogettare ChimicaWare usando un moderno ambiente di programmazione, accetterebbe la sfida?

No!

Grazie ancora per la cortesia e buon lavoro!



Conclusioni (a cura del Prof. Cardellini)

Questo progetto è nato dal mio interesse per il computer e la mia passione per la didattica. Negli anni '70 ho utilizzato l'IBM 360 per eseguire calcoli di Chimica teorica. Oltre al FORTRAN, negli anni successivi ho utilizzato il PL/1; quando ho deciso di sviluppare i programmi ChimicaWare ho utilizzato il BASIC, perché disponibile nello Spectrum. È stato scelto il Sinclair ZX Spectrum perché gli altri computer erano molto più costosi. Aveva

oltre alle capacità di calcolo anche discrete possibilità grafiche.

Il progetto ha avuto inizio nel 1983 ed ho presentato le idee e alcuni programmi alla 7th International Conference on Chemical Education, a Montpellier, in Francia. In quegli anni c'era molta fiducia nelle capacità e possibilità dell'uso del computer nell'insegnamento e alla 8th International Conference on Chemical Education, a Tokyo nel 1985 e nella conferenza di apertura del congresso è stato affermato che *"We are sure that education which neglects computers and images will become obsolete in a very near future"*.

Il software è stato auto-prodotto; al programma NOME ha partecipato il Prof. Jean-Paul LeBlanc, dell'Université de Moncton, Canada, mentre in visita perché interessato al progetto ChimicaWare. Fin dall'inizio il progetto è stato pianificato per rispondere a delle esigenze didattiche; la sua realizzazione ha richiesto un tempo lunghissimo, molte notti e week end davanti al televisore per scrivere e verificare la funzionalità dei programmi. Alle volte sono dovuto ricorrere a degli escamotage per superare le limitazioni del linguaggio. Ecco perché non rifarei di nuovo un lavoro simile. Anche perché oggi la tecnologia offre diverse possibilità, probabilmente per la sua diffusione lo smartphone è lo strumento più adatto per usi didattici.

Programma	Livello	Approccio	Operatività
SIMVAL	8.0	8.7	8.2
NOME	7.5	8.0	8.5
FORMULA	8.4	9.1	8.6
CALCFOR	8.2	7.8	8.2
SINCOM	8.4	8.9	8.7
BILANCIA	8.8	9.2	8.4
REALIM	8.5	7.6	7.4
GAS1	8.4	8.9	8.5
TITOLA	7.4	8.4	6.4

Appendice

Ho avuto modo di consultare (presso la Biblioteca Nazionale Centrale di Firenze) la monografia "Chimicaware" che contiene alcune note introduttive sulla didattica e computer, le istruzioni d'uso dei programmi del pacchetto precedute da brevissimi richiami teorici e da esempi numerici.

La copia consultata è assai delicata e si sta "spaginando". Pertanto il cortese personale della Biblioteca può concedere solo la "consultazione in Sede".

Purtroppo immagino che anche per la copia in custodia a Roma, valga la stessa disposizione.

Una B.N.C. può mettere on-line un'opera solo quando sono scaduti i 70 anni dalla morte dell'autore. Tuttavia se un'opera è fuori commercio è possibile richiedere a una B.N.C. la riproducibilità al 100% (in formato cartaceo o elettronico) con pagamento secondo il tariffario vigente. È obbligatorio compilare la "Dichiarazione Sostitutiva dell'Atto di Notorietà" (dove si deve fornire il proprio codice di iscrizione alla Biblioteca e autocertificare sotto la propria responsabilità che l'opera è fuori commercio). Per avere informazioni più complete e corrette sulle modalità di riproduzione richieste da "remoto", si raccomanda di rivolgersi direttamente alle

Biblioteche stesse usando i recapiti telefonici indicati nei loro siti Internet (<http://www.bncrm.beniculturali.it/>, <http://www.bncf.firenze.sbn.it/>).

Chi ha la fortuna di recarsi facilmente a Roma o a Firenze, compilando sempre detta Dichiarazione, può fare richiesta per riprodurre un'opera con mezzi propri (per es. tablet, smartphone ecc.ecc. nel rispetto delle norme di comportamento fissate). Ogni riproduzione è solo per esclusivo uso personale. Per la B.N.C. di Firenze le norme di comportamento si trovano in <http://www.bncf.firenze.sbn.it/allegatonotizia/538.pdf> e il modulo per la Dichiarazione si trova in http://www.bncf.firenze.sbn.it/documenti/RegolamentoInterno/Modulo_dichiarazione_sostitutiva_a_tto_di_notoriet.pdf

Per avere la certezza che la monografia "Chimicaware" fosse fuori commercio, mi sono avvalso del servizio gratuito di ricerca offerto dalla Liberia Atlantide di Castel San Pietro (BO) (<https://buonelettture.wordpress.com/>).

Inoltre il prof. Cardellini (interpellato in proposito via e-mail) mi ha confermato che la monografia è pienamente riproducibile. Per prenotare la lettura della monografia occorre fornire il suo esatto codice di collocazione in Biblioteca (per Firenze "GEN B2 09437", per Roma lo ignoro). I codici di collocazione si trovano consultando i cataloghi BCNF e BCNRm.

Come detto in precedenza, una riproduzione è per uso personale. Quindi non chiedete a RetroMagazine di mettere la monografia on-line. Non si può (altrimenti lo avremmo già fatto volentieri aggiungendo un altro link nel sito).

Poi chissà, altre copie della monografia potrebbero essere custodite presso le Biblioteche di alcune Università italiane.

Qui di seguito il link per scaricare il software ChimicaWare dal sito di RetroMagazine:

<http://www.retromagazine.net/download/ChimicaWare.zip>

Ghost'n COBOL - parte 2 - I file sequenziali

di Francesco Fiorentini

Dopo la doverosa introduzione del linguaggio COBOL affrontata nel numero scorso e soprattutto dopo aver visto come compilare un semplice *Hello World* con l'Abacus Cobol 64 sul nostro biscottone, direi che sia finalmente arrivato il momento di vedere qualcosa di piu' impegnativo.

Prima di cominciare a fare sul serio, vorrei che teneste sempre a mente che si', stiamo parlando del COBOL, uno dei linguaggi piu' utilizzati in ambito professionale fino a pochi anni fa, ma in questo caso specifico parliamo di un compilatore COBOL da usarsi con il C64, un computer con appena 64 KB di RAM. Ovviamente il dialetto COBOL sviluppato con il compilatore dalla Abacus dovra' tenere per forza di cose conto di alcune limitazioni imposte dall'hardware.

ATTENZIONE: a costo di sembrare ripetitivo vi invito a tenere a portata di mano questa tabella. Per chi e' abituato a scrivere programmi in BASIC l'indentazione riveste un aspetto secondario, ma il COBOL invece non fa sconti.

Colonne	Descrizione
1-6	Numero di riga, colonne ignorate dal compilatore.
7	Caratteri speciali. Un asterisco "*" indica una riga commentata.
8-11	Area A - DIVISION, SECTION, Intestazione di procedure e livelli 01 e 77.
12-72	Area B - Tutto il resto che non è ammesso in Area A.
73-80	Area usata per indicare la numerazione delle schede perforate. Ignorata dal compilatore.

Come anticipato nella prima parte della guida, la vera forza del COBOL e' la gestione degli archivi e l'Abacus Cobol 64 permette di utilizzare 2 diversi tipi di archivio per la memorizzazione dei dati: i file **Sequenziali** ed i file **Relative**.

Un file **sequenziale** e' un archivio che puo' essere acceduto solo in sequenza. Se per esempio avessimo la necessita' di leggere il record numero 10 dovremmo scorrere tutti i 9

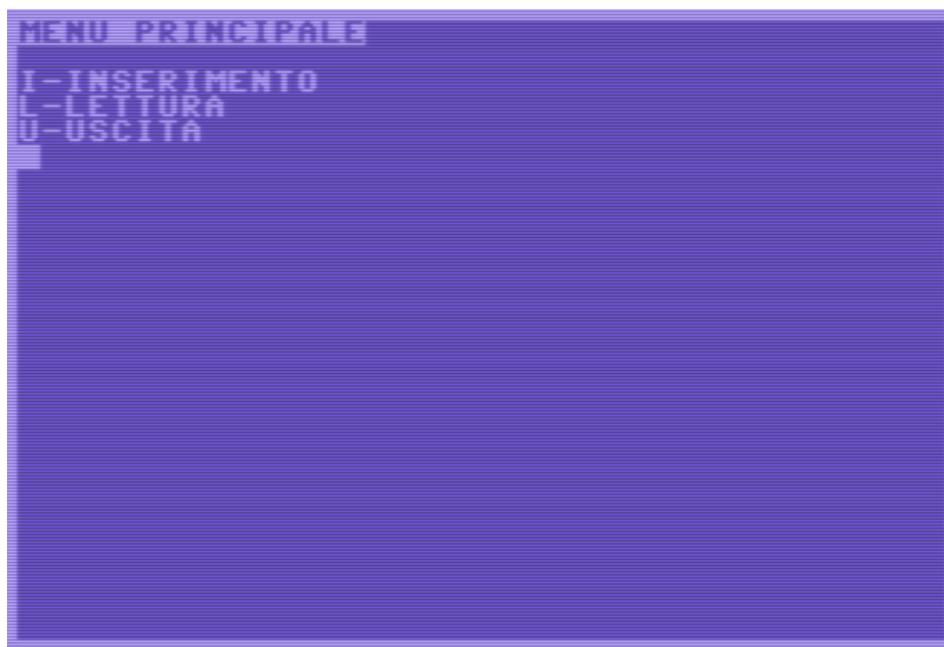


Figura 1 – Il menu' principale del nostro programma in COBOL

record precedenti prima di posizionarci al record che ci interessa.

Un file **relative** e' invece un archivio che puo' essere acceduto sequenzialmente, come nei file sequenziali, oppure dinamicamente, usando una chiave come indice. Nel secondo caso per esempio, se volessimo leggere il record numero 10 potremmo, utilizzando l'opportuna chiave, accedere direttamente al record che ci interessa, saltando a pie' pari i 9 record precedenti.

Per semplicita', nella nostra prima vera prova, utilizzeremo un archivio sequenziale. Inoltre ho riscontrato una limitazione piuttosto singolare dell'Abacus Cobol 64 nella gestione di questo tipo di file e vorrei condividerla con voi.

Ma cominciamo con ordine; nella **input-output section** dobbiamo definire il file che andremo ad utilizzare assegnandolo alla periferica corretta.

```
input-output section.
file-control.
  select data1 assign to disk-1541 drive-8
  file status is file-st.
```

In questo caso il nostro file avra' come identificativo l'etichetta **data1** e verra' memorizzato sul **disk drive 1541** identificato come periferica **8**. Inoltre, non specificando nessuna informazione circa la tipologia di file da creare, il compilatore assumerà per **default** di creare un **file sequenziale**.

Il **file status**, identificato nella riga successiva come **file-st**, verra' utilizzato per controllare, dopo ogni operazione effettuata sul file, se si siano verificati errori o se tutto sia andato come previsto.

Dopo aver definito la tipologia del nostro file dobbiamo assegnargli un nome. Per fare questo dobbiamo usare il costrutto **fd** (file description) nella **file section** della **data division**.

```
data division.
file section.
fd data1
  label records are omitted
  value of file-id is "@o:data1".
```

E sempre nella file section andremo a definire il tracciato record del nostro file. In questo caso ho optato per un banalissimo archivio

con nome, cognome ed indirizzo di una ipotetica rubrica:

```
01 data-record.
  02 nome pic x(15).
  02 cognome pic x(15).
  02 indirizzo pic x(20).
```

Il costrutto **PIC X(nn)** indica al compilatore di creare un campo alfanumerico di (nn) caratteri. Nel caso di un campo numerico avremmo invece dovuto specificare l'opzione **PIC 9(nn)**.

A questo punto il terreno per il nostro archivio e' preparato e possiamo passare a scrivere le istruzioni vere e proprie del nostro programma, non prima pero' di aver definito ed inizializzato nella **working-storage section** le variabili che utilizzeremo nel codice.

```
working-storage section.
77 scelta pic x.
77 clear-home value chr 147 pic x.
77 return-codice value chr 13 pic x.
77 rvs-on value chr 18 pic x.
77 file-st pic xx.
```

Interessante notare come tutte le variabili che useremo nel nostro programma devono essere definite come livello **77**. Questo livello specifica le **variabili indipendenti**, ovvero non considerabili strutture dati come avvenuto invece per il tracciato record definito nella file section a livello **01** e con un sottolivello **02**. Come detto nella prima parte del corso tutte le variabili in COBOL sono da considerarsi **globali**.

Anche in questo caso si tratta di variabili alfanumeriche, ma per 3 di esse (clear-home, return-codice e rvs-on) c'e' da fare una piccola digressione. L'Abacus Cobol 64 e' in generale povero di istruzioni e, come se non bastasse, le poche istruzioni a disposizione hanno anche poche opzioni... Per esempio, non esiste un'istruzione per pulire lo schermo. Non esiste nemmeno un'opzione del comando **DISPLAY** per scrivere in reverse mode (molti compilatori hanno a disposizione l'opzione **REVERSE-VIDEO**). Per fortuna ci viene in aiuto il Commodore 64 con i control character che possiamo assegnare direttamente in fase di definizione delle nostre variabili.

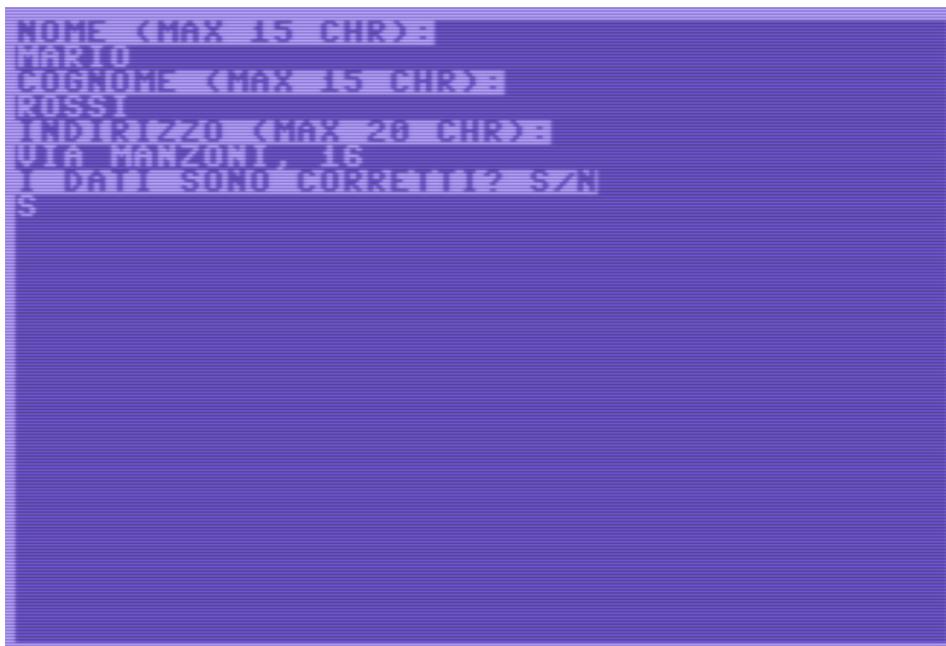


Figura 2 – La fase di inserimento dei record, prima dell'effettiva scrittura su disco.

A questo punto non c'e' bisogno che vi spieghi cosa facciano le 3 variabili in questione, date un'occhiata alla lista dei control character riportata in https://www.c64-wiki.com/wiki/control_character e provate ad indovinarlo da soli. 😊

Bene, a questo punto non ci resta altro che cominciare a scrivere la parte di codice che gestira' il tutto. Dove? Ma nella **Procedure Division** ovviamente.

Come potete evincere dal listato e dagli screenshot allegati, la logica del programma e' volutamente semplice. Presenteremo a video un menu' con una triplice scelta: i per inserire nuovi record, l per leggere i record gia' presenti sul file, u per uscire dal programma.

Una regola fondamentale in COBOL va sempre tenuta a mente quando si lavora con gli archivi: **scrivi record, leggi file!** Vuol dire che quando leggiamo il file con il comando **READ** dobbiamo fare riferimento al nome del file, nel nostro caso **read data1**. Quando invece scriviamo il file, con il comando **WRITE**, dobbiamo fare riferimento al record che vogliamo scrivere, nel nostro caso **write data-record**. Perche' questa differenza? Perche' il COBOL permette di gestire anche strutture record differenti all'interno dello stesso file e quindi il compilatore deve sapere quale record andremo a scrivere. Ma questa e' un'altra storia e francamente mi sembrerebbe

premature parlare di tracciati record disomogenei gia' a questo punto.

Vi ricordo che prima di ogni lettura e scrittura il file deve essere sempre aperto, tramite l'istruzione **OPEN** e prima di terminare il programma e' sempre buona norma chiudere tutti i file aperti tramite il comando **CLOSE**.

```
000100 identification division.
000200 program-id. test.
000250 author. retromagazine.
000300 environment division.
000400 configuration section.
000500 source-computer. c64.
000600 object-computer. c64.
000601 input-output section.
000602 file-control.
000603 select data1 assign to disk-1541
drive-8
000606 file status is file-st.
001000 data division.
001001 file section.
001002 fd data1
001003 label records are omitted
001004 value of file-id is "@@:data1".
001010 01 data-record.
001011 02 nome pic x(15).
001012 02 cognome pic x(15).
001013 02 indirizzo pic x(20).
001100 working-storage section.
001101 77 scelta pic x.
001102 77 clear-home value chr 147 pic x.
001103 77 return-codice value chr 13 pic
x.
001104 77 rvs-on value chr 18 pic x.
001114 77 file-st pic xx.
001300 procedure division.
001400 mostra-menu.
001401 display clear-home.
001499 display rvs-on "menu principale"
return-codice.
001500 display " " return-codice.
```

```

001501 display "i-inserimento" return-
codice.
001502 display "l-lettura" return-
codice.
001503 display "u-uscita" return-
codice.
001504 accept scelta.
001505 if scelta equal to "u"
001506 display rvs-on
"arrivederci..."
001600 stop run.
001601 if scelta equal to "l" perform
leggi-file thru end-leggi-file.
001602 if scelta equal to "i" perform
scrivi-file thru end-scrivi-file.
001603 go to mostra-menu.
001700 leggi-file.
001701 display rvs-on "-- lettura dati
--" return-codice.
001702 open input data1.
001703 if file-st not equal to "00"
001704 display "errore in apertura
file"
001705 stop run.
001706 leggi-loop.
001710 read data1 at end go to end-
lettura.
001711 if file-st not equal to "00"
001712 display "errore in apertura
file"
001713 stop run.
001720 display "nome: ".
001725 display nome return-codice.
001730 display "cognome: ".
001735 display cognome return-codice.
001740 display "indirizzo: ".
001745 display indirizzo return-codice.
002450 go to leggi-loop.
002490 end-lettura.
002492 accept scelta.
002493 close data1.
002494 if file-st not equal to "00"
002495 display "errore in chiusura
file"
002496 stop run.
002497 perform mostra-menu.
002498 end-leggi-file.
002499 exit.
002500 scrivi-file.
002501 display rvs-on "-- inserimento
dati --" return-codice.
002505 open output data1.
002506 if file-st not equal to "00"
002507 display "errore in apertura
file"
002508 stop run.
002510 accetta-dati.
002511 display clear-home.
002530 display rvs-on "nome (max 15
chr):" return-codice.
002540 accept nome.
002550 display rvs-on "cognome (max 15
chr):" return-codice.
002560 accept cognome.
002570 display rvs-on "indirizzo (max
20 chr):" return-codice.
002580 accept indirizzo.
002590 display rvs-on "i dati sono
corretti? s/n" return-codice.
002600 accept scelta.
002620 write data-record.
002630 if file-st not equal to "00"
002640 display "errore in scrittura
file"

```



Figura 3 – Il record letto dal file e mostrato a video.

```

002650 stop run.
002660 close data1.
002670 if file-st not equal to "00"
002680 display "errore in chiusura
file"
002690 stop run.
002700 perform mostra-menu.
002710 end-scrivi-file.
002711 exit.

```

Se a questo punto avete già copiato il codice, compilato il tutto ed eseguito il programma vi sarete accorti di una cosa. Il programma permette effettivamente di inserire un solo record. Tutte le volte che proverete ad aggiungere un nuovo record il programma non farà altro che sovrascrivere il record precedente. E' un errore? E' un bug? No, e' semplicemente il comportamento che ho voluto che il programma tenesse per parlarvi di una limitazione che ho riscontrato nell'Abacus Cobol 64 compiler. Avrei potuto scrivere il programma diversamente ma in questo modo posso facilmente mettervi a parte del problema.

Ebbene, con mia somma sorpresa ho scoperto che l'Abacus Cobol 64 ha soltanto 2 opzioni per l'apertura di un file sequenziale. Come riportato anche nel manuale, il comando **OPEN** riferito ad i file sequenziali accetta soltanto i parametri **INPUT** e **OUTPUT**.

Fin qui niente da eccepire, con INPUT il file viene aperto in sola lettura, mentre con OUTPUT il file viene aperto in scrittura. Il problema e' che quando il file viene aperto in

OUTPUT non c'è modo di specificare nessuna opzione per l'APPEND. In pratica il file viene sempre sovrascritto, vanificando di fatto la possibilità di aggiungere record all'archivio. Onestamente a me sembra una grossa limitazione per un compilatore che prometteva di insegnare ad usare il COBOL sul nostro Commodore 64. Se tra i lettori qualcuno trovasse il famigerato parametro e' pregato di farmelo sapere. E no, non sono ne' EXTEND, ne' APPEND, li ho provati anche io!

Prima di chiudere vi lascio i compiti a casa. Il codice e' tutto tranne che completo. Vi invito a provare ad inserire, per esempio, i controlli necessari per evitare che i valori accettati per nome, cognome ed indirizzo siano non conformi al tracciato record (nel manuale potrete trovare suggerimenti).

Links utili

Abacus Cobol 64:

<https://www.commodoreserver.com/PublicDiskDetails.asp?DID=0FE97DA68C4044D2B30B2106828ACB28>

Manuale dell'Abacus Cobol 64:

https://www.lyonlabs.org/commodore/onrequest/COBOL-64_Software_System.pdf

Control character:

https://www.c64-wiki.com/wiki/control_character

NB: le istruzioni per compilare il programma si trovano nella prima parte della guida, nel numero precedente (il 10) di RetroMagazine.

Un fantasmagORICO home computer - parte 2

Atmos, Stratos ed il tramonto di Oric

di David La Monaca (Cercamon)

Nella prima parte, cari folli appassionati di Oric e amici del retrocomputing, avete appreso della rocambolesca nascita e anche della prematura scomparsa del primo home computer di Oric: il modello Oric-1. E sulla scia, o sarebbe meglio dire dalle ceneri di questo sfortunato computer nacque il nuovo modello denominato Atmos, una macchina dallo stile rinnovato ed elegante, finalmente con dei veri tasti di buona qualità e con la scocca dai colori vividi rosso e nero (fig. 1). Un notevole passo avanti rispetto all'Oric-1 ed un passo avanti anche a livello industriale perché l'Atmos fu progettato fin dall'inizio con l'obbiettivo di tenere bassi i costi di produzione per essere più competitivi su un mercato ormai divenuto, a metà degli anni Ottanta, molto agguerrito, in UK e in Europa.

Ma, per dirla con un vecchio proverbio, non era tutto oro quello che luccicava: l'Atmos non costituiva poi tutto questo grande upgrade o miglioria hardware. In pratica si trattava semplicemente di una motherboard versione 4 dell'Oric-1, incastonata in un nuovo case progettato da Paul Durgin. L'unico vero aggiornamento era contenuto nella ROM di sistema, ora portata alla versione 1.1, per apportare miglioramenti in generale e per



Figura 1 - Oric Atmos

cercare di rimediare ai bug conclamati della prima release. Ma purtroppo anche la nuova versione presentava alcuni problemi (primo fra tutti, ancora l'interfaccia di gestione del registratore a cassette protagonista in negativo). Oric produsse persino una patch per la cassetta di benvenuto allo scopo di tentare di evitare agli utenti gli errori più gravi. Ma c'erano anche dei lati positivi nella release 1.1 e i bug peggiori della versione 1.0

della ROM erano stati corretti, anche se, come vedremo più avanti, a scapito della piena compatibilità del software e dei giochi usciti fino a quel momento.

Eppure l'Atmos non rappresentava la nuova macchina che molti fan dell'Oric si aspettavano. E se è per questo, anche la stampa specializzata non ci andò con mano leggera: "Non capisco proprio come l'Atmos possa essere considerato una novità", brontolò Bryan Skinner, recensendo la macchina per conto della rivista Your Computer. Non c'è da stupirsi, quindi, che nel febbraio del 1984 Barry Muncaster stesse già parlando di un ulteriore modello successore dell'Atmos. "Il nome del probabile nuovo computer sarà Stratos" (fig. 2), riportava Personal Computer News. Dunque una terza macchina targata Oric, che, a detta della dirigenza, sarebbe uscita sul mercato persino prima delle periferiche (modem, unità a dischetti e stampante) che erano state promesse appena dopo la commercializzazione del primo home computer della serie! Una bizzarra strategia di mercato da parte di Oric, che, come vedremo, non porterà molto lontano.

Gli utenti dovettero aspettare fino all'ottobre del 1984 per vedere nei negozi il modem e Personal Computer News non mancò di sottolineare sarcasticamente l'evento affermando: "Oric ha prodotto l'equivalente nel mondo dei microcomputer della prova che il mostro di Loch Ness esiste: ha messo in commercio un modem!" Ben cinque mesi di ritardo, nonostante Oric Products International avesse annunciato nel marzo precedente che aveva già le confezioni del modem pronte per la vendita. Ebbene sì, la produzione ed il marketing dei prodotti Oric era un mondo a dir poco confuso. Un altro annuncio comparve sul numero 6 di Oric Owner (febbraio/marzo 1984) in cui era pubblicata la notizia che l'Atmos era già in produzione dal 16 gennaio 1984, dopo essere stato presentato ufficialmente al Which Computer Show di Birmingham, tenutosi

soltanto il 17 gennaio! Rob Kimberley si trovava alla fiera per la rivista Oric Computing: "Sorpresa a non finire in casa Oric! Non solo una nuovissima unità a dischetti (in realtà si tratta solo di un prototipo, ancora senza prezzo), ma anche un Oric nuovo di zecca a 170 sterline chiamato Atmos... La definizione di 'atmosfera' nel mio dizionario è "involucro gassoso sferoidale che circonda i corpi celesti". OK, include la nuova ROM versione 1.1, ma è un po' esagerato definirlo "nuovo"! Personalmente non mi piace la nuova tastiera e sono così abituato al bel click sicuro del mio fidato Oric-1 che le tastiere meccaniche "standard" mi sembrano poco familiari e trasandate. Il nuovo modello Oric è un po' più veloce del precedente e viene fornito con un ulteriore "tasto funzione" – il cui utilizzo purtroppo nessuno dei venditori Oric presenti ha saputo spiegarmi (forse per l'eccitazione generale il tasto FUNCT sarà parso un miracchio?!)". Un altro tipico errore di marketing e d'immagine da parte di Oric.



Figura 2 - Oric Stratos sistema completo

Su altre riviste i commentatori sembravano essere più impressionati dall'arrivo dell'ultima novità di Sinclair, l'innovativo e "pionieristico" Sinclair QL che fu lanciato sul mercato nello stesso mese dell'Atmos. Quando fu pubblicata una recensione dell'Atmos, molte altre storie dell'orrore sull'uso del registratore vennero fuori, più imbarazzanti che mai per la casa costruttrice. Un utente descrisse in questo modo la sua personale esperienza cercando di caricare il nastro dimostrativo incluso con il computer: "Seguo le istruzioni e digito il comando per caricare il nastro demo. Nella parte superiore dello schermo appare il messaggio "Searching...", presto seguito da

un rassicurante "Loading * C" e infine "READY". L'istinto mi spinge poi a digitare "RUN" ed attendere fiducioso la partenza del programma, ma il risultato è solo un altro messaggio "READY" e nient'altro. Ulteriori tentativi usando l'altro lato della cassetta, vari riavvolgimenti, l'uso di altri registratori si sono tutti rivelati infruttuosi; e visto che il manuale non fornisce alcuna informazione sulla cassetta dimostrativa, la delusione è stata totale". Sentimento questo purtroppo condiviso da tantissimi altri acquirenti dell'Atmos, tutti speranzosi che l'aggiornamento della ROM avesse finalmente risolto per sempre i problemi di caricamento. Per la cronaca, il "Welcome Tape", una volta partito correttamente, mostra l'animazione di un grande uccello giallo che si libra attraverso lo schermo accompagnato dal brano "The Entertainer" di Scott Joplin. Un'animazione davvero ben fatta che mostra le potenzialità grafiche e sonore del computer e che avrebbe senza dubbio aiutato ad incrementare le vendite, ma quando persino il software fornito in bundle non funziona, agli occhi del pubblico tutto diventa un imbarazzante fiasco. La



Figura 3 - Oric Microdisc drive

maggior parte delle recensioni dell'Atmos si concludevano con la stessa riflessione: "Ad un prezzo di 170 sterline per un home computer con 48K di RAM, l'Atmos è molto più costoso dei suoi concorrenti. Essendo così simile all'Oric-1, il nuovo Atmos offre le stesse funzioni competitive dell'Oric-1 rispetto ad altre macchine con caratteristiche simili e cioè praticamente nessuna."

E come reagì Oric a tutte queste critiche negative? Beh, le ignorarono del tutto e si concentrarono sull'annuncio dei piani per l'Oric-3! Il comandante in capo Barry Muncaster comparve nelle news usando tutto

il potenziale di fuoco e dichiarò: "Il prossimo sistema Oric sarà un microcomputer integrato. Incorporerà infatti disk drive, modem - probabilmente con la funzione autodial - e potrebbe essere basato su Z80. Il nome prescelto è Stratos ed il suo lancio è previsto per la prima metà di quest'anno... Riteniamo che la platea di appassionati acquirenti di home computer stia per mettere fuori gioco molti dei produttori attuali, lasciandone circa quattro fra i migliori per sfruttare al meglio il mercato interno. Oltre a Oric (naturalmente) troveremo (in ordine decrescente) Sinclair, Commodore e Acorn... Più avanti, nel corso dell'anno, speriamo di produrre un altro modello della famiglia Oric, questa volta basato sul processore 8086".

Di lì a poco, il 4 febbraio 1984, Oric organizzò una conferenza stampa per presentare il nuovo microdrive, ma quasi immediatamente dopo la cancellò, nonostante il mondo intero stesse aspettando novità concrete. Qualche mese dopo, nell'aprile 1984, Oric acquistò una nuova fabbrica automatizzata, impianti di produzione e assistenza, affermando che la produzione era cresciuta fino a 1200 macchine al giorno. Con il nome di "M3", questo nuovo stabilimento comprendeva uno spazio di circa 1800 metri quadrati nell'Hampton Farm Industrial Estate (si trovava alla fine dell'autostrada M3) dove il nuovo Atmos 48K sarebbe stato il primo prodotto ad essere realizzato, utilizzando le più recenti tecniche di assemblaggio e collaudo automatizzati. C'erano grandi aspettative che questa mossa strategica avrebbe alleviato molti dei crescenti problemi economici che avevano spinto in alto i costi di produzione dell'Oric-1 l'anno precedente. Un'altra recensione dell'Atmos, quasi un'eccezione rispetto ai giudizi generalmente negativi o neutri, apparve in "What Micro?" quello stesso mese: "Se i problemi di caricamento da nastro sulle prime macchine possono essere risolti, [l'Atmos] dovrebbe andare bene come computer per un uso generale. Certamente non possiede grandi punti di forza che possano distinguerlo dai suoi rivali, ma non ha nemmeno le loro singole debolezze".

Intanto, in Francia, nell'aprile 1984 uscì il primo numero di Théoric, senza dubbio la più professionale rivista per Oric mai pubblicata. La prima stampa di 25.000 copie andarono

esaurite rapidamente, tanto che altre 3.000 copie andarono in ristampa. Oric, si diceva, ora puntava su Germania, Italia e Spagna, e stava discutendo anche un accordo di distribuzione per l'America. E sempre nello stesso mese fu pubblicato dalla software house francese Loriciels uno dei più alti esempi di programmazione su Oric-1: "L'Aigle d'Or", solo la grafica valeva il prezzo di acquisto del titolo. Quasi a premiare il mercato francese arrivò l'annuncio di una tastiera AZERTY per l'Atmos, qualcosa che non era mai accaduto con altri microcomputer. Oric Computing pubblicizzò anche un programma di gestione cartelle cliniche utilizzato a Praga - la prova che la macchina era finalmente penetrata attraverso l'allora Cortina di Ferro.

Nel frattempo, tornando alla saga del rilascio/non rilascio delle periferiche promesse, il disk drive per Oric-1 nel case blu e grigio scomparve definitivamente dai radar e non fu mai prodotto, ma nel numero 7 di Oric Owner, Aprile/Maggio 1984, apparve - finalmente - una pagina pubblicitaria con i disk drive in stile Atmos (rosso e nero) pronti per la vendita al pubblico. I prezzi? Atmos £170, la stampante £150, il Microdisc drive £260 - non esattamente economici per le già martoriolate tasche dei fan dei prodotti Oric. Lo stesso mese la società ITL Kathmill lanciò sul mercato la loro unità a dischi "Byte Drive 500", che era in sviluppo sin dal luglio precedente e che era stata presentata in anteprima alla stampa nel dicembre 1983. La



Figura 4 - L'Aigle d'Or

rivista "What Micro?" recensì positivamente la periferica soprattutto per il suo set esteso di istruzioni rispetto al Microdisc di Oric, ma fu criticata perché la ROM del DOS si trovava sotto la memoria dello schermo e comportava qualche problema di programmazione.

Era già estate quando molti utenti comprarono un Atmos e le cose sembravano finalmente mettersi per il meglio per Oric.

Eppure molti pensavano che Oric non sarebbe sopravvissuta al 1984. Il direttore finanziario Allan Castle dovette diverse volte negare che l'azienda fosse in difficoltà. In agosto affermò che le vendite in corso ammontavano a circa 2 milioni di sterline al mese, toccando anche il record di 2,5 milioni di sterline in giugno. Verso la fine dell'anno, Barry Muncaster vantava vendite di Oric-1/Atmos per oltre 350.000 unità nei primi due anni di vita.

Per inciso, la campagna pubblicitaria comparativa per l'Atmos, che lo metteva a confronto favorevole con il Commodore 64, fu oggetto di aspre critiche alla fine di aprile. L'Advertising Standards Authority accolse il reclamo della Commodore riguardante



Figura 5 - Space 1999

l'affermazione che "il C64 perde 26K della sua memoria da elefante per la grafica ad alta risoluzione". L'ASA dichiarò che Oric non specificò che il Commodore 64 può rendere disponibili fino a 58K di memoria utilizzabile quando si usa il linguaggio macchina ed anche che quando si programma in BASIC il C64 permette di posizionare la grafica ad alta risoluzione sotto la ROM del sistema operativo, lasciando inalterata la memoria di base disponibile. E la memoria del C64 resta quindi sempre maggiore di quella disponibile sull'Atmos. Una portavoce di Oric, commentando la sentenza ASA, disse: "Questa campagna pubblicitaria non è più in corso e quindi tutto il discorso non è più rilevante".

Il numero 8 di Oric Owner, giugno/luglio 1984, portava buone notizie: sarebbe stato pubblicato mensilmente nelle edicole a partire dal numero di ottobre. Segno che si andava verso una stabilizzazione della base di utenti e di lettori. La produzione di Atmos aumentò fino a 10.000 unità al mese e i drive Microdisc erano ora disponibili al prezzo

riveduto di 299 sterline. E "What Micro?" nell'agosto 1984 annunciò finalmente che il modem era disponibile e che, addirittura, una stampante a 80 colonne sarebbe stata messa in commercio.

In Francia, la gamma di periferiche di terze parti era sorprendentemente ampia: sui cataloghi si potevano acquistare un sintetizzatore vocale, una scheda analogica a 8 linee, una scheda I/O a 16 vie ed una scheda di espansione con 3 slot. I drive Microdisc non costavano meno di 3.600 franchi, oltre 600 sterline di allora! L'edizione di Théoric del luglio 1984 riportava che nel 1983 Oric aveva raggiunto un fatturato di 25 milioni di sterline e ne prevedeva ben 45 per il 1984. La rivista esaminò a fondo il Microdisc e il produttore francese del drive concorrente "Jasmin", TRAN, prevedeva una produzione mensile compresa tra 700 e 1000 unità.

Tansoft, la software house di Oric, fu presentata sulla rivista Leisure Electronics Trader nell'agosto 1984. Le dichiarazioni del direttore Paul Kaufman furono riprese da varie riviste: "Tansoft raddoppierà presto la sua gamma di titoli una volta concluse le trattative con una software house francese. Tansoft importerà tutti i titoli dell'azienda nel Regno Unito[...] Con i giochi, il nostro modo di fare business è quello di sviluppare un concetto completamente innovativo[...] Il software gestionale e per le aziende sta generando molto interesse ora che i disk drive Oric sono in commercio. Una suite di programmi per la contabilità è già in fase di sviluppo e sarà lanciata a settembre".

E' da questo momento che possiamo tracciare l'inizio della fine. Oric Owner non fu pubblicato in agosto o a settembre. Neppure ad ottobre le edicole mostravano traccia della rivista. Probabilmente il primo segnale di problematiche serie in corso fu un articolo di Personal Computer Newsweek apparso il 4 agosto 1984. Con il titolo "La richiesta di Pan fa tremare Oric", il pezzo raccontò che Pan Books, che vantava un credito di £120.000 verso Oric per la produzione del manuale Atmos, era intenzionata a richiedere un'istanza di fallimento se il pagamento non fosse arrivato entro pochi giorni. I responsabili di Pan Books avevano incontrato Allan Castle due settimane prima ed avevano ottenuto come risposta che Oric doveva 2 milioni di

sterline a 12 grandi fornitori. PCN interpellò allora Castle: "La posizione debitoria è sostanzialmente corretta. Abbiamo debiti di circa 2,5 milioni di sterline verso 25 fornitori. Stiamo immettendo molto denaro nella produzione dei computer in vista della corsa per il mercato di Natale e questo periodo dell'anno di solito si presenta difficile per quel che riguarda il flusso di cassa. Pan Books è stata così gentile da permetterci di avere più credito di quanto ci si aspetterebbe normalmente e Oric li pagherà a breve. La nostra attuale posizione non è affatto insolita fra le imprese che producono microcomputer". PCN diede seguito alla storia il successivo 25 agosto: "Nella scorsa settimana i debiti di Oric sono stati stimati in circa 4 milioni di sterline ed è stato addirittura suggerito che l'azienda potesse uscire dal mercato britannico. Oric ha cercato di aiutare a superare l'impasse attuale con il taglio di 15 dipendenti presso la sua sede centrale di Ascot. Per placare le voci ed i timori sulla sua liquidità, Oric ha raccolto ordini per 2,75 milioni di dollari per la versione tedesca dell'Atmos. Saranno consegnati in Germania, Austria e Svizzera, con 2 milioni di dollari di ordini che arrivano da due grandi catene di vendita al dettaglio tedesche. L'azienda è ancora sul punto di concedere sconti alle principali catene di vendita al dettaglio. Solo poche settimane fa Oric aveva aumentato il prezzo dell'Atmos di 20 sterline, ma i prezzi nei negozi sono ancora intorno alle 150 sterline piuttosto che 190 sterline come Oric vorrebbe".

Inoltre, il successivo 8 settembre: "I principali creditori di Oric Products, affamati di liquidità, questa settimana dovevano riunirsi per verificare se l'azienda fosse in grado di superare il periodo di difficoltà. Gli attuali problemi di liquidità sono peggiorati la scorsa settimana quando Oric è andata in tribunale per rispondere ad un'ingiunzione della KMP, la sua agenzia pubblicitaria, che reclamava 200.000 sterline di fatture non pagate. Il tribunale ha respinto una controrichiesta di Oric che sosteneva che KMP avesse fornito una consulenza sbagliata ed ha condannato l'azienda a pagare l'importo dovuto insieme alle spese legali di KMP. Assembly and Automation Electronics, l'azienda che produce l'Atmos nello stabilimento M3, ha invece espresso comprensione per le difficoltà di Oric. Parteciperà alle discussioni anche

Hitachi, che fornisce unità disco e altri componenti a Oric. Un risultato probabile è che i creditori di Oric accetteranno di riprogrammare i debiti in modo che il saldo non avvenga fino a quando le vendite prima di Natale cominceranno a decollare". Ed infatti l'esito delle discussioni tenute dai creditori a metà settembre fu proprio quello auspicato. Un rinvio di qualche tempo per concedere ad Oric di incassare i proventi delle vendite natalizie. Quando il numero 9 di Oric Owner, ottobre/novembre 1984, arrivò in edicola e senza sorprese fu comunicato che i progetti di tornare in edicola regolarmente erano stati abbandonati. Invece fu lanciato l'Oric Club, insieme ad una rivista interna chiamata "Oricall". Nello stesso numero fu confermata la disponibilità di un modem a marchio Oric – soltanto 18 mesi dopo che era stato annunciato – e al prezzo di ben 100 sterline. Altri annunci da parte di Oric si susseguivano, fra cui quello che annunciava: "I computer Oric devono acquisire un numero sostanziale di pagine su Prestel per creare un vero database di software dedicato Oric". Nella realtà, naturalmente, niente del genere vide mai la luce. Per quanto se ne sa, non si è mai lavorato a nient'altro se non un numero fittizio di Oricall.

Per avere un'idea del mercato di allora, è interessante confrontare i prezzi di vendita dei concorrenti diretti di Oric a Natale del 1984 in UK: Oric Atmos £179, Sinclair Spectrum 48k £129, Acorn Electron £199, Commodore Vic 20 £129 e Amstrad CPC464 £349.

I francesi nel frattempo mettevano il fieno in cascina finché il sole splendeva. Nell'ottobre 1984 la software house d'oltralpe "No Man's Land" rilasciò tutti in una volta 23 titoli di software Oric nel Regno Unito. E tutti gli appassionati Oric inglesi capirono che cosa si erano persi sino ad allora. Per non essere schiacciati, i rivenditori Butex di Cardiff pubblicizzarono i loro prodotti Oric sulle riviste francesi – con il prezzo inglese. E c'erano nuove periferiche disponibili: il joystick programmabile Protek, le interfacce joystick dual port DK'Tronic, tutte messe in vendita da ottobre.

Ma gli eventi che accadevano in Francia confermavano ormai i primi segnali di difficoltà. Il capo degli importatori per la

Francia ASN., Denis Taieb, si dimise il 1° ottobre. Il giornale francese Hebdogiciel lo prese in castagna: "Rimango con Oric International come consulente. Ma tutto questo è influenzato solo dalle difficoltà di Oric UK di questo periodo. Hanno problemi finanziari con Pan Books, gli editori del manuale Atmos. Hanno creato una tempesta in un bicchier d'acqua, ma Oric ha ora concordato una moratoria sui debiti con loro.[...] Lo scorso aprile i nostri rivenditori in Francia hanno rifiutato di effettuare il pagamento dei loro ordini (come richiesto a noi stessi da Oric UK) e hanno insistito per ottenere termini di credito mensili. Abbiamo imposto le stesse condizioni a Oric UK e nei fatti è da quel momento che hanno avuto difficoltà. [...] Non sono riuscito a conciliare le visioni di Oric France e Oric UK". Hebdogiciel ha poi rivolto la sua attenzione al nuovo capo, Claude Taieb (il fratello di Denis!): "Oric International ha negoziato l'assistenza del governo britannico, quindi le sue finanze sono tornate alla normalità. Sei mesi fa Amstrad ci ha chiesto di distribuire la loro macchina, ma noi abbiamo rifiutato perché eravamo fedeli all'Oric, perché crediamo nella macchina [...]. Spero di organizzare una nuova forma di collaborazione con l'Oric UK - quando cambiano le personalità, cambiano anche le politiche che si possono attuare".

E i cambiamenti erano davvero nell'aria. Muncaster e Johnson comprarono la partecipazione di maggioranza in Tansoft e Paul Kaufman e Cathie Burrell se ne andarono subito per fondare la software house Orpheus con Geoffrey Guy e Geoff Phillips. Tansoft era ora guidata da Adrian Rushmore, il suo ex direttore marketing. "Stiamo concentrando tutto il nostro lavoro ed il nostro supporto sull'Atmos", dichiarò appena assunta la sua nuova carica. Il 15 novembre 1984, Personal Computer World rivelò i nuovi piani di Oric: lanciare tre nuovi computer nella primavera del 1985 tra cui lo Stratos / IQ164! Dovevano essere i successori dell'Atmos. Erano previsti il nuovo modello Stratos ad un prezzo intorno alle 250 sterline, un desktop IBM compatibile 8086 a partire da 400 sterline (e con interfaccia per cassette!!!) ed un portatile compatibile 8086 da 3.000 sterline. Questi primi cloni del PC IBM dovevano essere prodotti in USA ma marchiati Oric.



Figura 6 - Atmos e periferiche originali

L'Oric Stratos / IQ164, inizialmente progettato come aggiornamento dell'Atmos, non fu mai messo in produzione o venduto sul mercato, ma quando Euréka acquistò la Oric nel 1985, il progetto Stratos ispirò i nuovi designer per la produzione di un computer chiamato Telestrat, lanciato nel 1985 e compatibile con Oric-1 e Atmos. Il Telestrat fu progettato più come un terminale Minitel (la versione francese del sistema videotext noto in Italia col nome di Videotel).

Il numero 10 di Oric Owner, dicembre/gennaio 1985, fu come sempre ottimista. Il nuovo caporedattore era Carolyn Grunewald, ex direttrice pubblicitaria. L'Oric Stratos/IQ164 fu annunciato ai lettori e Bruce Everiss fu nominato amministratore delegato di Tansoft. La sua prima dichiarazione fu in un certo senso memorabile: "Il mio primo obiettivo è quello di collocare l'Oric Atmos nella sua giusta posizione di mercato". Altrettanto memorabile fu la risposta di Paul Kaufman alla notizia: "La sua reputazione dice tutto. L'unica cosa che mi infastidisce della sua nomina ad amministratore delegato è che ora sta guidando quella che una volta era la mia Mercedes".



Figura 7 - Oric Telestrat

Questo fu anche l'ultimo numero di Oric Owner - una rivista utile, anche se non particolarmente straordinaria. I 5 numeri di Oric Computing rappresentano l'unica rivista inglese per Oric prodotta in maniera davvero professionale. Nessun editore commerciale, visti i numeri, ha mai pubblicato una rivista specifica per le macchine Oric, anche se, a dire il vero, le riviste di giochi coprivano

abbastanza bene i titoli che uscivano per l'Oric-1 e l'Atmos.

In Francia, nel frattempo, la rivista Théoric andava sempre più forte, diventando mensile e vendendo molte migliaia di copie al mese in edicola. Si trattava di una rivista ricca di articoli utili e di ottimi programmi. Chiudiamo il 1984 con un'altra citazione di Paul Kaufman: "Se Oric sopravvive al Natale, allora sopravviverà al 1985". I problemi di Oric vennero a galla all'inizio di gennaio 1985. La sede centrale di Ascot fu chiusa e il personale ausiliario fu licenziato. Il futuro dello stabilimento inglese dipendeva dalle vendite della prima metà del 1985. Oric stava negoziando in Francia per un impianto di produzione a Longwy, vicino al confine con il Lussemburgo; secondo Lorraine Horne, allora responsabile delle operazioni di Oric, lo stabilimento sarebbe stato operativo entro la metà dell'anno e avrebbe prodotto le macchine Oric per il mercato francese e per il resto d'Europa. Il governo francese avrebbe supportato finanziariamente l'operazione con sovvenzioni di circa 150.000 sterline e prestiti a tasso agevolato. Oric e ASN investivano 500.000 sterline ciascuno. E Bruce Everiss fu molto sincero per una volta (in una rivista di settore): "La performance di Oric nel Regno Unito quest'anno è stata un disastro totale. L'azienda ha accumulato debiti enormi e deve rientrare di 3,5 milioni di sterline nei confronti di molti creditori entro marzo".

La migliore espressione di ottimismo sposata al tempismo, tuttavia, doveva ancora arrivare. In Your Computer di gennaio 1985, apparve una lunga intervista a Barry Muncaster e a Bruce Everiss, accompagnata da molte immagini significative. C'era una pagina doppia che valeva la pena di incorniciare a futura memoria e certamente alcune risposte dei due manager che vale la pena ancor oggi di citare: "Ai blocchi di partenza per Oric quest'anno ci sono cinque nuovi computer. Si va dallo Stratos, una macchina da 200 sterline basata sull'Atmos, ad un micro affiliato allo standard MSX, poi un computer in stile QL basato su 68008, quindi un desktop IBM compatibile e infine, al top della gamma, un laptop compatibile 8086. Anche se il 1984 è stato l'anno in cui Oric è quasi scomparsa dal mercato britannico degli home computer, la nostra fiducia per il 1985 si basa su due risorse: un boom delle vendite in

Europa e l'esperienza maturata nella progettazione dei chip ULA[...]. 350.000 unità sono state vendute nei primi due anni [...]. Lo Stratos sarà in vendita in Francia a partire da febbraio".

Con dichiarazioni del genere, la fine non poteva che essere vicina. E fu al Salone del Computer di Francoforte il 1° febbraio 1985 che venne lanciata l'Oric Stratos – una macchina che ormai poteva dirsi solo francese - completa di porte per cartucce, BASIC esteso ed una serie di altre eccellenti caratteristiche. L'inevitabile colpo finale non tardò ad arrivare. Giovedì 2 febbraio 1985 la Edenspring mise in amministrazione controllata la Oric. Oric aveva accumulato debiti non inferiori a 5,5 milioni di sterline contro un patrimonio stimato di soli 3 milioni di sterline.

Va ricordato che in quello stesso periodo, le cose andavano male anche per altri produttori di home computer del Regno Unito e per alcuni di questi si era raggiunto il punto di non ritorno. Solo per citarne un paio, la Computers ed il loro Lynx e la Dragon Data con il Dragon 32 erano già stati infilzati, altre aziende boccheggiavano ovunque (ad es. Jupiter Cantab e il loro Ace, la Grundy, la Memotech) e, cosa più preoccupante, i grandi attori del mercato, Acorn e Sinclair, navigavano già in grossi guai finanziari, soprattutto a causa di scelte poco lungimiranti.

*Un ringraziamento speciale va ad **Andy "8BitBaz" Barr** per avermi fornito documenti, articoli e altre preziose risorse informative (oltre che due splendidi esemplari di Oric-1 e Oric Atmos) e a **Cristiano Bey** per avermi fatto scoprire, con la sua passione per l'Oric, questa piccola perla fra gli home computer degli anni Ottanta.*

Riferimenti e fonti

La rivista francese Théoric - [Link](#)
 Il gioco "L'Aigle D'Or" - [Link](#)
 The Oric Story – <http://oric.free.fr>
 Gli oltre 30 anni di Oric - [Link](#)
 Oric FAQ – [Link](#)
 Il gioco "Space: 1999" - [Link](#)
 Computers Lynx Computer - [Link](#)
 Dragon Data Dragon 32 - [Link](#)

ORIC Atmos in breve

Modello: Oric Atmos

Produttore: Oric – Regno Unito

Commercializzazione: febbraio 1984

Processore: CPU 6502A 1 MHz

ROM: v1.1 - 16 KB

RAM: 48 KB

Linguaggio: Tangerine Basic (Microsoft)

Tastiera: QWERTY, tastiera meccanica, 58 tasti, 4 tasti cursore

Modo testo: 40 caratteri x 28 linee, 8 colori, caratteri 6x8 pixel

Modo grafico: 240 x 200 pixel + 3 linee testo

Suono: AY-3-8912, 3 voci, 8 ottave

Porte: Alimentazione, porta di espansione, porta stampante/Centronics, porta DIN per registratore, uscita video RGB e RF TV

Alimentatore: PSU esterno 9V 600 mA

Prezzo iniziale UK: £169.95

ORIC Stratos / IQ 164 in breve

Modello: Oric Stratos / IQ 164

Produttore: Oric – Regno Unito

Commercializzazione: (mai messo in vendita)

Processore: CPU 6502A 1 MHz

ROM: 48KB, fino a 112KB con 2 cartucce

RAM: 64 KB – 37486 bytes free

Linguaggio: Oric Super Extended Basic

Tastiera: QWERTY, tastiera meccanica, 58 tasti, 4 tasti cursore

Modo testo: 40 caratteri x 28 linee, 8 colori, caratteri 6x8 pixel (modo 80x26 progettato)

Modo grafico: 240 x 200 pixel + 3 linee testo, modo alta risoluzione a 16 colori progettato

Suono: AY-3-8912, 3 voci, 8 ottave

Disk drive: 3" (400KB) incorporato

Porte: Alimentazione, 2 porte per cartucce, 2 connettori joystick, porta Shugart disk drive, connettore RS232c, porta

stampante/Centronics, porta di espansione, porta DIN per modem, porta DIN per registratore, uscita video RGB e RF TV

Alimentatore: PSU esterno 9V 600 mA

Prezzo iniziale UK: £249.95 (previsto)

RetroPie Utilities

di Giorgio Balestrieri

In una rivista che si occupa di retrotecnologia e retrogaming, non poteva mancare un articolo su RetroPie. Per quanto strano possa sembrare, scrivere qualcosa di originale su questo sistema non è semplice; la popolarità del software è tale che ormai sul web si trova una quantità enorme di materiale a riguardo il che, per quanto rappresenti un vantaggio per chi utilizza questa distribuzione, non rende facile il compito a chi volesse scrivere qualcosa a riguardo senza ripetere o riportare concetti ed idee già noti ed ampiamente discussi.

Per questo motivo, abbiamo deciso di presentare qui una suite di piccole utility da utilizzare con RetroPie che forniscono alcune utili funzionalità per semplificare il controllo del sistema, soprattutto a chi volesse utilizzarlo per creare sistemi bartop o simili da utilizzare in modalità kiosk, senza collegarlo ad una tastiera o ad una rete e limitando al massimo l'interazione utente con il sistema sottostante.

RetroPie

Con più che ottime probabilità, chiunque stia leggendo questo articolo ha usato o comunque conosce RetroPie, perciò sarebbe inutile darne qui una descrizione dettagliata. Per completezza dell'articolo e per quei (davvero) pochi lettori che non ne abbiano mai sentito parlare, riassumeremo in poche righe cos'è RetroPie e per cosa viene utilizzato. La distribuzione Linux RetroPie, in circolazione già da un po' di anni, è un software dedicato alla gestione ed all'esecuzione del maggior numero di emulatori di piattaforme di retrogaming possibile. Nonostante sia nato originariamente come distribuzione Linux per Raspberry (v. riquadro 1), forma in cui è tuttora maggiormente utilizzato, oggi è disponibile anche per altri dispositivi hardware come gli Orange Pi o i Banana Pi e persino come applicativo da utilizzare su una installazione Linux pre-esistente su PC tradizionali. Il sistema si compone sostanzialmente di due parti, una che si occupa di gestire e configurare nel modo più uniforme possibile gli emulatori ed una che ne rappresenta l'interfaccia utente, dedicata al lancio dei giochi ed alla configurazione (non completa) del sistema utilizzando un joystick collegato alla macchina. L'interfaccia utente per antonomasia di RetroPie è EmulationStation, ma con un'adeguata conoscenza del sistema è possibile utilizzarne altre, comprese interfacce prodotte in proprio, a seconda dei gusti e delle esigenze dell'utilizzatore.

Uno dei punti di forza di RetroPie è proprio l'elevato grado di personalizzazione che offre; una volta che si accumula abbastanza esperienza per smanettarci su, è possibile eseguire configurazioni ed hacking di notevole livello.

Per contro, proprio la sua versatilità può rivelarsi una debolezza: addentrarsi nei dettagli del sistema non è lavoro per chi è a digiuno di nozioni sistemiche e di un certo grado di esperienza con

Linux. Inoltre, non tutte le configurazioni possono essere compiute tramite le interfacce di EmulationStation, l'uniformità della gestione degli emulatori non è completa al 100% e particolari modifiche o personalizzazioni richiedono una conoscenza piuttosto approfondita di RetroPie e di Debian/Raspbian/Armbian Linux per essere portate a termine.

Ciò detto, è anche vero che se avete bisogno di un controllo totale del sistema per un vostro particolare progetto ed avete le conoscenze adeguate, è più probabile che, invece di usare un sistema come RetroPie, preferiate procedere "from scratch", partendo da una distribuzione Linux "vergine" (o anche Windows, non siamo razzisti qui), installando e configurando a mano quel che occorre.

Se invece volete creare una vostra macchina da gioco multemulatore, magari da inserire in un cabinato o box qualsivoglia autocostruito e non volete impazzire troppo con i vari aspetti coinvolti in questo tipo di compito, RetroPie è un ottimo software da prendere in considerazione. Di fatto, esistono moltissimi progetti creati da persone che hanno realizzato la console dei loro sogni basandosi su questa distribuzione, costruendo da soli anche il box contenitore per il Raspberry secondo i loro gusti personali.

In vendita è anche possibile trovare box già belli e pronti in cui infilare un microcomputer SoC con RetroPie installato, a cui collegare pulsanti e comandi per avere con poca fatica un cabinato con cui far colpo sugli amici (un po' meno su mogli/mariti e compagne/i...) o per realizzare console di stampo classico, anche portatili, senza dover armeggiare con CAD e seghetti per costruirne una da zero.

Esistono anche diversi negozi, on-line e tradizionali, che vendono cabinati e console basate su RetroPie che includono già un Raspberry ed una schedina SD con RetroPie installato e zeppa di giochi pronti all'uso. Se, nonostante i prezzi mediamente alti (in alcuni casi scandalosi), questa vi sembrasse una soluzione comoda per mettere le mani su un sistema immediatamente fruibile, ricordiamo che questo tipo di sistemi sono completamente illegali ed il loro commercio rappresenta un reato, senza se e senza ma.

Oltre all'ovvio problema di distribuire, anche in forma gratuita, rom di giochi di cui non si posseggono i diritti/licenze (dimenticate la questione dell'uso per 24 ore prima di cancellarle, è una totale sciocchezza), la licenza d'uso di RetroPie vieta espressamente di distribuire, foss'anche a titolo gratuito, il sistema già installato su una schedina SD, né inserita in un hardware predisposto, né fornendola come dotazione abbinata, né distribuendola a parte come componente singola. L'unica possibilità legalmente concessa, è quella che il possessore dell'hardware

scarichi il software e lo installi personalmente su una schedina SD o su un sistema Linux e provveda ad inserirlo nell'hardware su cui intende utilizzarlo. Qualunque altra forma è vietata e costituisce reato.

Riquadro 1

Sistemi SoC Based

Con questo termine ci si riferisce ad una nutrita gamma di hardware incentrata sull'utilizzo di chip SoC (system-on-a-chip), ossia chip che integrano oltre ad un microprocessore, anche la circuiteria che di solito è alloggiata su una scheda madre, come controller per la ram, video, ethernet usb e così via. Non tutti i SoC sono uguali e le funzionalità che integrano possono variare anche di molto, ma permettono tutti di creare sistemi completi in spazi molto ridotti a costi contenuti. Vengono impiegati in sistemi embedded, come negli smartphone, dove i vincoli di spazio e costi sono di primaria importanza, ma grazie alla potenza di calcolo che hanno raggiunto ed al numero di funzionalità integrate, sono ormai impiegati anche nella creazione di microcomputer general purpose perfettamente funzionanti a scopo principalmente hobbistico. Il capostipite di questi microcomputer è il Raspberry, il cui modello di punta, il Raspberry 3B+, è dotato di un processore a 4 core, 1GB di ram, 4 porte USB, una ethernet, una interfaccia wifi ed una scheda video con uscita HDMI. Ad oggi è ancora il sistema più economico sul mercato ma prodotti concorrenti (più costosi), come l'Orange Pi 2+ (il nostro preferito) o i modelli Banana Pi offrono dotazioni hardware ben più estese e prestazioni nettamente maggiori degli equivalenti Raspberry che ne permettono l'impiego anche in ambiti professionali. Un tratto comune a tutti questi dispositivi è la presenza di una interfaccia GPIO, a cui è possibile collegare una enorme quantità di dispositivi che estende grandemente la versatilità di questi sistemi.

RetroPie Utility

Una volta installato e configurato, RetroPie consente la gestione ordinaria del sistema tramite l'uso del solo joystick, permettendo di eseguire azioni come la scelta di un emulatore, il lancio dei giochi e la loro chiusura, lo spegnimento o il riavvio del sistema e la cattura di una schermata del gioco in esecuzione. A patto di sapere cosa si sta facendo, sempre tramite joystick è possibile agire su una grande quantità di parametri di configurazione e personalizzazione del sistema, più che sufficienti per la maggior parte degli usi e degli utenti.

Nonostante ciò, esistono dei casi particolari in cui un minimo di "smanettamento" è necessario per coprire quei sistemi (emulatori) il cui utilizzo non viene reso omogeneo da RetroPie o, con un impegno decisamente maggiore, per portare

all'estremo la personalizzazione del sistema, come nel caso della creazione di un tema ex novo, caratteristica questa certamente interessante e non priva di fascino, di cui però non ci occuperemo in questo articolo dove ci concentreremo su altri aspetti, come lo spegnimento ed il riavvio del sistema senza l'utilizzo del menù di Emulation Station, la cattura di una schermata e dell'uscita dai giochi, funzionalità queste ultime due curiosamente non implementate dal sistema con sufficiente integrazione per essere utilizzate con le stesse modalità in tutti gli emulatori. L'operazione di chiusura di un emulatore ad esempio, è normalmente affidata alla pressione simultanea sul joystick dei tasti **hotkey + start**, dove **hotkey** è il pulsante scelto al primo avvio di RetroPie nella fase di configurazione del controller. Purtroppo però questo metodo funziona solo con gli emulatori basati su retroarch; gli altri debbono essere chiusi secondo modalità che variano da emulatore ad emulatore, oppure è necessario modificare a mano la configurazione (quando possibile) dei sistemi che non rispondono alla combinazione **hotkey + start** per far sì che la riconoscano. Lo stesso discorso vale per la cattura di una schermata di gioco: per gli emulatori non retroarch il sistema varia a seconda del software utilizzato. Inoltre, benché la gestione di tastiera, mouse e (ovviamente) joystick usb sia perfettamente integrata nel sistema, lo stesso non si può dire per la gestione della GPIO e dei dispositivi collegati a questa interfaccia. Anche se nella stragrande maggioranza degli scenari il mancato supporto per la GPIO non costituisce un problema, il suo impiego potrebbe far comodo nel caso in cui si voglia utilizzare il raspberry in un box personalizzato o autocostruito, come per nei progetti di console "custom made" e bartop o case arcade da usare in ambito domestico o pubblico (attenzione alle licenze dei giochi!), ad esempio nella vostra associazione di appassionati di retrogaming, magari tirata su seguendo i consigli di Starfox Mulder nel numero 8 di RetroMagazine. In questi casi, probabilmente azioni come lo spegnimento/riavvio del sistema o lo "scatto" di una schermata di gioco si preferirà vietarle per evitare che un utente troppo intraprendente crei problemi al sistema (ed agli altri aspiranti giocatori), oppure si vorrà optare per sistemi più comodi dell'uso della voce del menù utente per spegnere il Raspberry, che tra l'altro è disponibile solo in Emulation Station. Dal momento che il case in cui verrà inserito l'hardware verrà costruito (o anche solo modificato) ad hoc, si potrebbe ad esempio aggiungere un bottone per lo spegnimento ed uno per il riavvio. Grazie alla GPIO, opzioni di questo tipo sono molto semplici da implementare e da gestire su un Raspberry, ma bisogna poi saper integrare il tutto in RetroPie. In tutti questi scenari, le RetroPie Utility vengono in aiuto a chi avesse una o più di queste necessità, sollevandolo dal compito di dover scrivere da zero il software e di integrarlo nel sistema. Le utility si compongono di una serie di eseguibili, sia binari che script shell, tutti progettati e programmati dall'autore di questo articolo tranne uno, **raspizpng**, creato da Andrew Duncan ed utilizzato per salvare il contenuto del framebuffer in un file. Con esse è possibile monitorare la temperatura del sistema, comandare lo spegnimento, il riavvio o il salvataggio di quanto mostrato a video tramite interruttori collegati alla GPIO, la chiusura dell'emulatore in esecuzione (qualunque emulatore) ed il salvataggio di una

schermata, tramite la pressione di un tasto (o combinazione di) sul joystick, uniformando così questa operazione per tutti i software di emulazione disponibili in RetroPie, da quelli basati su retroarch agli emulatori di laser disc (Daphne), avventure grafiche (ScummVM), PC DOS (DOSBox) ecc.

A differenza di quanto normalmente avviene nella creazione di utility per Raspberry, in particolare quelle che utilizzano la GPIO, per questo set di programmi, ad eccezione di **raspizpng** scritto in C, è stato scelto il **FreeBasic** come linguaggio di programmazione al posto del classico Python, principalmente per tre ragioni. La prima, è che un eseguibile generato da **FreeBasic** occupa molta meno memoria quando è in esecuzione ed è capace di girare a velocità enormemente superiori rispetto al Python, grazie al fatto che non necessita di una macchina virtuale (interprete) per essere eseguito, poiché il codice viene compilato in formato ELF per ARM ed eseguito nativamente dalla distribuzione Linux su cui RetroPie si basa. La seconda è che qui a RetroMagazine siamo molto affezionati al Basic, per ragioni puramente nostalgiche, e se c'è l'occasione di utilizzarlo o di parlarne, non ce la facciamo assolutamente sfuggire. Il terzo motivo è puramente personale: come linguaggio, il Python non è tra quelli più amati dall'autore, anzi, che ha immediatamente colto al volo la possibilità di NON utilizzarlo per questo progetto. In più, considerando che questa scelta avrebbe scatenato le più vive proteste da parte di Leonardo Giordani, vero alfieri del Python, dovevamo per forza scartare questo linguaggio di programmazione: una simile occasione per stuzzicarlo non poteva essere persa. Motivazioni nostalgiche e personali a parte, la possibilità di risparmiare memoria e tempo di calcolo su un sistema non proprio dotatissimo come il Raspberry non poteva essere trascurata ed è di fatto questo il vero motivo dell'impiego di **FreeBasic**. *(Occhio Giorgio che gli alfieri si muovono in diagonale e te li trovi alle spalle quando meno te lo aspetti. Comunque è chiaro, siamo su una rivista di retrotecnologie, per cui dimentichiamoci pure che il mondo è andato avanti. Come minimo mi aspetto che tu tiri fuori dalla cantina qualche dinosauro tipo il BASIC. NdLeo) (I dinosauri hanno dalla loro la velocità e la potenza e gli alfieri nascono sfigati già di loro visto che possono controllare solo metà della scacchiera... NdGiorgio).*

Prima di passare a descrivere in dettaglio l'uso delle RetroPie Utility, vogliamo parlarvi un po' del **FreeBasic**, certi del fatto che potrebbe tornare utile anche a quei lettori che volessero sviluppare proprie utility o in generale software più efficiente di quello ottenibile con un linguaggio interpretato e che non costringa all'utilizzo del linguaggio C, certamente il migliore dal punto di vista prestazionale ma che richiede al programmatore maggiori abilità ed è caratterizzato da una curva di apprendimento più impegnativa.

FreeBasic

Creato nel 2004 con l'obiettivo di far rivivere il glorioso QuickBasic, il **FreeBasic** nel tempo si è arricchito di un gran numero di funzionalità, come la capacità di utilizzare la programmazione ad oggetti, che gli hanno fatto perdere la compatibilità con il Basic di Microsoft, a vantaggio di forme di

sviluppo più moderne. *(Ecco appunto. Si parlava giusto un attimo fa di dinosauri. NdLeo).*

Disponibile per sistemi DOS, Windows e Linux, il **FreeBasic** si compone di una serie di tool per la creazione di programmi scritti in un dialetto Basic (moderatamente) orientato agli oggetti e capace di interfacciarsi con le librerie di sistema Linux e Windows. Gli eseguibili prodotti possono girare sia in una shell a caratteri, come il prompt dei comandi di Windows o la shell di Linux, che in un ambiente a finestre. L'installazione di default prevede il supporto per una nutrita serie di librerie, che spaziano dalla gestione grafica, OpenGL e SDL comprese, al supporto dei database o del multithreading "vero", non emulato come capita in alcuni linguaggi di programmazione che vanno per la maggiore (non ci riferiamo al Python in particolare, ma esso è comunque un ottimo esempio). *(I thread sono come le tasse, tutti promettono di gestirli in maniera migliore e la gente va in visibilo. Poi la verità è che la gente non capisce a cosa servano e il nuovo governo non fa nulla di diverso dal precedente. NdLeo).*

Inoltre, il formato dei binari generati e dei tipi primitivi del linguaggio sono compatibili con quelli del C, il che porta come gradita conseguenza la possibilità di utilizzare qualsiasi libreria C, in primis quelle di sistema, semplicemente convertendo (o prendendoli già tradotti) i file header ".h" in ".bi". Questo dona al **FreeBasic** un'enorme versatilità, rendendo al programmatore la possibilità di scrivere una grande varietà di software che, con le dovute accortezze, è anche discretamente portabile. La semplicità del linguaggio lo rende comodo da usare in tutte quelle situazioni in cui si utilizza un linguaggio di scripting, come ad esempio l'automazione e la gestione di compiti amministrativi, middleware di integrazione di sistemi o software multipiattaforma, con in più il vantaggio di ottenere eseguibili in grado di girare con velocità notevolmente superiori a qualunque linguaggio interpretato e con un minor utilizzo della memoria. Guardando al rovescio della medaglia, i punti deboli di questo linguaggio sono un supporto alla programmazione ad oggetti non sufficiente per gli standard odierni, una sintassi troppo legata allo stile degli anni '90 (che per molti retroappassionati probabilmente è un pregio) e la mancanza di un'IDE di sviluppo paragonabile agli ambienti Visual di Microsoft o anche NetBeans/Eclipse di Oracle. Seppur esista qualche tool di supporto, specie per lo sviluppo in ambienti a finestre, l'unico modo pratico di produrre software in **FreeBasic** è l'adozione di un editor di testi per programmatori (Sublime Text è il nostro preferito). Per chi volesse utilizzarlo solo sotto Microsoft Windows, può avvalersi di FBIDE, un ambiente di sviluppo integrato scritto in **FreeBasic** stesso che, seppur non possa competere con gli IDE di Microsoft o di Oracle, può comunque semplificare lo sviluppo in caso di progetti complessi. *(Ma i veri programmatori usano vi. O Emacs? NdLeo) (Eccerto e l'unico linguaggio vero è l'assembly, le interfacce a finestre sono il male e le mezze stagioni non ci sono più... NdGiorgio)*

Nonostante questi limiti, il **FreeBasic** è un ottimo linguaggio general purpose, particolarmente adatto per creare utility di gestione ed automazione di sistemi ed un eccellente strumento di sviluppo su sistemi SoC based come i Raspberry Pi, dove i

vantaggi sul fronte della velocità e consumo della memoria rispetto ad un linguaggio interpretato sono molto preziosi su questo tipo di piattaforma, senza trascurare la maggior semplicità di scrittura del codice rispetto al linguaggio C. In particolare, utilizzandolo insieme alla libreria **WiringPI**, creata in e per il C da Gordon Henderson, può essere impiegato con efficacia nella creazione di moduli di controllo dei dispositivi connessi alla GPIO per gestire pulsanti, sensori di varia natura ed in generale, di tutto l'hardware collegabile via GPIO. Anche se pensata per l'uso da linguaggio C, la **WiringPI** è perfettamente utilizzabile in **FreeBasic** semplicemente adottando un header ".bi" ad hoc convertito dal relativo ".h" del C. A questo proposito, facciamo notare che la conversione dell'header della WiringPi dal C al **FreeBasic** è un po' più complesso rispetto ad altri header ma, per la fortuna dei lettori, ne abbiamo trovato uno già tradotto e funzionante, che alleghiamo ai sorgenti delle RetroPie Utility. Di questa libreria possono beneficiare anche il Python o il Ruby, ma in questo caso è necessario adottare un opportuno wrapper che non è purtroppo disponibile per tutti i linguaggi.

Installazione delle RetroPie Utility

Dopo aver descritto in generale cosa sono e per cosa possono essere utilizzate le RetroPie Utility ed aver brevemente discusso del linguaggio con cui due di esse sono state create, possiamo finalmente vedere in dettaglio come si utilizzano e come possono aiutarci nella personalizzazione del nostro bartop perfetto, o perlomeno tale agli occhi di chi ha passato ore a modellarlo con i più svariati attrezzi da bricolage.

Prima di poter utilizzare le utility, dovremo ovviamente installarle. Per raggiungere questo scopo, dovremo lavorare con la console di RetroPie via ssh. Per gli utenti meno esperti questo può essere un po' complesso se non si ha un minimo di esperienza con il sistema e qualche conoscenza basilare di Linux, ma se siete giunti alla conclusione che queste utility vi sono necessarie, probabilmente avrete già gli "skill" necessari e sufficiente dimestichezza per procedere in tutta tranquillità. Ad ogni modo, tenteremo di essere il più chiari, possibile senza diventare eccessivamente prolissi.

Un prerequisito fondamentale, è la presenza della libreria **WiringPI** compilata a partire dai sorgenti ottenibili dal sito di Henderson. Per questa operazione, occorre prima rimuovere l'eventuale installazione pre-esistente prima di procedere con la compilazione; l'intero processo non è difficile ed è ben spiegato sulla home page della libreria. Un altro software che sarebbe bene installare, anche se non necessario al funzionamento delle RetroPie Utility ma molto utile se le si vuole ricompilare o sviluppare in proprio software per il Raspberry, è naturalmente il **FreeBasic**, facilmente ottenibile seguendo le istruzioni che trovate nei link indicati alla fine di questo articolo.

Per semplificare la vita ai nostri lettori però, abbiamo predisposto uno script bash che provvede ad installare entrambi i software in un colpo solo, eseguendo tutte le operazioni necessarie automaticamente. Assicuratevi di aver collegato il Raspberry alla rete (WiFi o ethernet, non importa) e segnate il suo IP (menù RetroPie->Show IP),

quindi collegatevi al dispositivo utilizzando un client ssh come "putty" sotto Windows o "ssh" sotto Linux. Se la connessione dovesse non riuscire, controllate che sul Raspberry sia attivato il servizio ssh (menù RetroPie->Raspi-Config->Interfacing Options->SSH) e che le credenziali per l'accesso siano corrette (di default sono username: pi e password: raspberry).

Una volta aperta la sessione ssh, procedete indicando in sequenza i comandi:

1. wget <http://www.retromagazine.net/download/rpu.zip>
2. unzip rpu.zip
3. cd rpu
4. ./install_fb_wiringpi

Se dopo aver lanciato l'ultimo comando della sequenza, il sistema dovesse rispondere con "permesso negato" (o l'equivalente in lingua inglese), inserite questi due comandi:

- chmod a+x ./install_fb_wiringpi
- chmod a+x ./install_rpu

lanciate di nuovo il comando al passo 4 e attendete la fine del processo. Poiché l'installazione prevede la compilazione sia di **FreeBasic** che della **WiringPI**, saranno necessari un po' di minuti prima che l'operazione si concluda (e se usate un Raspberry 2, un bel po' di minuti...). Se tutto è andato liscio, il sistema verrà riavviato e si potrà procedere con l'installazione delle RetroPie utility.

Una volta installati i prerequisiti software, collegatevi di nuovo in ssh al dispositivo e date i comandi:

1. cd rpu
2. ./install_rpu

Seguite le istruzioni a schermo per la configurazione dei tasti (o combinazione di) da associare alle azioni di chiusura di un emulatore o di salvataggio di una schermata. Quando verrà lanciato **joytest**, premete uno o più tasti sul vostro controller e segnate il numero che appare alla riga "Buttons", colonna "Dec". Ripetete l'operazione una seconda volta per scegliere un altro tasto o combinazione. Se avete più joypad, utilizzate i tasti "+" e "-" per cambiare controller. Ciò fatto, segnate anche l'id del joypad (indicato in alto, di default è zero) e premete "q" per chiudere **joytest**. Inserite nell'ordine:

1. L'id del controller da utilizzare.
2. Il codice tasti da assegnare alla chiusura di un emulatore.
3. Il codice tasti per salvare uno screenshot.

I passaggi 2 e 3 sono opzionali e possono essere saltati semplicemente premendo "invio", ma almeno uno andrebbe eseguito...

A questo punto la procedura continuerà ed il sistema verrà riavviato di nuovo. Quando Emulation Station sarà ripartito, lanciando un gioco si potrà chiuderlo semplicemente premendo la combinazione di tasti scelta, indipendentemente

dall'emulatore utilizzato (dosbox e scumm-vm compresi) ed in maniera simile sarà possibile catturare schermate.

Descrizione delle RetroPie Utility

Completando la procedura di installazione descritta nel precedente paragrafo, ci si ritroverà con i seguenti file nella cartella di sistema "/usr/bin":

- joyrun
- gpio_buttons
- gpio_readall
- joytest
- tempmon
- raspizpng

Di cui la prima, **joyrun**, viene integrata in RetroPie dallo script di installazione per comandare lo shutdown dell'emulatore ed il salvataggio delle schermate. La seconda, **gpio_buttons**, si occupa di gestire eventuali pulsanti collegati sull'interfaccia GPIO ed eseguire operazioni come lo shutdown o il riavvio del sistema alla pressione di un bottone ma la sua configurazione ed integrazione del sistema deve essere compiuta dall'utente. Le tre rimanenti sono utility "di servizio", da utilizzare alla bisogna (**raspizpng** viene sfruttata anche da **joyrun** e **gpio_buttons**). Passiamo ora alla descrizione completa di ogni utility, analizzando le modalità di configurazione e come integrarle in RetroPie (dove possibile).

joyrun

Come avrete ormai ben compreso, questa utility si occupa di controllare se su uno dei joypad collegati al Raspberry viene premuto un tasto o una combinazione di tasti a cui è stata associata un'azione, eseguendola in caso di un riscontro positivo.

Il suo comportamento viene definito da un file di configurazione, che deve trovarsi nella stessa cartella dell'eseguibile e può essere integrato in RetroPie tramite lo script shell **runcommand-onstart.sh** posto in "/opt/retropie/configs/all/" che RetroPie invoca prima del lancio di un emulatore. Lo script di installazione provvede a modificare **runcommand-onstart.sh** inserendovi i comandi necessari per agganciare **joyrun** all'emulatore lanciato, a copiare l'eseguibile dell'utility in "/usr/bin/", il suo file di configurazione "joyrun.conf" in "/home/pi/" ed a collegarlo in "/usr/bin/" tramite un link statico per renderlo accessibile e modificabile senza permessi di root, ma potete spostare eseguibile e file di configurazione dove preferite, a patto poi di modificare anche lo script di RetroPie.

Il file di configurazione "joyrun.conf" prevede la definizione di una serie di registri utilizzati da **joyrun** per il suo funzionamento:

```
joypad_id = <ID joypad>
```

Indica il numero identificativo del joypad da associare a **joyrun**. Può essere determinato utilizzando **joytest**, come indicato nel paragrafo dedicato all'installazione.

```
screenshots_path = </percorso/degli/screenshots>
```

Indica dove salvare gli screenshot. Se non viene definito, gli screenshot finiscono nella cartella dell'eseguibile, in questo caso `"/usr/bin"`, una pessima idea. Lo script di installazione crea la cartella `"/home/pi/screenshots"` e la associa a questo registro.

```
idle_timeout = <timeout definito in secondi>
```

Definisce un tempo di timeout espresso in secondi, trascorso il quale l'emulatore in esecuzione viene chiuso per fare ritorno ad Emulation Station. La pressione di un tasto sul controller resetta il timer. Questo registro è utile nel caso di un sistema da usare in pubblico, per non lasciare un gioco in esecuzione se non c'è nessun giocatore ai comandi, comportamento preferibile ad esempio nel caso di giochi con un attract mode particolarmente vivace o per non dare l'impressione che il vostro sistema faccia girare un solo gioco. Non può essere disabilitato, ma di default viene impostato a 1296000 secondi, ossia 15 giorni, il che equivale a non farlo scattare praticamente mai nella totalità degli scenari d'uso. In questa versione, il timeout viene resettato **solo** dalla pressione dei pulsanti, in future versioni probabilmente verrà considerata anche la pressione della croce direzionale o l'uso dei controlli analogici.

```
action_trigger = <codice tasti>, <azione>
```

Associa un'azione da compiere alla pressione di un tasto o di una combinazione di pulsanti. Utilizzando **joytest** si può determinare il valore di `<codice tasti>`, mentre `<azione>` può assumere uno dei seguenti valori:

- **take_screenshot**
scatta uno screenshot.
- **quit_emulator**
chiude l'emulatore.
- **take_screenshot_then_quit**
salva una schermata e chiude l'emulatore.

L'ultimo valore può tornare comodo nel caso in cui si voglia generare una serie di schermate da utilizzare nella personalizzazione di un tema di RetroPie, permettendo di accelerare il processo di raccolta degli screenshot.

E' possibile indicare più registri `action_trigger`, a seconda di quante operazioni vogliamo implementare. E' da notare che **joyrun** non obbliga in nessun modo a definire un'associazione tasti/azione, ma è ovviamente necessario fornirne almeno una o l'uso di questa utility non avrà alcun senso...

E' anche possibile inserire commenti nel file di configurazione, semplicemente indicando un diesis (`#`) come primo carattere di una riga. Oltre al file di configurazione, è possibile indicare alcuni parametri inserendoli sulla linea di comando di **joyrun** in questo ordine:

```
joyrun <nome processo> [<timeout> <codice tasti> [<joystick id>]]
```

questa modalità non è però consigliata perché ne riduce grandemente le possibilità d'uso. Di fatto, è utile solo in particolari configurazioni, ad esempio in

combinazione con un'altra utility o script bash per agganciarlo ad un processo in esecuzione, come nel caso di **runcommand-onstart.sh**. I parametri di configurazione passati sulla linea di comando hanno la precedenza su quelli definiti nel file di configurazione.

gpio_buttons

Come abbiamo già spiegato in precedenza, questa utility permette di associare azioni alla pressione di pulsanti collegati al GPIO. Come per **joyrun**, il suo comportamento è definito da un file di configurazione, `"gpio_buttons.conf"`, che deve trovarsi (o essere collegato via link simbolico) nella stessa cartella dell'eseguibile (di default `"/usr/bin"`).

I registri definibili sono:

```
screenshots_path = </percorso/degli/screenshots>
```

Definisce il percorso in cui salvare gli screenshot. Se viene omesso, verrà utilizzata la stessa cartella dell'eseguibile.

```
assign_pin = <id pin>, <azione>, <tempo di pressione>
```

Assegna ad un pulsante un'azione da compiere. Il valore di `<id pin>` può essere ricavato utilizzando l'utility **gpio_readall**. Il parametro `<azione>` può valere:

- **take_screenshot**
salva la schermata corrente.
- **shutdown**
esegue lo shutdown del sistema.
- **reboot**
esegue il riavvio del sistema.

L'ultimo parametro, `<tempo di pressione>`, permette di definire un ritardo, espresso in secondi, prima di eseguire l'azione associata ad un tasto. Questo è utile per evitare di dare inizio alla procedura di un'azione se un tasto viene premuto per errore. Ritrovarsi con la macchina in riavvio mentre si sta segnando il proprio record personale a Donkey Kong solo perché un amico di passaggio ha premuto per sbaglio il tasto di reset non è bello...

A differenza di **joyrun**, **gpio_buttons** non viene installato automaticamente nel sistema, per il semplice motivo che potrebbe non essere necessario all'utente. Un joypad è necessario per il funzionamento di RetroPie ed è scontato che chi ha un sistema basato su questa distribuzione abbia un controller collegato al RetroPie, ma collegare pulsanti alla GPIO è un'operazione che viene compiuta solo in casi particolari. Per contro, integrazione nel sistema è più semplice rispetto a **joyrun**: basta lanciarlo come demone in background e dimenticarsene. Ci sono diversi metodi da adottare per questo scopo, come l'utilizzo di **systemd** o del **crontab** opportunamente configurati per lanciare **gpio_buttons** all'avvio di RetroPie. Agli utenti meno esperti, consigliamo l'uso del **crontab**, che è meno efficiente rispetto al **systemd** ma decisamente più semplice da usare di quest'ultimo. Per indicare al **crontab** di lanciare il gestore dei pulsanti su GPIO al boot, si può procedere come segue:

- Collegarsi al raspberry via client ssh
- Digitare: **crontab -e** (premere invio)
- Se è la prima volta che si lancia **crontab**, verrà chiesto quale editor usare. Premete invio per utilizzare quello suggerito per default (nano)
- Spostare il cursore in fondo al file (dopo le righe che iniziano per `#`)
- Digitare:
@reboot /usr/bin/gpio_buttons &
- Premere `ctrl-x`, quindi `Y` e poi invio.
- Riavviare il sistema (**sudo shutdown -r now**)

Voilà, quando il sistema ripartirà **gpio_buttons** verrà lanciato in background, pronto a rispondere ai vostri comandi.

Per testare il tutto, occorre collegare due interruttori sulla GPIO, il che richiede l'acquisto di un po' di cavetti, di due pulsanti e l'uso di un saldatore. Cavi e pulsanti possono essere presi a pochi euro in qualunque negozio di elettronica o su aliexpress a prezzi davvero ridicoli. Per la nostra prova, abbiamo utilizzato quelli mostrati in *fig. 1*, particolarmente economici ed adatti a questo scopo. Il collegamento dei cavi ai pulsanti dipende dal tipo di bottone che utilizzate; quelli da noi utilizzati vanno saldati come mostrato in *fig. 2* (perdonate la qualità della saldatura, non siamo particolarmente ferrati nell'uso del saldatore). Il prossimo passo è quello di collegarli alla GPIO, noi abbiamo scelto i pin 8 e 9, collegando un polo al pin GPIO corrispondente e l'altro ad uno dei ground. Fate riferimento allo schema in *fig. 3* per scegliere i pin di vostro gusto e all'utility **gpio_readall** per trovarne l'id. Per il pin 8, le posizioni sono la 3 e la 9, mentre per il pin 9 occorre collegare la 5 e la 6. Fate molta attenzione in questa fase, collegare nel modo sbagliato pin sulla GPIO può danneggiare gravemente il Raspberry. Nel nostro test, abbiamo deciso di assegnare un bottone allo shutdown e l'altro al reboot, con un tempo di pressione di tre secondi prima di eseguire l'azione e dunque le righe di configurazione necessarie da inserire in `"gpio_buttons.conf"` sono:

```
assign_pin = 8, shutdown, 3
assign_pin = 9, reboot, 3
```

Copiatele ed incollatele nel file `"/home/pi/gpio_buttons.conf"`.

Lanciate **gpio_readall** per controllare che i pulsanti funzionino a dovere e se è tutto a posto, lanciate **gpio_buttons** e provate a tenerne premuto uno per almeno tre secondi per accertarvi che tutto proceda come previsto. Se tutto risponde come deve, potete configurare **crontab** per lanciare l'utility al boot del sistema come spiegato qualche paragrafo fa e da quel momento in poi potrete usare i pulsanti per spegnere e riavviare RetroPie in qualunque stato si trovi la macchina. Questa caratteristica è particolarmente utile per configurazioni "kiosk mode", dove l'uso dei menù di gestione viene normalmente inibito per evitare che l'utilizzo da parte del pubblico crei problemi alla configurazione del sistema.

joytest

Questa semplice utility, analogamente a **jstest** presente nella distribuzione RetroPie, può testare il funzionamento dei joypad collegati al sistema. Nel nostro caso però, **joytest** può (e deve) essere utilizzato per configurare **joyrun**, visto che i codici tasti utilizzati sono gli stessi (sono differenti da quelli riportati da **jstest**). L'uso è molto semplice, basta lanciarlo digitando: **joytest** e premere i tasti e muovere i controlli direzionali (digitali ed analogici) del joypad per controllare se tutto funziona a dovere. Se ci sono più controller connessi, con i tasti '+' e '-' si può passare dall'uno all'altro. Premendo il tasto 'q', si torna al prompt di shell.

raspi2png

Con questa utility, l'unica non creata da noi ma da Andrew Duncan, potete riversare in un file .png tutto ciò che è mostrato a video. Di fatto, **joyrun** e **gpio_buttons** si affidano ad essa per le funzionalità di salvataggio degli screenshot. L'uso è molto semplice, basta lanciarla per ritrovarsi dopo qualche secondo con un file "snapshot.png" nella cartella da cui la si è invocata. E' possibile modificarne il comportamento tramite i seguenti switch da passarle su linea di comando:

- **--pngname,-p**
indica un nome file per lo screenshot (di default è "snapshot.png").
- **--height,-h**
altezza dell'immagine. Di default è posto pari all'altezza dello schermo.
- **--width,-w**
larghezza dell'immagine. Di default è pari alla larghezza dello schermo.
- **--compression,-c**
livello di compressione dell'immagine (0-9).
- **--delay,-d**
attende per il numero di secondi indicati prima di salvare la schermata (di default la salva immediatamente dopo essere stata invocata).
- **--display,-D**
ID del display da considerare per lo snapshot (default zero).
- **--stdout,-s**
manda l'output allo stdout
- **--help,-H**
mostra informazioni sull'uso.

tempmon

Nella sua estrema semplicità, **tempmon** può rivelarsi un prezioso alleato per chi ha problemi di surriscaldamento del RetroPie. Una volta lanciata, questa utility stampa a video la temperatura del sistema ad intervalli di 10 secondi. Se temete che nel box dove è alloggiato, il raspberry stia soffrendo per il gran caldo o volete assicurarvi che la temperatura resti sotto il livello di guardia anche durante le attività più impegnative, **tempmon** può darvi una mano. L'unico parametro variabile (da linea di comando) è il tempo intercorso tra una rilevazione e l'altra, espresso in secondi. Ad esempio, digitando: **"tempmon 3"** (senza doppi apici, naturalmente), verrà effettuata una misurazione ogni 3 secondi invece che 10. Tutto qui.

gpio_readall

Siamo giunti all'ultima utility di questa raccolta. Con essa possiamo mostrare lo stato di tutti i pin dell'interfaccia GPIO, permettendoci di testare i pulsanti ed i dispositivi connessi. E' più comoda dell'utility **gpio** fornita con la WiringPI, poiché l'output viene continuamente mostrato ad intervalli di un secondo invece che una volta e basta (di fatto, lancia **"gpio readall"** in loop), permettendo di effettuare più facilmente i nostri test. Per chiuderla, basta premere ctrl-c.

Disclaimer (ovvero, fatelo a vostro rischio e pericolo)

Tutte le utility presentate qui e scaricabili in formato sia sorgente che binario tramite il sito di RetroMagazine, benché siano state testate con successo su sistemi Raspberry 3B e Raspberry 3B+, vengono fornite senza alcuna garanzia da parte dell'autore o di RetroMagazine e lo stesso vale per le istruzioni date in questo articolo per il loro uso. Danni di tipo logico e fisico di portata non determinabile a priori possono essere causati, a titolo di esempio ma non limitatamente a questi, da malfunzionamenti, bug non noti e/o imperizia nell'uso, come pure da errori e imprecisioni nelle istruzioni d'uso o nella loro applicazione, in special modo quelle relative alla costruzione ed installazione di pulsanti fisici sull'interfaccia GPIO. L'autore e RetroMagazine declinano qualunque tipo di responsabilità e non risponderanno di alcun danno, di qualunque entità, provocato dall'uso delle utility e/o delle relative istruzioni d'uso.

Conclusioni

Le RetroPie Utility, RPU per gli amici, sono state scritte in un arco temporale di due anni, evolvendosi dai semplici script iniziali alla forma più complessa ma più elastica e funzionale che avete qui conosciuto. Dalla loro nascita ad oggi, ci hanno aiutato a personalizzare diversi sistemi di gioco basati su RetroPie, aumentandone l'usabilità e permettendoci di passare ore con la sperimentazione di questa efficace distribuzione orientata al retrogaming. Ci auguriamo che possano portare qualche beneficio anche a voi lettori e che possano crescere tra le vostre mani o vi stimolino a crearne altre, magari in **FreeBasic**, che personalmente riteniamo un eccellente strumento per lo sviluppo di questo tipo di applicazioni. In entrambi i casi, fatecelo sapere, ci farà enorme piacere.

Link

- **Link per scaricare le RPU:**
<http://www.retromagazine.net/getrm.php?id=rpu>
- **RetroPie:**
<https://retropie.org.uk>
- **RetroOrangePie:**
<http://www.retroorange.org/>
- **Home di raspizpng:**
<https://github.com/AndrewFromMelbourne/raspizpng>
- **FreeBasic:**
<https://www.freebasic.net/>
- **Installare FreeBasic su Raspberry:**
<http://scruss.com/blog/2015/03/25/runnig-freebasic-on-raspberry-pi/>

- **WiringPi:**
<http://wiringpi.com/>
- (www.python.org un linguaggio moderno NdLeo) (e che sul Raspberry può tirarvi giù la cpu come fosse un attacco DDoS... NdGiorgio)



Figura 1

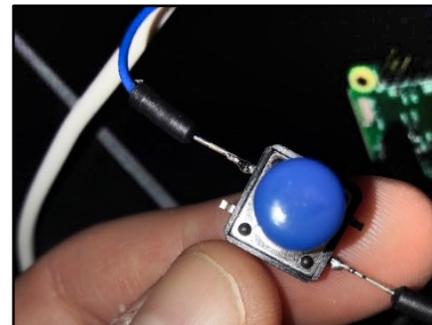


Figura 2

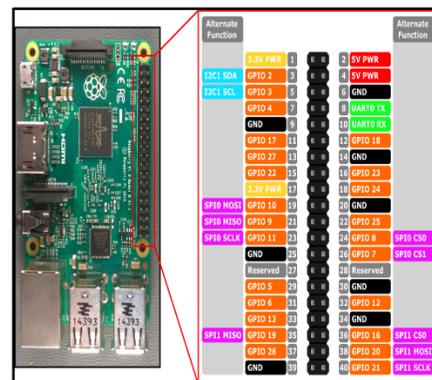


Figura 3

BENVENUTI IN UN MONDO FANTASTICO

Introduzione al gioco di ruolo

di Starfox Mulder



Era il 1996. I videogiochi avevano iniziato ad esplorare il 3D in maniera significativa ed un certo Daggerfall della Bethesda tentò di superare i limiti della macchina in modi mai visti prima per portare il giocatore a vivere l'esperienza più libera che si potesse avere su un personal computer. Ricordo anche che la scuola era appena cominciata. I primi giorni delle superiori portarono nuovi amici e tra loro conobbi Nico, il quale un giorno mi disse "Se ti piace Heroquest, questo ti piacerà di più". Si riferiva a quello che già per l'epoca era considerato un mito: Dungeons and Dragons. Radunammo un gruppo di giocatori spulciandoli tra i miei ed i suoi amici ed iniziammo a vederci spessissimo nel dopo scuola per vivere assieme infinite avventure grazie ad alcuni manuali, qualche dado, matite e fogli, più tantissima fantasia.

Ciò che nacque come una scusa per evadere dalle ansie adolescenziali si trasformò velocemente in una solida amicizia e quei ragazzi conosciuti al tempo sono ancora oggi i migliori amici che abbia mai avuto, pronti a sostenermi nei momenti difficili ed a farmi da testimoni al matrimonio.

Il gioco di ruolo fu solo un pretesto, sono sicuro che ci siano tanti validissimi modi per fare amicizia e creare legami duraturi, ma sospetto anche che quello capitato a me innescò delle dinamiche differenti.

Prima ancora di diventare amici infatti eravamo stati compagni d'arme in battaglia, investigatori dell'occulto impegnati in un'eterna lotta contro delle orribili creature ancestrali, esseri dannati costretti a nascondersi dalla luce del sole e tanto altro ancora. Eravamo giocatori di ruolo ed i mille mondi vissuti assieme ci avevano mostrato

parti del nostro carattere che nessun'altra esperienza avrebbe potuto evidenziare senza porci in un reale pericolo.

Ma qui mi fermo o chi sta leggendo penserà che a vivere le avventure da me citate fossimo noi in prima persona. Non è così. Giocare di ruolo significa calarsi nei panni di qualcun altro, un alter ego immaginario capace di affrontare imprese a noi precluse ma spesso anche incapace di arrivare dove la nostra conoscenza di ragazzi del ventesimo secolo ci permetteva.

Seguitemi, vi mostrerò l'infinito.



Tutti voi avete giocato di ruolo almeno una volta nella vita. Giocare a guardia e ladri è interpretare un ruolo. Certo, ne è una versione estremamente blanda e semplificata ma di fatto almeno per quel momento vi siete calati nei panni di un poliziotto o un malvivente. Proviamo ora ad ampliare il concetto e codificarlo.

Io decido di interpretare il malvivente ma non mi limito a recitarne la parte come se affrontassi un'improvvisazione teatrale, bensì creo su carta un mio alter ego. Lo delinco dandogli un nome, un aspetto fisico e delle caratteristiche che lo identifichino, come potrebbe essere un'ottima agilità o una gracile costituzione. Queste caratteristiche già sarebbero sufficienti a renderlo ben diverso da me ed a portarmi ad un'interpretazione delle sue gesta funzionali al carattere che deve aver formato nel tempo. Un individuo cagionevole di salute non amerà certo passare buona parte della giornata all'aria aperta ma viceversa potrebbe trovare pratico vivere dei suoi scippi se scopertosi

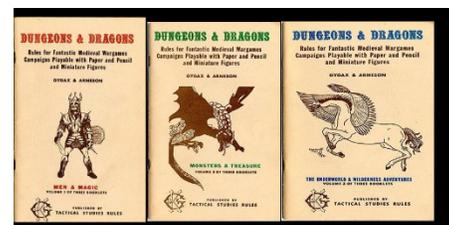
abbastanza abile da non venir mai individuato dalle sue vittime.

Con questo veloce volo di fantasia ho creato un personaggio ed ora non mi resterà che fargli vivere tante avventure finché non avrà deciso di ritirarsi a vita agiata o sarà morto nel tentativo. Mi basta così poco? Di certo no, come minimo ho bisogno di un altro giocatore che interpreti il game master.

La figura del Game Master (GM da ora in poi) è fondamentale in qualsiasi gioco di ruolo poiché senza di lui nulla avrebbe inizio. Se il giocatore interpreterà un singolo personaggio di fantasia, al GM spetterà interpretare tutti gli altri personaggi non giocanti (PNG, individui non interpretati da alcun giocatore) presenti nel mondo di gioco, descrivere gli ambienti in cui il personaggio vivrà le sue avventure e tutti gli eventi che accadranno in esso.

Se volessimo fare un paragone videoludico immaginate che in un qualsiasi titolo della saga di Final Fantasy i giocatori interpretano un singolo elemento del gruppo mentre il GM è l'avventura stessa, tutto il resto del videogioco.

Non spaventatevi però poiché non ci sono complicati software da sviluppare con chissà quale linguaggio di programmazione, tutto il mondo di gioco si dipanerà nella testa dei giocatori grazie alla semplice narrazione.



"...appena entrati nella caverna vi accorgete che l'aria è umida ed il soffitto basso ma quello che vi circonda è parzialmente opera dell'uomo. Le pareti sono scavate, una fiaccola illumina leggermente l'area e di fronte a voi notate due gallerie dividersi davanti a voi. La prima pare insinuarsi verso il basso, pur mantenendo la direzione Ovest che avete preso entrando nella caverna, mentre l'altra non sembra scendere ma fa una curva verso sud. Cosa fate?"

Questo potrebbe essere un esempio di ciò che il GM dirà durante una sessione di gioco. Cosa farà il vostro personaggio? Tornerà indietro o si avventurerà in una delle due gallerie? Perlustrerà la stanza o stanco dalle fatiche accumulate si accamperà per un sonnellino? Le possibilità sono infinite e lo sono in una maniera che mai nessuno gioco di ruolo gestito da un software potrà esserlo perché solo la mente umana può sviluppare infinite variabili così da adattarsi ad ogni possibile contesto si vada a creare di lì in avanti.

Avrete a sto punto intuito che sì, è possibile giocare anche solo in due persone, ma ci si diverte poco.

Creato un personaggio e trovato un volenteroso che ricopra il ruolo di GM è altamente consigliato arruolare altri giocatori che interpretino insieme a noi altrettanti personaggi da far muovere all'interno delle avventure che andremo a vivere. Quanti? Dipende dal gioco che andremo a fare. Nel già citato Dungeons and Dragons, storico esempio di gdr fantasy le cui avventure spesso di svolgono in sotterranei irti di pericoli, giocare in 7 o 8 persone potrebbe ancora andare bene ma in gdr più investigativi, come nel lovecraftiano "il Richiamo di Cthulhu", numeri superiori ai 4-5 partecipanti rallentano incredibilmente il gioco e rischiano di annoiare alcuni.

Questo avviene perché ovviamente non tutti possono parlare assieme e le azioni vengono divise in turni, nei momenti in cui questo si riveli necessario, come ad esempio durante gli scontri fisici.



"...sbucati dalla galleria vi trovate di fronte ad un lago sotterraneo ma notate subito che intorno ad esso stanno banchettando, con dei resti umani, alcuni goblin. Tirate l'iniziativa e ditemi cosa fate."

Si iniziano a tirare i dadi e l'azione prende vita. Sarebbe troppo facile dire che il mio personaggio, abile guerriero, cala la spada e tronca in due il busto dell'avversario. Se ci riuscirà o meno sarà determinato in parte dal talento del personaggio ma in parte anche il tiro di dado adibito a tale verifica farà da discriminante. Una volta determinato l'ordine in cui i personaggi (compresi quelli interpretati dal GM) agiranno, ognuno dirà cos'ha intenzione di fare e tirerà un dado per vedere se riuscirà o meno nell'impresa. Un guerriero potrebbe caricare l'avversario ed attaccarlo con la sua spada mentre un mago potrebbe rimanere fermo sul posto e formulare un incantesimo di protezione. Un ladro potrebbe nascondersi nelle ombre con l'intento di aggirare il nemico e colpirlo alle spalle mentre un arciere potrebbe incoccare una freccia e mirare alla testa di una delle creature. Anche qui le possibilità sono infinite e di fatto anche la fuga è concessa. Dichiarata l'azione i dadi rotolano e ci diranno se saremo o meno riusciti in ciò che volevamo far fare al nostro alter ego.



Il GM dice "Siete di fronte al Capo dei Coboldi che vista la disfatta del suo seguito appare spaventato di fronte a voi. La creatura esclama: <Vi prego, abbiate pietà di me. Se mi risparmierete vi condurrò al posto in cui tengo il mio tesoro.> Cosa fate?"

Fabio interpreta Gerald, il ladro sopravvissuto a tante avventure, che rivolgendosi a Larandhiel, l'elfa guerriera interpretata da Claudia, dice: <L'ultima parola è tua capo, ma se vuoi fidarti di sto canide lascia che perlustri la zona che ci indicherà per evitare di cadere in una trappola.> Claudia se la ride per il gesto di improbabile cavalleria proposto dal personaggio di Fabio alla sua elfa e di risposta le fa esclamare: <E' un'ottima idea Gerald, se usciremo di qui ancora vivi magari ti permetterò di offrirmi quel calice di vino di cui parlavi prima.>

Il gioco è imprevedibile ed i giocatori, come il master, dovranno essere sempre pronti a reagire in base all'occasione, proprio come nella vita reale. Un buon GM crea una storia e degli intrecci narrativi ma non obbliga i giocatori in binari prestabiliti: si adatta alle loro azioni. Ogni situazione è mutevole ed anche i rapporti tra i personaggi interpretati non sono sempre scontati. Quello che parte come un gruppo affiatato potrebbe trasformarsi in un covo di vipere così come quel personaggio non giocante che inizialmente ha dato del filo da torcere ai PG potrebbe in seguito diventare un solido alleato.

DUNGEONS & DRAGONS



Volendo iniziare a parlarvi di uno specifico gioco di ruolo ho scelto il capostipite. Scontato? Non così tanto. Come detto poc'anzi ho iniziato la mia avventura nel mondo ruolistico con D&D ma dopo poco me ne sono discostato preferendogli giochi più maturi e complessi. Ero poco più che adolescente all'epoca e questa scoperta di nuovi sistemi di gioco mi portò a rifiutare il primo D&D, proprio il gioco con cui mi avvicinai al genere. Lo iniziai a vedere come limitato, desueto, l'equivalente musicale di Elvis per chi ha appena scoperto il punk rock. Passarono gli anni e, vuoi la nostalgia, vuoi la curiosità di vedere se quel vecchio amico era ancora disposto a farmi divertire, ripresi in mano la famosa Scatola Rossa circa cinque anni fa, rivalutandolo in toto.

D&D fu scritto da Gary Gygax e Dave Arneson nel 1974. La gestazione fu lunga ed originariamente doveva essere solo un regolamento per un wargame ma poi si evolse fino a creare il primo ambiente di gioco

ruolistico. C'erano i dadi sì, ma c'erano anche le miniature nell'intenzione degli autori. Le graziose statuine sono da me molto apprezzate durante le sessioni di dungeon ma tutt'altro che indispensabili, quindi non fatevene un cruccio se non le avete.

La prima edizione di D&D era una raccolta di fascicoli contenuti in una scatola ma in breve iniziò a crescere e ne arrivarono una seconda ed anche una terza edizione. Proprio quest'ultima approdò in Italia a metà degli anni 80, nella versione curata da Frank Mentzer e proposta a noi dalla Editrice Giochi. Quattro scatole: Rossa (Basic), Blu (Expert), Verde acqua (Companion) e Nera (Master). Ce ne sarebbe stata una quinta grigia (Immortal) ma da noi non venne mai tradotta.

Cosa cambiava tra le varie scatole? Il livello delle avventure.



Nella prima erano contenute le regole per personaggi dal 1° al 3° livello, divise su due manuali (quello del giocatore e quello del master) più un set di 7 dadi dalle forme più svariate. A parte il noto dado cubico a 6 facce, presente in tantissimi giochi da tavolo o d'azzardo, erano anche presenti infatti dadi a 4, 8, 10, 12 e 20 facce, utili ognuno in un contesto differente.

Il secondo set, l'Expert, permetteva poi di portare i personaggi dal 4° al 14° livello, aggiungeva molte regole aggiuntive come quelle per le lunghe campagne di gioco (insieme di avventure concatenate) e le avventure all'aperto.

Il terzo set, il Companion, introduceva i livelli dal 15° al 26° così come le regole per gestire grandi battaglie e lo Schermo del Master. Infine il quarto set, il Master, portava i personaggi fino al 36° livello. Quando parlo di regole per personaggi di tot livello intendo che tutto quel che li riguardava era descritto in essi, dai tipi di mostri che potevano

affrontare all'equipaggiamento che avrebbero potuto reperire. Nessuno vietava al GM di utilizzare mostri più deboli ma sarebbero risultati noiosi da sconfiggere, così come mostri più potenti avrebbero significato morte certa per i personaggi.

Lo Schermo del Master è un accessorio oggi molto comune a tutti i GDR ma per l'epoca avveniristico. Un foglio di cartoncino piegato in tre parti perché fungesse da "muro separatorio" tra il master ed i giocatori. Da un lato una bella immagine mentre dall'altro (quello interno) le tabelle più utili da tenere sott'occhio per non stare a sfogliare costantemente tutti i manuali accumulati. Lo schermo aveva un'altra importante funzione: quella di celare ai giocatori i risultati dei dadi. Se da una parte è bello il dominio del caso, dall'altro dei piccoli ritocchi ai risultati permisero già ai primi master di non sterminare un gruppo di avventurieri alle prime armi di fronte ad un eccesso di fortuna da parte dei mostri da lui "governati". Certo, alcuni GM particolarmente bastardi usarono l'espedito al contrario ma non ci volle molto ai giocatori furbi per capire da quali sadici soggetti stare alla larga.

D&D fu un fulmine a ciel sereno per l'epoca. Prendendo a piene mani dall'immaginario fantasy di scrittori come Tolkien, Moorcock o Howard, così come dalle epiche germaniche o norrene, l'ambientazione di dungeons and dragons (alcune delle ambientazioni dovrei dire, dato che ce ne furono svariate e per tutti i gusti) calò i giocatori nel ruolo di maghi, guerrieri, nani, elfi e molto altro ancora.

La cerca di un antico tesoro, la salvezza di una principessa rapita o la sconfitta di un orribile minaccia che grava su uno sperduto villaggio sono solo alcuni degli incipit da cui partivano le avventure, per poi magari portare i giocatori a scoprire che la principessa non era affatto stata rapita, il villaggio era la vera minaccia ed il tesoro...

Il regolamento può inizialmente apparire complesso ai non esperti ma in poco tempo lo si riuscirà ad apprendere al meglio. Tutto sommato le regole da imparare sono poche e nella scatola rossa si perde più tempo a fare esempi di gioco che a regolamentarlo seriamente. In fondo furono i primi ed è

comprensibile per l'epoca non poter dare nulla per scontato.

Il grande pregio ed il grande difetto di D&D sta proprio nelle regole: poche!

In tutti i gdr successivi, a partire dal suo diretto seguito Advanced D&D, furono introdotte le abilità, capacità che i personaggi avevano acquisito nel tempo atte a determinare se fossero in grado di scalare quella parete o ascoltare quel rumore proveniente dalla foresta a fianco, mentre in D&D non ve ne era traccia.

Sei caratteristiche (forza, destrezza, costituzione, intelligenza, saggezza, carisma), il tiro per colpire, l'indicatore dell'armatura, un pugno di talenti se si interpreta il ladro ed incantesimi se si è mago, chierico o elfo: fine!

Oddio, proprio fine no, ci sarebbe da aggiungere l'equipaggiamento e i punti ferita, ma quel che conta in sostanza è la semplicità del tutto. Un personaggio di D&D può essere creato in 5 minuti e tutte le regole non scritte (tantissime) possono essere compensate da un tiro su una delle sei caratteristiche o sul semplice buonsenso, a seconda di come ci si trovi meglio.

Il bello ed il brutto dell'avere poche regole sta proprio nella flessibilità di tutto il sistema.

Da ragazzo me ne disinnamorai per quello, da adulto la fiaccola si è riaccesa per lo stesso motivo.

Un anno fa ho terminato una campagna durata 3 anni con amici vecchi e nuovi. Io facevo il game master, loro gli 8 personaggi protagonisti. Gente conosciuta quando avevamo 14 anni che ora ne ha compiuti 36. Ragazzi che all'epoca non si facevano neppure la barba ed ora portano il figlio alle sessioni di gioco. Una vita passata insieme condividendo emozioni reali ed immaginarie ma pur sempre emozioni ed ancora più importante: vissute assieme.



RetroMath:

La matematica degli stormi di uccelli

di Giuseppe Fedele

Chi non ha assistito, osservando il cielo al tramonto, al meraviglioso spettacolo di stormi di uccelli che si aggregano in migliaia di individui coordinandosi in maniera perfetta e dando vita a bellissime coreografie e forme?

Quali sono i meccanismi alla base di questi comportamenti collettivi?

In questo articolo verranno introdotti alcuni modelli di interazione tra individui che consentono di spiegare alcune dinamiche del mondo animale.

Si analizzeranno alcune proprietà di questi modelli e si vedrà come implementarli sui nostri amati retroPC.

Bibliografia:

[1] B.A. Francis, M. Maggiore, Flocking and Rendezvous in Distributed Robotics. Springer, 2016.

[2] M. Mesbahi, M. Egerstedt, Graph Theoretic Methods in Multiagent Networks. Princeton University Press, 2010.

[3] F. Cucker, S. Smale, Emergent Behavior in Flocks. IEEE Trans. Automatic Control, 52, 5, 2007, pp.852-862.

[4] G. Fedele, L. D'Alfonso, A Model for Swarm Formation with Reference Tracking, 56th Annual Conference on Decision and Control, 2017.

Uno swarm (stormo) di agenti è un insieme di componenti interagenti, come animali, robot, ecc. che danno luogo ad un comportamento globale che dipende in maniera non lineare dalle leggi che governano le singole componenti.

Immaginiamo uno stormo di uccelli che migrano verso sud prima dell'inverno. Lo stormo è composto da uccelli identici, ma nessun uccello è il capo, e nessuno può vedere sempre tutti gli altri uccelli. Gli uccelli hanno ovviamente un obiettivo comune: volare da un'area generale ad un'altra area specifica. Gli uccelli cooperano per svolgere la funzione generale di migrazione di gruppo. Possiamo pensare a questo stormo di uccelli come un sistema distribuito, l'aggettivo "distribuito" non significa necessariamente geograficamente distribuito, ma piuttosto distribuito in funzione o autorità. Oltre ad uno stormo di uccelli si potrebbe immaginare un branco di pesci o una colonia di formiche. Ora pensiamo ad un sistema di questo tipo, dove ogni uccello (o pesce o formica) è sostituito da un robot mobile: un rover a quattro ruote a terra o un quadricottero in aria.

Il problema del controllo coordinato di una rete di robot mobili autonomi è di interesse per il controllo e la robotica a causa della vasta gamma di potenziali applicazioni: esplorazione planetaria, operazioni in ambienti pericolosi, giochi come il calcio robotico, ecc. [1].

Persino molti giochi si sono ispirati alle logiche di swarm. Swarm! (Figura 1) è un gioco sparattutto, per Commodore Vic-20, sviluppato nel 1982 dalla Tronix Publishing Inc., basato sull'incontro con uno sciame di vespe androidi intergalattiche, allevate da una sfuggente vespa regina e la sua scia di uova mutanti. Nel 1998 la Reflexive Entertainment sviluppò per PC il gioco Swarm (Figura 2) in cui un giocatore doveva combattere contro creature aliene al fine di ottenere minerali speciali. Quelli citati e altri giochi simili hanno in comune il movimento di oggetti basato su logiche di swarm (in realtà spesso più che in swarm, gli oggetti si muovono in modo random, ahimè!!!).

In questo articolo vogliamo approfondire alcuni aspetti matematici di queste logiche in

modo che ognuno possa simulare, magari utilizzando il proprio retroPC, uno stormo di agenti.

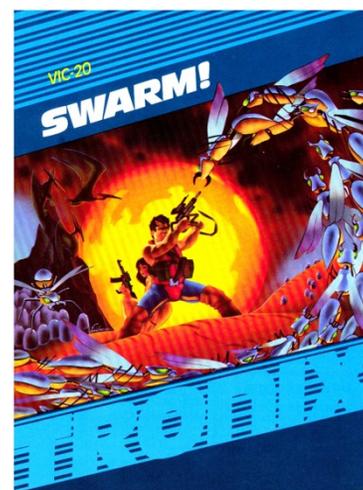


Figura 1. Swarm! Per Commodore Vic-20, 1982.

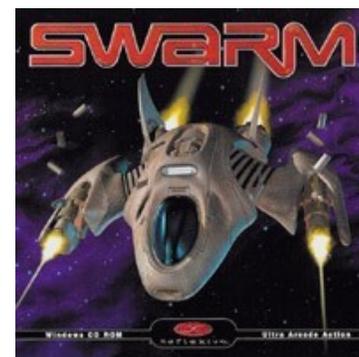


Figura 2. Swarm. Per PC, 1998.

Il problema del consenso

In una rete di agenti, il **consenso** indica il raggiungimento di un accordo riguardo una certa quantità di interesse che dipende dallo stato di tutti gli agenti. Ad esempio si può volere che tutti gli agenti dello stormo vadano a finire in una posizione ben definita dello spazio, oppure che gli agenti si muovano in una stessa direzione con una stessa velocità.

Un algoritmo o **protocollo di consenso** è una legge di interazione che regola lo scambio di informazioni tra un agente ed i suoi vicini nella rete. Ogni agente utilizza lo stesso algoritmo

e prende le proprie decisioni sulla base delle informazioni locali disponibili e a quelle che riceve dagli altri.

Un classico esempio di consenso è rappresentato dal seguente modello differenziale:

$$\dot{z}_i(t) = \sum_{j \in \mathcal{N}_i} (z_j(t) - z_i(t)), \quad i = 1, \dots, n$$

dove $z_i(t)$ rappresenta l'informazione associata all'agente i -mo, mentre \mathcal{N}_i è l'insieme dei vicini dell'agente i , ovvero l'insieme degli agenti con cui è possibile scambiare informazioni. L'informazione contenuta in $z_i(t)$ è proprio l'informazione che serve per il coordinamento degli agenti. Questa può essere: posizione, velocità, temperatura, ecc.

Diciamo che gli agenti della rete raggiungono il consenso se e solo se

$$z_1 = z_2 = \dots = z_n$$

Esempio. Supponiamo che $z_i(t)$ rappresenti la posizione dell'agente i in uno spazio bidimensionale. Si vuole quindi che tutti gli agenti raggiungano il consenso in un punto dello spazio. In Figura 3 è mostrato, per ogni agente, il suo vicinato. Si tratta, come è ovvio, di un grafo in cui ad ogni nodo corrisponde un agente della rete, ed esiste un arco tra due nodi i e j , se l'agente corrispondente appartiene al vicinato dell'altro e viceversa (tra due agenti connessi da un arco è quindi possibile scambiare informazioni).

Nel grafo mostrato il vicinato del nodo 1 è {2}, del nodo 2 è {1,3,4} e così via.

In Figura 4 è invece mostrata la simulazione del protocollo del consenso. Come si vede, gli agenti partono dalle loro posizioni iniziali (pallini blu) per convergere ad una posizione comune (pallino rosso).

Si dimostra che il consenso viene raggiunto se il grafo delle connessioni è connesso, ovvero se per ogni coppia di nodi esiste un cammino che li collega. Inoltre il punto in cui avviene il consenso coincide con il centroide delle posizioni iniziali [2].

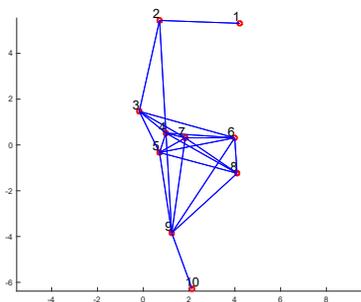


Figura 3. Grafo delle connessioni tra agenti

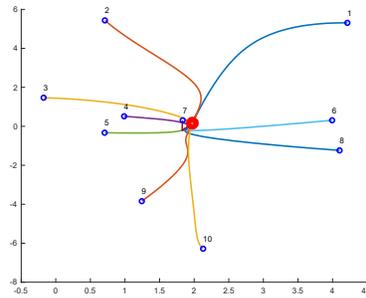


Figura 4. Protocollo del consenso per il grafo di Figura 3.

Il Problema del flocking.

Il **flocking** è un tipico comportamento osservabile nelle specie animali. Rientra nei comportamenti collettivi secondo cui gli animali si aggregano in stormi senza che ci sia un coordinamento centrale (ovvero senza la presenza di un leader che coordina i movimenti dei singoli). Nello specifico ogni componente segue delle semplici regole in maniera autonoma dando vita ad un meraviglioso risultato "macroscopico" in cui l'intero stormo appare perfettamente coordinato.

Un modello matematico che descrive la dinamica dell'allineamento degli uccelli è stato proposto da Felipe Cucker e Steve Smale nel 2007 [3]. Ogni uccello coordina il suo movimento rispetto al movimento degli uccelli vicini. Nello specifico si ipotizza che ciascun uccello aggiusti istante per istante la propria velocità sommando ad essa una media pesata delle differenze della sua velocità con quella degli altri uccelli.

Indichiamo con $v_i(t)$ la velocità dell'uccello i -mo che sarà quindi composta dalle velocità lungo le due direzioni x e y .

Il modello presuppone che ogni uccello aggiorni la sua velocità come

$$v_i(t+1) = v_i(t) + \sum_{j=1}^n a_{ij}(v_j(t) - v_i(t))$$

dove a_{ij} è un peso che decresce all'aumentare della distanza tra gli uccelli i e j :

$$a_{ij} = \frac{K}{(\sigma^2 + \|z_i(t) - z_j(t)\|^2)^\beta}$$

dove K, σ e β sono opportune costanti positive. Il termine a_{ij} modella quindi l'influenza tra coppie di uccelli: l'idea è che al crescere della distanza l'influenza che ciascuno esercita sull'altro diventa trascurabile. Se β è scelto minore di 0.5 allora le velocità dei vari uccelli tendono a diventare uguali e lo stormo si muove in maniera coordinata lungo una direzione (Figure 5 e 6). Le posizioni degli uccelli si modificano come

$$z_i(t+1) = z_i(t) + \Delta T v_i(t)$$

dove ΔT è l'intervallo di tempo tra un aggiornamento e l'altro.

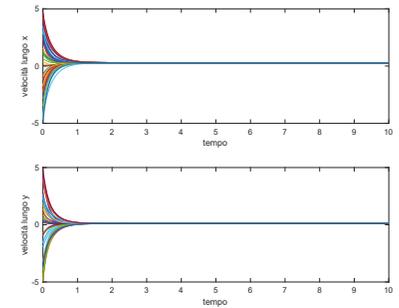


Figura 5. Velocità dello stormo.

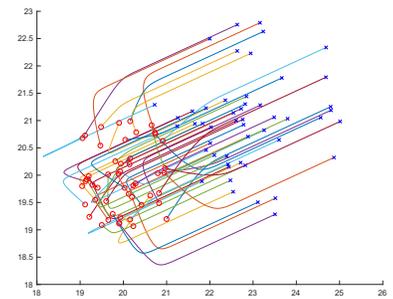


Figura 6. Posizioni dello stormo.

Un altro modello proposto recentemente in [4] è il seguente:

$$\begin{aligned} \dot{z}_i(t) &= -(z_i(t) - z_0) \\ &+ \frac{1}{n} M \sum_{j=1}^n \frac{z_i - z_j}{\|z_i(t) - z_j(t)\| + \epsilon}, \quad i = 1, \dots, n \end{aligned}$$

In questo modello z_0 rappresenta il target verso cui tutti gli agenti si muovono. ϵ è una quantità piccola positiva.

Invece M è una matrice che pesa e "mixa" in maniera opportuna le coordinate x e y di ogni agente. La caratteristica interessante di questo modello è che gli agenti dello swarm entrano in tempo finito in un cerchio centrato in z_0 di raggio che dipende dalla matrice M (in particolare dalla norma della matrice).

Inoltre scegliendo opportunamente tale matrice è possibile dare forme diverse allo swarm: cerchi, ellissi, rette. Ed infine è possibile che gli agenti ruotino attorno al target.

Ad esempio con

$$M = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

lo swarm si organizza in un cerchio (Figura 7) attorno al target (l'origine degli assi),

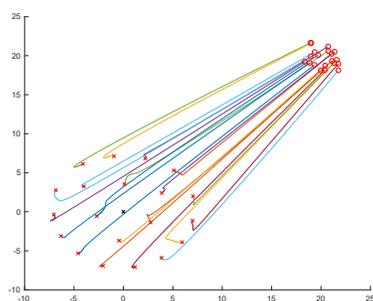


Figura 7. Swarm in un cerchio.

con

$$M = \begin{bmatrix} 10 & 0 \\ 0 & 5 \end{bmatrix}$$

lo swarm si organizza in una ellisse (Figura 8),

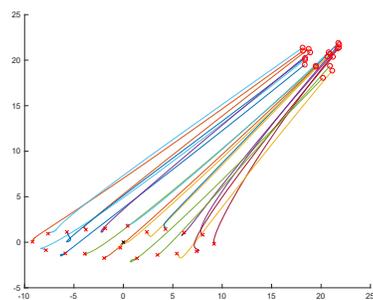


Figura 8. Swarm in una ellisse.

con

$$M = \begin{bmatrix} 0.1 & 0 \\ 0 & 10 \end{bmatrix}$$

gli individui dello swarm collassano su una retta (Figura 9). Infine con

$$M = \begin{bmatrix} 50 & 10 \\ -10 & 50 \end{bmatrix}$$

gli individui dello swarm raggiungono il target ed iniziano a ruotare attorno ad esso. La simulazione di questo swarm merita di essere realizzata utilizzando un retroPC, nello specifico il Commodore 128 per le sue routine grafiche (Figura 10). Il codice scritto usando CBM lo trovate nel riquadro.

La simulazione viene fatta discretizzando il modello in una finestra temporale di $T = 50$ con un passo di discretizzazione DT .

Per la discretizzazione basta ricordare che la derivata di un segnale $f(t)$ può essere approssimata come

$$\lim_{\Delta T \rightarrow 0} \frac{f(t + \Delta T) - f(t)}{\Delta T}$$

La posizione del target verso cui gli agenti tendono è contenuta nelle variabili x_0 e y_0 .

La subroutine che inizia dalla riga 5000 genera le posizioni iniziali degli NA agenti in maniera casuale. Le righe da 240 a 380 implementano le equazioni del modello. La posizione XY di ogni agente viene sovrascritta ad ogni iterazione e mostrata sullo schermo grafico con un cerchio di raggio 1 centrato nella posizione dell'agente (routine dalla riga 6000).

Modificando i valori della matrice M (riga 210) è possibile ottenere le varie configurazioni descritte.

Per adesso è tutto. Arrivederci ad un'altra avventura nel meraviglioso mondo di RetroMath.

Buone feste!!!

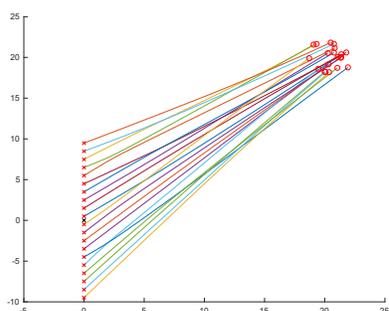


Figura 9. Swarm su una retta.

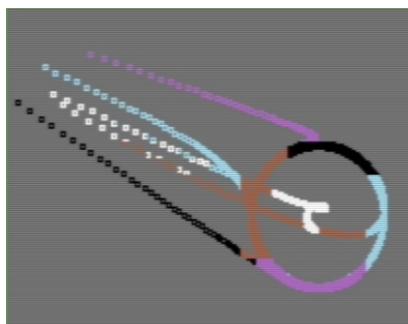


Figura 10. Swarm che ruota attorno al target

```

10 rem-----
20 rem Swarm model
30 rem Giuseppe Fedele
40 rem-----
120 rem tempo simulazione
130 T = 50
140 N = 1000
150 DT = T/(N-1)
160 rem numero agenti
170 NA = 5
175 ee = 0.001
180 dim X(NA), Y(NA)
190 rem M
200 dim M(2,2)
210 M(1,1)=50 : M(1,2)=10 : M(2,1)=-10 : M(2,2)=50
220 x0=160 : y0=100
230 gosub 5000
235 graphic 1,1
240 for k=1 to N-1
250 : for i=1 to NA
260 : ax=0 : ay=0
270 : for j=1 to NA
280 : if i<j then begin
290 : dd=sqr((X(i)-X(j))^2+(Y(i)-Y(j))^2)+ee
300 : ax=ax+(X(i)-X(j))/dd
310 : ay=ay+(Y(i)-Y(j))/dd
320 : bend
330 : next j
340 : bx=(M(1,1)*ax+M(1,2)*ay)/NA
350 : by=(M(2,1)*ax+M(2,2)*ay)/NA
360 : X(i)=X(i)-DT*(X(i)-x0)+DT*bx
370 : Y(i)=Y(i)-DT*(Y(i)-y0)+DT*by
380 : next i
385 : gosub 6000
400 next k
410 graphic 0
500 end
5000 rem condizioni iniziali
5010 for i=1 to NA
5020 : X(i)=20*(2*rnd(1)-1)+20
5030 : Y(i)=20*(2*rnd(1)-1)+30
5040 next i
5050 return
6000 rem visualizza
6030 : for i=1 to NA
6035 : color 1,i
6040 : circle 1,X(i),Y(i),1,1
6050 : next i
6060 : for j=1 to 1000:next j
6120 return
7000 for k=1 to N
7010 : for i=1 to NA
7020 : print X(i,k);" ";Y(i,k)
7025 : for j=1 to 1000:next j
7030 : next i
7040 next k

```

B.B.S. su C64? Si può fare!

di Marco Pistorio



Parte I – Introduzione

Abbiamo già parlato in un precedente articolo dei B.B.S. (ovvero "Bulletin Board Systems") e della possibilità di fruire dei loro contenuti tramite **VICE**, applicativo **TCPSER** ed un programma terminale che gira su C64.

Successivamente abbiamo visto come installare e configurare, almeno a grandi linee, uno dei software più popolari degli anni '90 per la gestione di un BBS, "MAXIMUS", che girava su sistemi MS-DOS.

Oggi vedremo un altro aspetto della questione. Proveremo a scoprire come far girare un B.B.S. all'interno del C64 (emulato tramite **VICE**), e permetterne l'accesso dall'esterno tramite **TELNET**.

Forse non è proprio immediato eppure, come da titolo di questo articolo, "Si può fare!.."

E' giusto premettere che il Commodore 64 non è, probabilmente, la piattaforma ideale per gestire B.B.S., sia a causa della esigua disponibilità di RAM, sia per le possibilità limitate offerte dal suo Floppy Disk Driver 1541 (e su e varianti 1541/II e così via)

Non è un caso infatti che, per questa finalità, venivano spesso adoperati PC MS-DOS equipaggiati con un certo quantitativo di memoria RAM e soprattutto con almeno un hard-disk quanto più capiente possibile.

Tuttavia anche il C64 può essere impiegato per un compito simile. Al posto del capiente hard-disk spesso si utilizzavano fino a 6 floppy disk drivers 1541 collegati al Commodore 64 simultaneamente, in maniera tale da disporre di più memoria di massa disponibile in tempi ragionevolmente brevi. E per quanto riguarda invece la RAM di base ci si accontentava di quella disponibile on-board sul "commie". Questo aveva un costo, chiaramente.

I programmi per gestire B.B.S. sul C64 mancano spesso di diverse "features" rispetto a quelli che girano su altre piattaforme, ad

esempio PC MS-DOS.

In genere non gestiscono velocità maggiori di 2400 baud (simboli trasmessi al secondo), non supportano tutti i protocolli di trasferimento dei files (XMODEM, YMODEM, YMODEM-G, ZMODEM ed altri ancora), non gestiscono spesso lo standard di visualizzazione caratteri ANSI e potrei continuare ad elencare ulteriori limitazioni ma preferisco fermarmi qui :)

Andiamo adesso a scoprire come implementare, in pratica, un BBS che giri su Commodore64.

La mia scelta è caduta su due soluzioni, molto diverse tra loro.

La prima è **CONTIKI-BBS**, un sistema minimale che permette agevolmente di mettere su un BBS che gira senza troppi problemi su Commodore 64.

Si tratta di una soluzione software abbastanza nuova, che sfrutta l'architettura **CONTIKIOS**.

E' un ambiente che supporta multitasking ed una suite TCP/IP (stack TCP/IP).

Il funzionamento di **CONTIKI-BBS** prevede l'uso di una cartuccia che funzioni da scheda ethernet per C64, che emuleremo tramite **VICE**.

Per maggiori approfondimenti:

<https://www.c64-wiki.com/wiki/Contiki>

<https://sourceforge.net/projects/contiki-bbs/files/0.3.0/>

Occupiamoci intanto di questa prima opzione.

Parte II – Contiki BBS

Contiki BBS è un sistema di Bulletin Board System (B.B.S.) per Commodore 64 ed una cartuccia Ethernet compatibile (ad esempio TFE, RRnet, 64NIC +).

I suoi punti di forza sono la sua leggerezza, la sua semplicità (sia nella configurazione che

nell'utilizzo) nonché l'accesso tramite connessione telnet standard.

Attualmente (versione 0.3.0) fornisce funzionalità BBS di base (selezionare da più boards, postare / leggere messaggi, pagina sysop).

Requisiti

1) **VICE versione 2.2 e superiori**

2) **Driver di rete in modalità promiscua.** Nei test ho adoperato **WINPCAP**, che è possibile scaricare liberamente dalla pagina: <http://www.winpcap.org/install/default.htm>

Dopo aver installato **WINPCAP** procedere con la configurazione di **VICE**.

Configurazione VICE

Menù Impostazioni, Impostazioni Cartuccia I/O, Ethernet Card Settings:

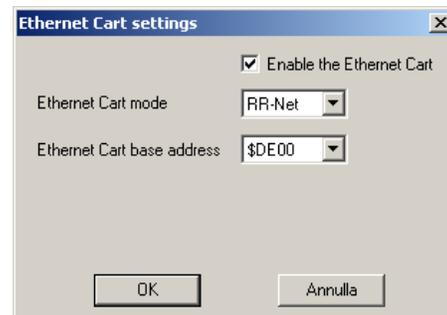
Spuntare: **Enable the Ethernet Card**

Impostare: **Ethernet Card Mode: RR-NET**

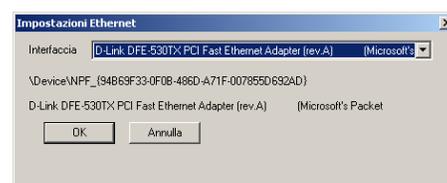
Ethernet Card Base address: **\$DE00**.

Dal menù Impostazioni, Impostazioni Ethernet scegliere l'interfaccia Ethernet che dovrebbe coincidere con la scheda di rete installata sotto Windows.

Dopo aver settato il tutto, salvare le impostazioni quindi chiudere e far ripartire **VICE**.



Al restart di **VICE**, la cartuccia "virtuale" ethernet Rrnet sarà collegata e disponibile.



Riferimento:

<https://www.commodoreserver.com/BlogEntryView.asp?EID=CDB68A9028654CCEA758D37C6DB3E05B>

A questo punto, sarete pronti per adoperare **CONTIKI-BBS**.

I passi successivi saranno:

A) Scaricare **CONTIKI-BBS** da questo link: <https://sourceforge.net/projects/contiki-bbs/>

B) Scompattare il file ctkbbs30.zip

C) Da VICE, "seleziona immagine disco", "drive 8" e scegliere il file immagine che è stato precedentemente scompattato, ovvero ctkbbs.d64

D) Configurare opportunamente **CONTIKI BBS**, lanciando il programma "BBS-SETUP.C64" contenuto nel file immagine ctkbbs.d64

Segue guida veloce relativa alla configurazione di **CONTIKI BBS**.

- Caricare il programma di configurazione di **Contiki BBS** impostando:
LOAD "BBS-SETUP.C64",8,1
- Eseguirlo con **RUN**

Apparirà il menù principale:

Contiki BBS configuration

*** Contiki BBS 0.3.0 setup ***

```
1...BBS base setup
2...BBS board setup
3...TCP/IP setup
4...User editor
q...Quit
```

1...Voce "BBS base setup" L'output seguente mostra un esempio di configurazione relativo alle impostazioni di base settaggio di base. In questa sezione potrai impostare i tuoi disk drivers, il prompt del BBS ed i valori di timeout.

Se hai più di un disk drive potrai conservare i boards/messaggi ed il file dati utenti in drives diversi. L'unico prerequisito è che il drive 8 è quello usato per il drive contenente il disco di avvio.

Nell'esempio sotto utilizzerò un secondo disk drive (device number 9) per memorizzare i dati board/messaggi ed un terzo disk drive (device number 10) per memorizzare il file dati utenti.

Chiaramente se si adoperava un solo floppy disk driver, tali files saranno contenuti nello stesso disco di avvio, ovvero il device number 8.

```
* BBS base setup
Enter boot drive #: 8
Enter board drive #: 9
Enter userdb drive #: 10
Enter BBS prompt: BBS>
Login timeout (seconds): 120
Session timeout (seconds): 360
```

Appena avrai risposto a tutte queste domande, ti verrà richiesto se confermare tali impostazioni oppure no. "Base data correct (y/n)? " Battendo 'y' confermerai i dati che verranno salvati. Se invece risponderai con 'n' occorrerà nuovamente reintrodurre tali impostazioni di base del BBS.

2...Voce "BBS board setup"

Le Boards ("tavole", "schede") sono i predecessori dei forum di oggi. È il posto dove le persone scrivono i loro post. Di seguito vedrai un esempio di output che mostra la creazione di una board. Si prega di essere pazienti, poiché la creazione dei file di dati potrebbe richiedere del tempo (specialmente quando si esegue **Contiki BBS** su hardware reale). Puoi aggiungere tutte le boards che vuoi finché hai a disposizione lo spazio su disco.

```
* BBS board editor
(a)add, (l)ist or (q)uit :a
Board #1
Board name : lounge
Access lvl.: 10
Board data correct (y/n)? y
```

È possibile utilizzare l'opzione (l) ist per stampare un elenco di tutte le schede disponibili. Seguendo l'esempio sopra, vedrai la scheda "lounge" elencata.

```
* BBS board editor (a)add, (l)ist or (q)uit :l
ID#|Acc|Boardname -----
001|010|lounge
```

Press key

3...Voce "TCP/IP setup"

Questo passo della configurazione permetterà di impostare il vostro hardware di rete per Commodore 64, incluso indirizzo IP, driver e locazione di memoria della cartuccia.

L'esempio mostra un tipico settaggio LAN impiegando una cartuccia RRnet oppure 64NIC+ all'indirizzo di memoria \$deo8. Entrambe le cartucce impiegano il driver cs8900a.eth driver.

I dispositivi di rete ETH64 possono usare il driver langc96.eth driver con il suo appropriato indirizzo di memoria.

NOTA: Non fare precedere l'indirizzo di memoria dal simbolo '\$'

Se la tua cartuccia ethernet risiede all'indirizzo \$deo8 devi impostare semplicemente 'deo8' quando ti verrà richiesta tale informazione.

ULTERIORE NOTA: La scheda, se emulata dentro VICE dovrà avere necessariamente un indirizzo IP DIVERSO da quello impostato da qualsiasi altro dispositivo di rete già presente, indirizzo di rete del PC ospite incluso ovviamente.

```
* BBS network setup
Host IP : 192.168.200.64
Netmask : 255.255.255.0
Gateway IP : 192.168.200.1
DNS IP : 192.168.200.1
Mem addr. ($deo8) : deo8
Driver (cs8900a.eth): cs8900a.eth
Write to drive # (8): 8
```

Network data correct (y/n)? y

4...Voce "User editor"

L'editor degli utenti è attualmente molto limitato. L'unica cosa che fa è aggiungere utenti al suo database, nient'altro. Al momento **Contiki BBS** non si preoccupa dei nomi utente, non fa nemmeno distinzioni tra utenti diversi. Questo cambierà nelle versioni future del programma. Per ora un singolo utente sarà tutto ciò di cui abbiamo bisogno. Nell'esempio seguente forniamo semplicemente un utente 'guest' con la password 'guest', così possiamo accedere al nostro nuovo BBS.

```
* BBS user editor
(a)add, (s)how, (l)ist or (q)uit :a
User # : 1
Username: guest
Password: guest
Access lvl.: 10
```

User data correct (y/n)? y

Abbiamo a questo punto configurato il nostro sistema **Contiki BBS** e siamo pronti per partire. Usciamo dal programma di configurazione del BBS dal menù principale scegliendo 'q'.

ESECUZIONE DI CONTIKI-BBS

- 1.) Carichiamo ed eseguiamo **Contiki BBS** impostando **LOAD "CONTIKI-BBS.C64",8,1**
- 2.) Battiamo: **RUN**
- 3.) Ora utilizziamo una seconda macchina per connetterci

adoperando l'indirizzo IP attribuito al Commodore 64 (emulato o meno). Tale indirizzo IP coinciderà con quello inserito nel passo 3 della configurazione di CONTIKI-BBS. Sfrutteremo quindi il protocollo telnet.

Seguendo il nostro esempio di configurazione, andiamo sulla seconda macchina, apriamo il prompt dei comandi e digitiamo:

telnet 192.168.200.64

Dovrebbe apparire qualcosa di simile a quanto segue:

```
Welcome to:
-----
>> Contiki BBS <<
Contiki BBS V0.3.0 (C) 2009-2018
by
N. Haedecke
```

Contiki BBS 0.3.0 login: _

Siamo ora connessi al nostro sistema **CONTIKI-BBS** e possiamo loggare inserendo il nome utente e la password specificata durante il passo 4 del programma di configurazione del BBS.

E' chiaro che tale connessione funzionerà **solo all'interno della Vostra LAN**.

Tuttavia, configurando opportunamente (**ed a Vostro rischio**) il router di casa potrete permettere anche connessioni esterne, provenienti da INTERNET.

WINPCAP e **VICE** servono solo nel caso in cui vogliate emulare il C64 tramite il Vostro PC.

Se così non fosse, basterà procurarvi una cartuccia "reale" RR-Net oppure 64NIC+ e far girare il software proposto, CONTIKI-BBS.

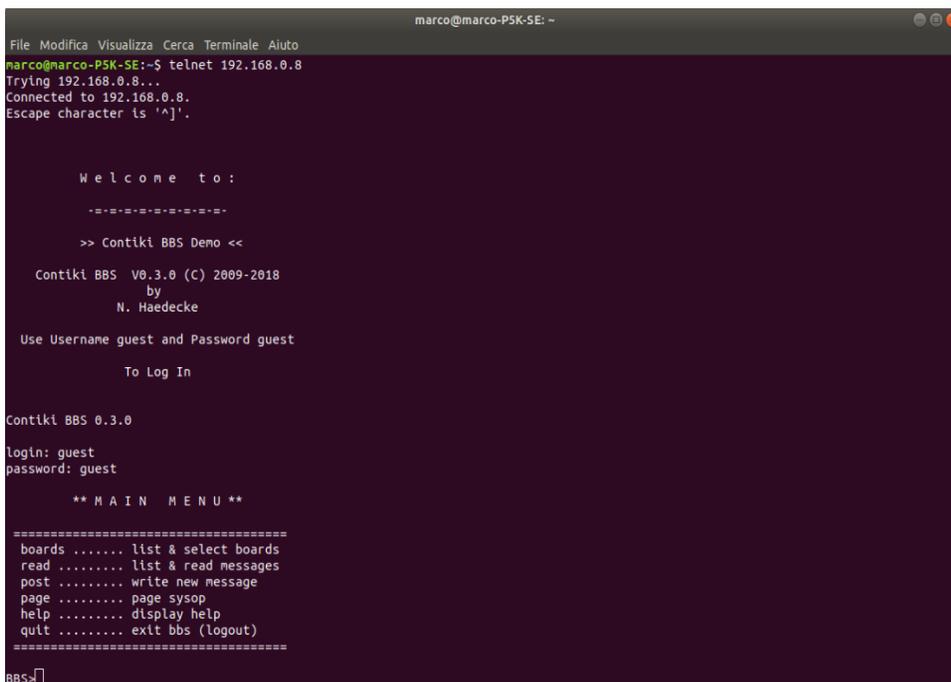
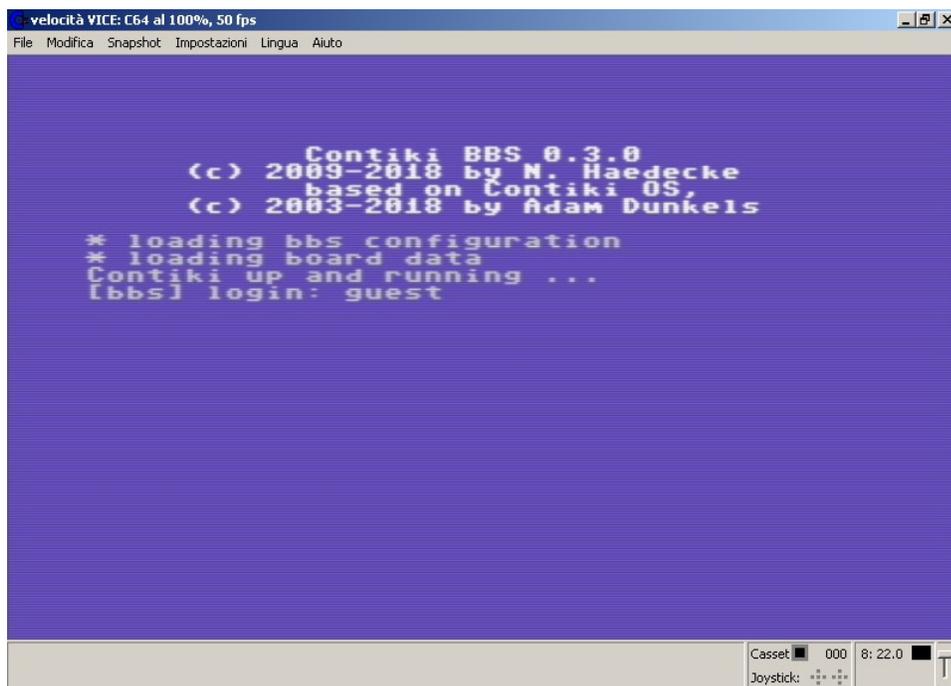
BEN FATTO, BUON DIVERTIMENTO!!!

Parte III – Commodore Mania BBS

La seconda scelta (che si è rivelata ben più onerosa del previsto) è stata l'impiego di un software BBS vecchio stile già esistente, ovvero **Commodore Mania BBS** (anno 1985), interfacciato tramite **VICE** e fruibile tramite protocollo **TELNET**.

Per maggiori info su **Commodore Mania BBS**:

<http://www.zimmers.net/bbs/cmbbs.html>



Per **scaricarlo** invece: <ftp://ftp.zimmers.net/pub/cbm/c64/comm/bbs/cmbbs60.zip>

L'interfacciamento viene gestito dal famigerato **TCPSE**R, software che Vi ho già presentato quando abbiamo scoperto come fruire dei BBS tramite **VICE** in un precedente articolo.

Purtroppo però, non appena ho provato a collegarmi al BBS via **TELNET**, sfruttando stavolta l'indirizzo IP della macchina dove gira **VICE** (ovvero la macchina ospite), leggo "strani" messaggi come questi:

ATMoS7=4oQ1EoS2=1So=1

PRESS RETURN

ma, tuttavia, non riesco ad interagire in alcun modo con il BBS.

A dirla tutta, il risultato è simile a quello che ottengo con diversi altri software analoghi. A volte leggo qualcosa durante il collegamento via **TELNET** ma non riesco ad interagire con il programma gestore del BBS in alcun modo.

Eppure, dal punto di vista teorico, il passaggio dei dati ricevuti via **TELNET** dovrebbe essere

gestito dal programma **TCPSER** bidirezionalmente verso l'interfaccia (emulata da **VICE**) rs-232. Da cosa nasce allora questo problema?

Il problema è già noto da diversi anni. E' presente in **VICE** un bug documentato all'indirizzo: <https://sourceforge.net/p/vice-emu/bugs/262/>

Si tratta della gestione del segnale "CD" ovvero "Carrier Detect" che tradotto letteralmente sta per "rilevazione di segnale portante"

Purtroppo **VICE** non si rende conto del fatto che sia in atto o meno una richiesta di connessione e, di conseguenza, non avviene alcun interscambio di dati da e verso il programma di gestione BBS.

Fu realizzata una "patch" non ufficiale, una correzione del software di **VICE** per gestire tale problema. La patch si trova all'indirizzo: <http://gcns.com/vice/vice.html>

ma è relativa ad una versione di **VICE** piuttosto obsoleta.

Si tratta di una patch destinata alla versione 1.19 di **VICE**, datata all'incirca **Gennaio 2006(!)**

Tale correzione non è stata mai inserita ufficialmente nel repository di **VICE**. Perché? Posso solo azzardare delle ipotesi.

La prima è che questa patch risolve un problema di collegamento ben preciso, che involve **VICE** ed il software **TCPSER** (e quindi non risolve il problema in tutti i possibili scenari d'uso)

La seconda ipotesi che posso formulare al riguardo è che la patch risolve solo in minima parte il problema di collegamento in quanto i meccanismi impiegati dai software per BBS su C64 per controllare il flusso dei dati sono di volta in volta diversi.

E nel frattempo cosa si può fare?

La soluzione potrebbe essere quella di sfruttare la patch segnalata, che però ci costringerebbe ad utilizzare la versione 1.19 di **VICE** (ovvero una versione che è stata sostituita da circa una trentina di rilasci fino ad oggi).

Grazie al fatto che tale patch è corredata delle modifiche apportate ai sorgenti, mi sono preso la briga di riallineare tali modifiche alla release attuale di **VICE**, la versione 3.2.

Nel corso degli anni il codice sorgente di **VICE** è cambiato parecchio, il codice è stato

riscritto, incapsulato, specializzato. Dove una volta esisteva una unica funzione, oggi ne esistono due o più, oppure la stessa funzione è stata soppressa oppure ancora incorporata all'interno di altre.

L'adattamento del codice della vecchia patch con gli odierni sorgenti di **VICE** non è stato quindi semplice, sebbene io sia stato "agevolato" perché non ho dovuto effettuare la correzione partendo da zero, tuttavia in qualche modo "l'innesto" è riuscito ed è in forza di questo risultato che ho potuto completare e verificare con esito positivo il collegamento tra **Commodore Mania BBS versione 6.0** (che gira sotto **VICE**) ed un utente che accede allo stesso tramite **TELNET**, e posso privilegiarmi di consegnare nelle Vostre mani il frutto di queste mie "fatiche" ☺

Il presente articolo sarà corredato di una cartella contenente gli eseguibili x64.exe ed x128.exe relativi a **VICE** versione 3.2 opportunamente modificati.

Per l'occasione ho "coniato" un nuovo numero di versione, 3.2.1

Non fate caso alla indicazione di versione "instabile" scegliendo l'apposita voce di menù "info su **VICE**" in quanto tale indicazione è legata esclusivamente al numero di versione che ho scelto di ottenere.

Gli eseguibili sono stati compilati su **Windows 10** e dovrebbero girare sia su sistemi 64 bit che su sistemi 32 bit. Anche su **Windows 10** comunque girerà a 32 bit. Non ho impostato l'ambiente di sviluppo per compilare **VICE** in modalità 64 bit.

Disclaimer

Né io, né "**RetroMagazine**", né tantomeno il team di **VICE** potremo fornirVi supporto e/o qualsivoglia garanzia di corretto funzionamento dell'emulatore **VICE** così modificato.

Vi chiedo di utilizzare tali eseguibili solo nel caso in cui vogliate effettuare dei test relativi al software presentato in questo articolo e farne quindi un uso limitato a questo unico scopo.

*Colgo l'occasione, nel contempo, per ringraziare (sia da parte mia che da parte di tutta la redazione di "RetroMagazine") tutti i membri, passati e presenti, del team di **VICE** per il loro immenso, straordinario lavoro, estremamente utile a tutti noi retro-appassionati del mondo Commodore.*

E' verosimile l'ipotesi che altri software BBS possano funzionare in virtù di questa versione "patchata" di **VICE**.

Tuttavia ho avuto la possibilità di testare in maniera un po' più approfondita solo "**Commodore Mania BBS 6.0**" (che chiamerò per brevità **CMBBS 6.0**) e pertanto, da qui in poi, mi riferirò esclusivamente a questo specifico software.

Esaminiamo in maniera puntuale il da farsi per poter sfruttare **CMBBS 6.0** tramite **VICE** via **TELNET**.



Materiale occorrente

- 1) **TCPSER**;
- 2) **cygwin1.dll** (libreria a collegamento dinamico necessaria al funzionamento di **TCPSER**);
- 3) file batch **go.bat** per avviare **TCPSER** con i necessari parametri;
- 4) **VICE x64** Versione 3.2.1;
- 5) Programma **CMBBS 6.0**;

Troverete tali elementi (tranne **CMBBS 6.0** da scaricare da link apposito) già inclusi all'interno di un file compresso, per Vostra maggiore comodità.

Procedura per **VICE** su **Windows**

a) scaricare e scompattare la cartella contenente il materiale messo a disposizione da "**RetroMagazine**" in apposita cartella, esempio: "**c:\vice**"

b) Su **VICE**, menù impostazioni, impostazioni RS-232, "Impostazioni ACIA..." Spuntare "Attiva emulazione interfaccia ACIA RS232"

Dispositivo ACIA: **Dispositivo RS232 #1**
Posizione ACIA: **\$DE00**

Interrupt ACIA: **NMI**

Modalità ACIA: **Normale**

Sempre dal menù impostazioni, impostazioni RS-232, "Impostazioni RS-232 su userport..."



Spuntare "Attiva l'emulazione della RS232 su"

Dispositivo su userport: **Dispositivo RS232 #1**

Velocità userport: **300**



Ed ancora, dal menù impostazioni, impostazioni RS-232, "Impostazioni RS-232..."

Dispositivo RS232 #1: **127.0.0.1:25232**

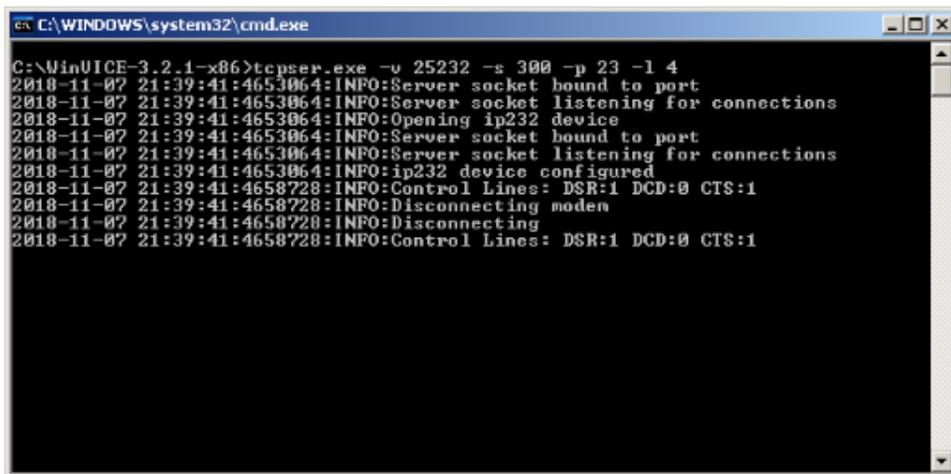
Dispositivo RS232 #2: **127.0.0.1:25232**

Dispositivo RS232 #3: **127.0.0.1:25232**

Dispositivo RS232 #4: **127.0.0.1:25232**



c) Lanciare il file batch "go.bat" presente nella cartella contenente tutto il materiale e **NON CHIUDERE LA SUA FINESTRA DI LAVORO!**



d) Su **VICE**, da "Avvia automaticamente immagine disco/cassetta..." scegliere la cartella contenente i due files relativi a CMBBS e fare doppio click sul file programma "BOOT BBS", contenuto nel file `cmbbs60-prg.d64`

e) Scegliere se l'ora del BBS sarà AM oppure PM (battere "a" oppure "p")

f) Impostare l'ora corrente del BBS (esempio digitare "04") e battere invio

g) Impostare i minuti relativi all'ora corrente del BBS (esempio digitare "30") e battere invio

h) Alla domanda "Load previous CONFIGURE file? [y/n]" digitare "y"

i) Attendere per qualche secondo. Alla richiesta "Insert DATA disk(s) and hit return?", dal menù File di **VICE**, "Seleziona immagine disco", "drive 8" scegliere il file "cmbbs60-data.d64" e battere invio

j) segnare l'indirizzo IP del pc dove gira **VICE**, digitando da **Prompt di DOS** il comando "ipconfig"

k) da una seconda macchina ove è installato il programma **TELNET** digitare da prompt "telnet xxx.xxx.xxx.xxx", sostituendo al posto di xxx.xxx.xxx.xxx l'indirizzo IP della macchina dove gira **VICE** con **CMBBS 6.0**

l) Battete per un paio di volte il tasto INVIO. Se avete fatto tutto correttamente, sarete in collegamento **TELNET** con il BBS che gira su **VICE** e potrete interagire con esso!

BEN FATTO, E BUON DIVERTIMENTO ANCHE STAVOLTA!!!

Vale anche qui la stessa considerazione già fatta in precedenza in merito a **Contiki BBS** ovvero che, anche per **CMBBS 6.0**, la configurazione vista funzionerà **solo all'interno della Vostra LAN**.

Tuttavia, configurando opportunamente (ed a **Vostro rischio**) il router di casa potrete permettere anche connessioni esterne, provenienti da **INTERNET**.

Contiki BBS sarà perfettamente fruibile quando girerà su **VICE** ed accederete allo stesso via **TELNET**.

CMBBS 6.0 gira su **VICE** ma non sarà possibile interagire con esso tramite **TELNET**, a meno che non utilizzate la versione **NON UFFICIALE 3.2.1** di **VICE** fornita da "RetroMagazine"

Procedura per l'uso di CMBBS 6.0 con VICE 64 fruibile via TELNET e Linux Ubuntu 18.04 LTS

Di seguito trovate i passi necessari nel caso in cui adoperiate Linux Ubuntu 18.04 LTS per i Vostri test.

1) Aprire la finestra terminale e Installare **TCPSER** (se non già installato) digitando:
sudo apt-get install tcpser

2) Installare **NETCAT** (se non già installato) digitando ancora:
sudo apt-get install netcat

3) All'interno della stessa finestra digitare:
nc 127.0.0.1 25232

4) Infine, ancora da terminale digitare:
sudo tcpser.exe -v 25232 -s 300 -p 23 -l 4
e lasciare questa finestra **APERTA**

5) Aprire **VICE** versione **3.2.1** (fornito da "RetroMagazine") e dal menù Impostazioni, Impostazioni RS-232 togliere la spunta su: "Emulazione interfaccia ACIA RS232", mettere la spunta su "Emulazione RS-232 su userport", impostare "Velocità RS-232 su userport" al valore **300**, impostare "Dispositivo RS232 su userport" a "Esegui processo" ed infine digitare alla voce di menù "Nome del programma da eseguire:" la seguente impostazione:
| nc localhost 25232

6) procedere poi con il caricamento del B.B.S. esattamente come già descritto relativamente all'uso di **VICE** sotto Windows

7) per testare il collegamento da altra finestra terminale digitare: **telnet 127.0.0.1** e battere qualche invio finché non si vede scorrere del testo all'interno della finestra.

Cosa è stato modificato su VICE?

La modifica essenzialmente si può descrivere in questi termini:
viene controllato il flusso dei dati ricevuti e, in presenza di un particolare carattere di controllo (specifico di **TCPSER**) presente all'interno di tale flusso, viene impostato a Vero il "Carrier Detect".
E' chiaro che tale modifica ha un raggio di azione molto modesto, ed è pensata per l'utilizzo congiunto di **VICE** e **TCPSER** e risulta pertanto inefficace con altri eventuali interlocutori.

La modifica del codice di **VICE** è stata effettuata in maniera "chirurgica", mantenendo quindi la quasi totalità del suo codice così come rilasciato ufficialmente dal team che lo sviluppa.

Servirebbe un approccio più serio, orientato alla completa risoluzione della problematica.

L'emulazione dovrebbe permettere di simulare la presenza di un modem collegato alla interfaccia rs-232 su userport, e gestire ovviamente tutto il relativo flusso di dati. Trattasi ovviamente di uno sforzo ben diverso da quello profuso per ottenere questa semplice patch.

Mi auguro che possa comunque servire da "sprono" al team di **VICE** per portare avanti lo sviluppo dell' emulatore anche su questo fronte, rendendolo quindi ancora più completo.

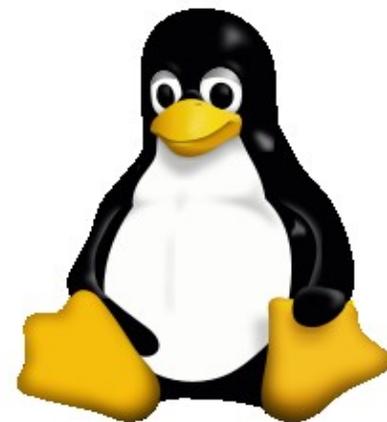
Per ottenere gli eseguibili di di VICE 3.2.1 per Windows:

<http://www.retromagazine.net/getrm.php?id=vicebin>

Per ottenere i sorgenti di questa versione non ufficiale di VICE:

<http://www.retromagazine.net/getrm.php?id=vicesrc>

Nota: E' presente nella stessa cartelle un mini-howto per compilare VICE ed ottenere i relativi eseguibili per Linux Ubuntu 18.04 LTS.



PERSONALIZZARE CMBSS 6.o

CMBBS può essere configurato e/o personalizzato senza particolari difficoltà, pur essendo già utilizzabile subito dopo essere stato scaricato e scompattato.

La configurazione di un BBS è una operazione delicata che richiede attenzione ed un pizzico di esperienza.

Per informazioni più dettagliate riguardo a **CMBBS** ed alle sue funzionalità, riporto alcuni utili links:

<http://www.zimmers.net/bbs/cmbbs.html>

<http://www.zimmers.net/bbs/docs/cmbbs.txt>

Nel caso vogliate realizzare una nuova configurazione di **CMBBS**, potrete lanciare il programma "SYSTEM CREATE" contenuto nel file **cmbbs6o-prg.d64**

E' possibile anche **NON** utilizzare il file di configurazione già esistente. In questo caso basta rispondere "n" alla domanda "Load previous CONFIGURE file? [y/n]" (vedi punto h della Procedura per VICE su Windows) e rispondere alle domande successive relative

al tipo di modem utilizzato, al livello di sicurezza richiesto per caricare e scaricare files, tempo limite (in secondi), se saranno ammessi o meno utenti "guest" ovvero ospiti (cioè non registrati), se sarà possibile o meno registrare nuovi utenti che si collegheranno al BBS, se gli utenti "guest" (ovvero ospiti) potranno mandare feedback oppure no, se il messaggio di benvenuto potrà essere saltato o meno, se sarà disponibile la funzionalità "auto private uploads", se il sistema BBS utilizzerà i "crediti", quale sarà il massimo livello di sicurezza che potrà essere impiegato dal sistema di crediti, quale sarà il numero di "crediti" per ciascun post/upload ed infine quale sarà il livello di sicurezza necessario per validare nuovi utenti.

Dopo aver risposto a tutte le domande elencate precedentemente, potrete scegliere se salvare tali impostazioni nel file di configurazione del BBS oppure no.

Nel secondo caso il sistema adopererà l'ultima versione del file di configurazione del BBS.

La nostra chiacchierata relativa ai B.B.S. che girano su C64 con VICE, fruibili tramite TELNET volge al termine.

Con hardware e software di nuova concezione tutto è più semplice.

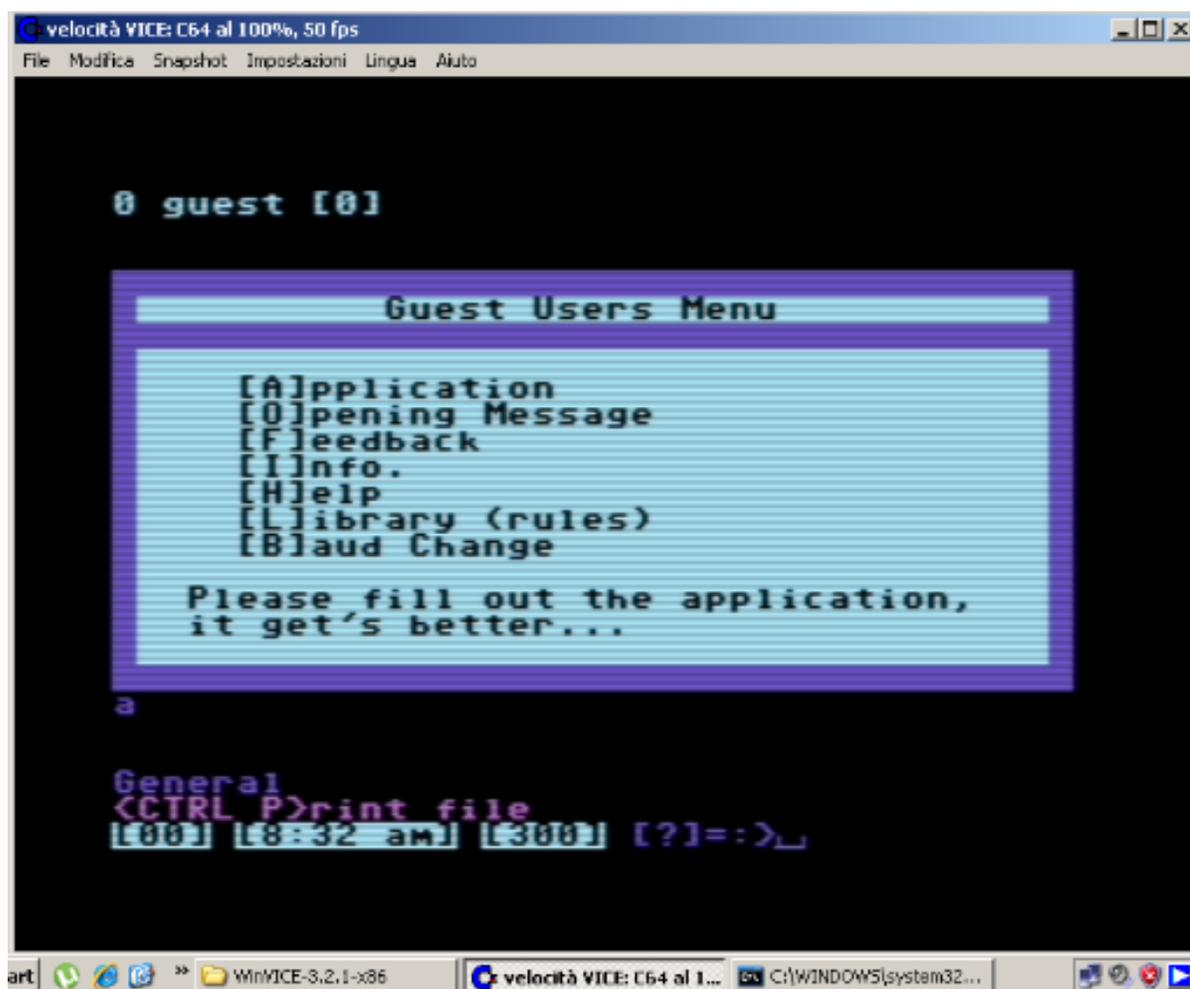
Quando invece abbiamo provato ad impiegare allo scopo un software di vecchia generazione sono iniziati i guai ma... non ci siamo dati per vinti.

RetroMagazine non ha esitato a mettere in piedi una patch (non ufficiale) di VICE per far provare ai Suoi lettori "l'ebrezza" di usare un vecchio software per B.B.S. "old-style" sfruttando nel contempo le tecnologie attuali.

Siamo riusciti a stupirvi? Speriamo di si!

Alla prossima, amici lettori!

Un caloroso ringraziamento da parte mia ai colleghi redattori **Dante Profeta** per i test di collegamento effettuati via TELNET da remoto ai BBS e a **Giorgio Balestrieri** per il supporto ai test in ambiente Linux + varie ☺



Esplorando l'Amiga - parte 3

di Leonardo Giordani

La jump table delle librerie

La libreria Exec è la libreria di base del sistema Amiga. Questa libreria è caricata in memoria al momento dell'accensione ed è sempre disponibile in memoria. Una volta caricata, ha la stessa struttura di qualsiasi altra libreria, per cui troveremo un preambolo contenente la jump table in ordine inverso e a seguire il codice effettivo della libreria.

Il trucco sta nel fatto che Exec è la libreria usata per caricare in memoria altre librerie, quindi la funzione che crea in memoria la struttura di una determinata libreria è contenuta qui. Per installare Exec in memoria abbiamo bisogno di usare una funzione che è parte di Exec stessa.

Questo è uno dei punti di forza del linguaggio Assembly, ovvero la capacità di trattare il codice come se fossero dati, e che è chiamata "omoiconicità". Questa proprietà si trova raramente nei linguaggi di programmazione, essendo connessa sostanzialmente alla mancanza dei livelli sopra la rappresentazione ad albero della sintassi (AST). Lisp è un ottimo esempio di linguaggio omoiconico.

Tornando alla tabella dei vettori, dobbiamo trovare un modo per usare la libreria Exec per installare in memoria Exec stessa. Il concetto non è particolarmente complesso, se lo si rappresenta con il seguente pseudocodice

```
vectors:
    function1-vectors
    function2-vectors
    function3-vectors
function1:
    code
    [...]
function2:
    code
    [...]
function3:
    code
    [...]
```

In questa situazione abbiamo 3 funzioni definite agli indirizzi 'function1', 'function2' e 'function3'. Da qualche parte nel codice, all'indirizzo 'vectors', c'è una lista che contiene gli indirizzi di queste funzioni. Siccome il codice può essere spostato in memoria questa lista contiene degli offset relativi alla tabella 'vectors' stessa, quindi il primo elemento della tabella sarà 'function1 - vectors', ovvero la sottrazione tra i due indirizzi, seguita da 'function2 - vectors' e così via.

per esempio potremmo avere

```
0042 0122
0044 01b8
0046 02d1
[...]
0164 code of function1
[...]
01fa code of function2
[...]
0313 code of function3
[...]
```

Dove il contenuto della tabella è $0x164 - 0x42 = 0x122$, $0x1fa - 0x42 = 0x1b8$ e $0x313 - 0x42 = 0x2d1$.

La tabella dei vettori, quindi, è la sorgente a partire dalla quale possiamo calcolare la jump table. Il codice per eseguire questa operazione, però, è contenuto in una delle funzioni stesse, per esempio la numero 2

```
function1:
    code
    [...]
function2:
    for each address of <table>
    create
        a jump table entry relative to
        <start>
function3:
    code
    [...]
vectors:
    function1-vectors
    function2-vectors
    function3-vectors
```

Come potete vedere la funzione all'indirizzo function2 (in questo esempio) dipende da due parametri <table> e <start> che saranno contenuti in qualche registro. A questo punto, siccome l'indirizzo di function2 è noto, ci sarà del codice che esegue la funzione sulla tabella che contiene il codice stesso

```
setup:
    run <function2> on
    <vectors_offset> and <setup>
function1:
    code
    [...]
function2:
    for each address of the vector
    table create
        a jump table entry relative to
        START
function3:
    code
    [...]
vectors:
    function1-vectors
    function2-vectors
    function3-vectors
```

dove <vectors_offset> è un offset fisso nel codice (visto che la distanza relativa tra vectors e <setup> è nota) e setup è l'indirizzo effettivo che la routine setup ha a runtime.

Questo meccanismo crea quindi una libreria che può installare altre librerie attraverso una specifica funzione, che può allo stesso tempo installare sé stessa.

La tabella dei vettori di Kickstart

Un esempio pratico del meccanismo dietro la tabella dei vettori può essere trovato nel codice di Kickstart. Questo è il BIOS dell'Amiga, e viene caricato all'accensione da disco (Amiga 1000 e alcuni Amiga 3000) o da una ROM.

Il codice di Kickstart 1.3 può essere trovato all'indirizzo

<https://www.romcollector.com/emulators-firmware-i-29.html> e disassemblato con vdsasm

```
vda68k Kickstart1.3.rom >
Kickstart1.3.asm
```

Nel codice troviamo un'implementazione pratica di quanto descritto precedentemente.

Disclaimer obbligatorio: per usare il Kickstart dell'Amiga dovete possedere una licenza. Questa rivista è contraria alla pirateria di sistemi ormai in disuso.

Nota sul disclaimer: questa frase ironica mi ha procurato alcuni litigi su Reddit, il che dimostra che la gente non ha il senso dell'umorismo (oppure non ha nulla da fare). Ad ogni modo, per dirla chiaramente, personalmente ritengo assurdo che un sistema ormai commercialmente fallito e morto sia ancora soggetto al copyright. Fine della nota ironica.

Quando si disassembla il codice binario ovviamente non si ottiene un codice sorgente scritto in un linguaggio ad alto livello. Beh, non son un disassembler semplice come vdsasm, ad ogni modo. Quello che otterrete è l'interpretazione di ogni singolo valore binario secondo le convenzioni del microprocessore usato, e questo include parti del file binario che sono solo dati. Il disassembler non ha modo di sapere se la rappresentazione binaria che sta elaborando rappresenta delle vere e proprie istruzioni o semplicemente dei

numeri. Inoltre, non c'è traccia delle etichette (label) usate dagli autori del codice, in quanto esse sono state perse nella "compilazione" in linguaggio macchina, dove sono state sostituite da puri indirizzi.

Il processo per cui si capisce come un sistema funzioni partendo dalla sua implementazione è detto "reversing" (rovesciare, rovesciamento), e personalmente la considero una delle attività più affascinanti che un programmatore può affrontare.

Lo scopo della seguente ricerca è di trovare la tabella dei vettori di Kickstart 1.3, per poi trovare grazie a quella la posizione e implementazione delle funzioni di Exec. Si parte!

Fase 1

È noto che MakeFunctions viene usata per creare in memoria la struttura di Exec stessa. Pertanto tale funzione deve venir chiamata da qualche parte nel codice che sto esaminando.

Siccome uno dei parametri della routine MakeFunctions è il nome della libreria un buon punto di partenza può essere una stringa contenente exec.library (che è il nome standard di questa libreria nel sistema Amiga). Una volta trovata quella stringa potrà cercare una chiamata di funzione che usa l'indirizzo di questa stringa come parametro.

La sequenza di byte che rappresenta quella stringa in ASCII è 65 78 65 63 2E 6C 69 62 72 61 72 79. In Kickstart 1.3 l'offset di questa sequenza di byte è 0x00a8.

Ricordate che quello che vedete nel codice disassemblato non è una stringa. Il disassembler prova a convertire tutto in istruzioni, quindi quello che avrete è

```
000000a8: 6578
bcs.b 0x122
000000aa: 6563
bcs.b 0x10f
000000ac: 2e6c 6962
movea.l 0x6962(a4),sp
000000b0: 7261
moveq #0x61,d1
000000b2: 7279
moveq #0x79,d1
```

Quando si ricercano stringhe è meglio utilizzare un editor esadecimale che può mostrare la rappresentazione ASCII del codice binario e ricercare stringhe dentro quest'ultima (Figura 1).

Sappiamo inoltre che Kickstart viene caricato all'indirizzo 0xfc0000 (Amiga System Programmer's Guide, pagina 67), quindi tutti gli indirizzi a 16 bit sono relativi a 0x00fc. Il

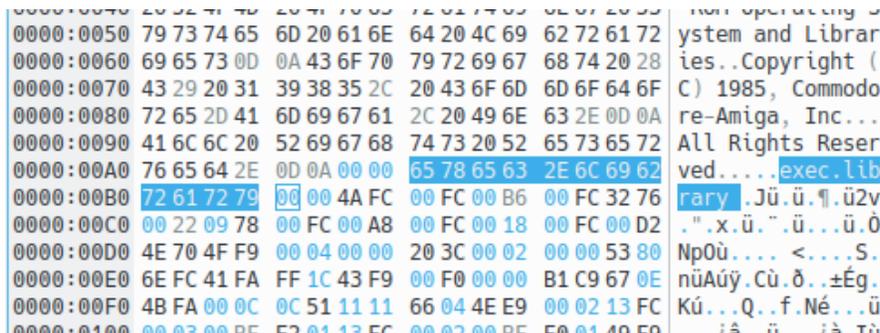


Figura 1 - Editor esadecimale che mostra la rappresentazione ASCII del codice binario.

puntatore al nome della libreria è quindi 00fc00a8.

Fase 2

Nel sistema Amiga tutte le librerie hanno una struttura specifica quando vengono caricate in memoria. Prima di tutto tutte le librerie in memoria sono nodi di una linked list, quindi ci aspettiamo di trovare la struttura di un nodo. Poi internamente a questo ci aspettiamo di trovare la struttura della libreria stessa.

Il file include_i/exec/nodes.i ci mostra la struttura di un nodo in una linked list standard

```
STRUCTURE LN,0 ; List
Node
  APTR LN_SUCC ; Pointer to
  next (successor)
  APTR LN_PRED ; Pointer to
  previous (predecessor)
  UBYTE LN_TYPE
  BYTE LN_PRI ; Priority,
  for sorting
  APTR LN_NAME ; ID string,
  null terminated
  LABEL LN_SIZE ; Note: word
  aligned
```

I due puntatori a 32 bit LN_SUCC e LN_PRED sono creati quando il nodo viene caricato in memoria, per cui dobbiamo concentrarci sul resto della struttura, in particolare 1 byte LN_TYPE, 1 byte LN_PRI e 4 byte LN_NAME. Dallo stesso file include_i/exec/nodes.i veniamo a sapere che il tipo di nodo di una libreria è 09

```
NT_LIBRARY EQU 9
```

Pertanto il pattern che stiamo cercando è 09XX 00fc 00a8, rispettivamente il tipo di nodo (09), una priorità sconosciuta (XX), e il puntatore al nome della libreria 00fc 00a8. Sappiamo inoltre che tale pattern è probabilmente salvato vero l'inizio della ROM, in quanto una delle prime cose che il BIOS deve fare è caricare la propria struttura in memoria. Quest'ultima assunzione non deve essere presa per vera a prescindere, ma è sufficientemente ragionevole.

Nel codice di Kickstart 1.3 questo pattern si trova all'offset 0x030c (Figura 2).

Se questa è la posizione corretta della struttura nodo ci aspettiamo di trovare subito dopo la struttura della libreria come descritta nel file include_i/exec/libraries.i

```
STRUCTURE LIB,LN_SIZE
  UBYTE LIB_FLAGS ;
  see below
  UBYTE LIB_pad ; must
  be zero
  UWORD LIB_NEGSIZE ;
  number of bytes before LIB
  UWORD LIB_POSSIZE ;
  number of bytes after LIB
  UWORD LIB_VERSION ;
  major
  UWORD LIB_REVISION ;
  minor
  APTR LIB_IDSTRING ;
  ASCII identification
  ULONG LIB_SUM ; the
  system-calculated checksum
  UWORD LIB_OPENCNT ;
  number of current opens
  LABEL LIB_SIZE ;Warning:
  Size is not a longword multiple!
```

Il codice binario di Kickstart 1.3 dall'indirizzo 0xfc030c, infatti, è il seguente

```
0000030c: 09 ; LN_TYPE
0000030d: 00 ; LN_PRI
0000030e: 00fc 00a8 ; LN_NAME
00000312: 06 ; LIB_FLAGS
00000313: 00 ; LIB_pad
00000314: 0000 ;
LIB_NEGSIZE
00000316: 024c ;
LIB_POSSIZE
00000318: 0022 ;
LIB_VERSION
0000031a: 0002 ;
LIB_REVISION
0000031c: 00fc 0018 ;
LIB_IDSTRING
00000320: 0000 0000 ; LIB_SUM
00000324: 0001 ;
LIB_OPENCNT
```

Da questo si può vedere che la versione di exec contenuta in questa versione di Kickstart è 34 (0x22) revisione 2 (0x02), e

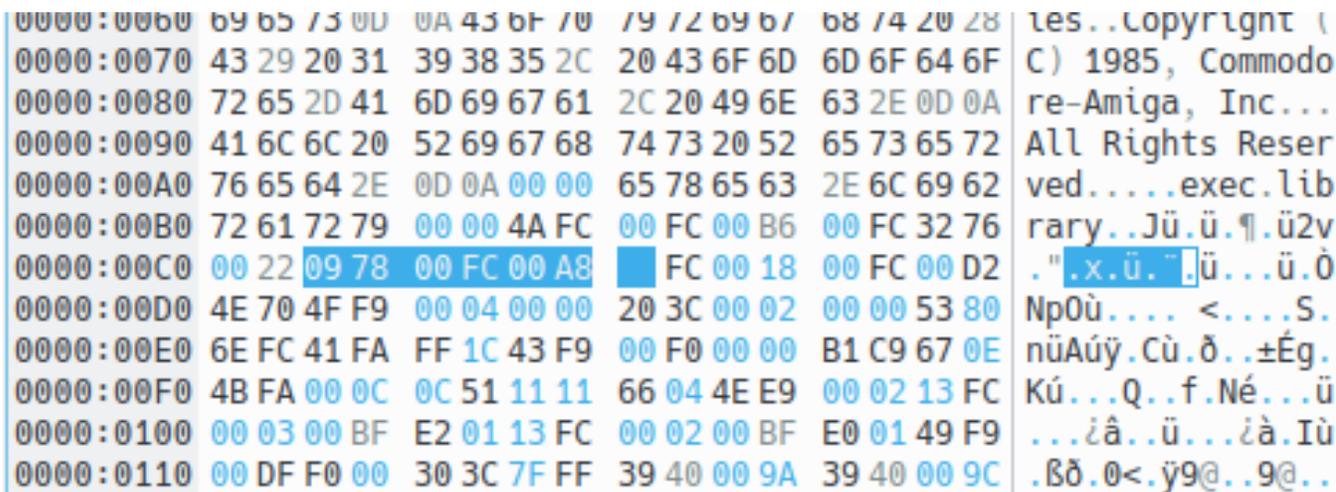


Figura 2 – Search Pattern string.

questo è confermato dall'ID all'indirizzo `0xfc0018`, che è `exec 34.2 (28 Oct 1987)` (Figura 3).

Fase 3

Quello a cui siamo effettivamente interessati, a questo punto, è dove l'indirizzo di questa struttura sia citato nel codice, visto che verrà usato per creare la struttura della libreria. Dal momento che la funzione `MakeFunctions` verrà chiamata dopo aver creato la struttura potremo ricostruire da questo dove essa viene definita.

La struttura è all'indirizzo `0x030c` e stiamo cercando un'istruzione simile a `lea 0x30c(pc), ax`, dove `ax` è uno dei registri di indirizzo `a0-a7`. Caricare l'indirizzo di una tabella in un registro è il metodo standard per effettuare dei cicli sulla tabella stessa al fine di modificare o copiare i byte da qualche parte.

Il Motorola 68000 non permette di eseguire un'istruzione `MOVE` con una destinazione relativa al Program Counter (PC). Nelle intenzioni dei progettisti del 68000, il codice non dovrebbe modificare sé stesso, infatti. Quindi, se dovete cambiare una tabella nel mezzo del codice, dovete puntare ad essa con un'istruzione `LEA TABLE(PC), An` e quindi modificare l'indirizzo attraverso `An`. Codice che si automodifica è pericoloso specialmente per programmi del 68000 che dovranno un giorno girare sul 68020, in quanto la cache delle istruzioni di quest'ultimo normalmente assume che il codice sia puro. (Da <http://www.easy68k.com/paulrsm/doc/trick68k.htm>, traduzione dell'autore)

All'indirizzo `0x0364` troviamo il seguente codice

```
0360: 43ee 0008
lea    0x8(a6), a1
```

```
0364: 41fa ffa6
lea    0x30c(pc), a0
0368: 700c
moveq  #0xc, d0
036a: 32d8
move.w (a0)+, (a1)+
036c: 51c8 fffc
dbf    d0, 0x36a
```

che installa la libreria `Exec` in memoria. Analizziamolo istruzione per istruzione.

L'indirizzo `ExecBase` è contenuto in `a6` (questo viene fatto precedentemente nel codice); questo indirizzo è incrementato di 8 e il risultato è copiato nel registro `a1`. Gli 8 byte lasciano spazio per i puntatori `LN_SUCC` e `LN_PRED`. Il codice poi carica l'indirizzo della tabella in `a0`.

Il ciclo viene eseguito su 26 byte. Il numero `12 (0xc)` viene copiato in `d0`, ma l'istruzione `dbf` (`dbra` in alcuni assembler) continua a saltare a `0x36a` fino a che il valore di `d0` è `-1`, cosicché il ciclo viene effettivamente eseguito 13 volte. L'istruzione `move.w` sposta words (2 byte), per cui stiamo copiando 26 byte, che è esattamente la dimensione del nodo di una libreria da `LN_TYPE` a `LIB_OPENCNT` inclusi.

Le successive 5 istruzioni sono

```
0370: 204e
movea.l a6, a0
0372: 43fa 1708
lea    0x1a7c(pc), a1
0376: 2449
movea.l a1, a2
0378: 6100 1238
bsr.w  0x15b2
037c: 3d40 0010
move.w d0, 0x10(a6)
```

Dalla [documentazione \(http://amigadev.elowar.com/read/ADCD_2_1/Includes_and_Autodocs_3_guide/node021_A.html\)](http://amigadev.elowar.com/read/ADCD_2_1/Includes_and_Autodocs_3_guide/node021_A.html) sappiamo che `MakeFunctions` ha il seguente prototipo

```
size = MakeFunctions(address,
                      vectors, offset)
d0                                a0      a1
a2
```

dove `address` è l'indirizzo dove la jump table verrà costruita, `vectors` è una tabella che elenca gli indirizzi delle funzioni (quella che stiamo cercando) e `offset` comunica alla funzione se gli indirizzi sono assoluti (valore 0) o relativi (nel qual caso `offset` è la base per il dislocamento). La lista degli indirizzi deve terminare con `-1 (0xffff)`.

La prima riga, quindi, salva in `a0` il contenuto di `a6`, che è l'indirizzo di `ExecBase`. Questo è l'indirizzo dove vogliamo installare la libreria. La seconda riga carica l'indirizzo della tabella dei vettori in `a1` e lo stesso valore è salvato in `a2`. Infine il codice salta alla subroutine all'indirizzo `0x15b2` che a questo punto sappiamo essere l'indirizzo di `MakeFunctions`.

Fase 4

Abbiamo estratto molte informazioni estremamente utili da questo codice. In primo luogo la tabella dei vettori è all'indirizzo `0x1a7c`, e in seconda battuta abbiamo scoperto l'indirizzo della subroutine `MakeFunctions`, che è `0x15b2`. Quest'ultimo valore sarà utile per fare un doppio controllo del contenuto della tabella dei vettori.

Dopo che `MakeFunctions` è stata eseguita il codice ritorna alla chiamata e l'istruzione successiva salva la dimensione finale della jump table 16 byte dopo l'indirizzo contenuto in `a6`. Con l'aiuto della struttura mostrata precedentemente sappiamo che a quell'indirizzo troveremo il campo `LIB_NEGSIZE`, che contiene la dimensione della jump table stessa, ovvero il numero di byte prima della libreria.

```

0000:0000 11 11 4E F9 00 FC 00 D2 00 00 FF FF 00 22 00 05 ..Nù.ü.ò..ÿÿ.."..
0000:0010 00 22 00 02 FF FF FF FF 65 78 65 63 20 33 34 2E ..ÿÿÿÿexec 34.
0000:0020 32 20 28 32 38 20 4F 63 74 20 31 39 38 37 29 0D 2 (28 Oct 1987)
0000:0030 0A 00 00 00 FF FF FF FF 0D 0A 0A 41 4D 49 47 41 ....ÿÿÿÿ...AMIGA
0000:0040 20 52 4F 4D 20 4F 70 65 72 61 74 69 6E 67 20 53 ROM Operating S
0000:0050 79 73 74 65 6D 20 61 6E 64 20 4C 69 62 72 61 72 ystem and Librar
0000:0060 69 65 73 0D 0A 43 6F 70 79 72 69 67 68 74 20 28 ies..Copyright (
0000:0070 43 29 20 31 39 38 35 2C 20 43 6F 6D 6D 6F 64 6F C) 1985, Commodo
0000:0080 72 65 2D 41 6D 69 67 61 2C 20 49 6E 63 2E 0D 0A re-Amiga, Inc...
0000:0090 41 6C 6C 20 52 69 67 68 74 73 20 52 65 73 65 72 All Rights Reser
0000:00A0 76 65 64 2E 0D 0A 00 00 65 78 65 63 2E 6C 69 62 ved....exec.lib
0000:00B0 72 61 72 79 00 00 4A FC 00 FC 00 B6 00 FC 32 76 rary..Jü.ü.¶.ü2v

```

Figura 3 – Versione di exec contenuta nel Kickstart 34 - exec 34.2 (28 Oct 1987)

Dobbiamo ora controllare se ciò che abbiamo trovato è corretto. Ci dovrebbe essere una tabella all'indirizzo `0x1a7c` che contiene gli indirizzi delle funzioni nell'ordine fornito dal file `include_i/exec/exec_lib.i`. Siccome `MakeFunctions` in quel file è citata all'undicesimo posto possiamo controllare che il contenuto della tabella sia consistente. Secondo il codice precedentemente mostrato, quell'indirizzo dovrebbe puntare alla funzione salvata all'indirizzo `0x15b2`.

I valori a `0x1a7c` sono i seguenti

```

00001a7c: 08a0
00001a7e: 08a8
00001a80: 08ac
00001a82: 08ac
00001a84: ee6a
00001a86: f420
00001a88: f446
00001a8a: 04f8
00001a8c: f4a0
00001a8e: f4ea
00001a90: f58e
00001a92: f0b0
00001a94: f188
00001a96: faac
00001a98: fb36
00001a9a: f080
[...]
```

Il file `include_i/exec/exec_lib.i` non contiene i primi 4 vettori riservati (le funzioni `Open`, `Close`, `Expunge` e lo spazio riservato), quindi considerando che essi sono nella tabella dovremo leggere il 15esimo vettore, dove troviamo `0xfb36`. Questo è un offset relativo all'inizio della tabella stessa, per cui la funzione `MakeFunctions` è all'indirizzo `0x1a7c + 0xfb36 = 0x15b2` (gli indirizzi sono a 16 bit), che coincide con quanto già scoperto.

Questo mostra che la procedura che abbiamo seguito è corretta. La tabella dei vettori di Kickstart 1.3 è all'indirizzo `0x1a7c` e da lì

possiamo raggiungere ed analizzare tutte le funzioni contenute nella libreria di base dell'Amiga.

Prossimi articoli...

Nel prossimo articolo parleremo dei vettori riservati, di `MakeFunctions` e del codice automodificante contenuto nel BIOS dell'Amiga.

*** Note for the English readers ***

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

<http://blog.thedigitalcatonline.com/blog/2018/06/08/exploring-the-amiga-3/>

Risorse

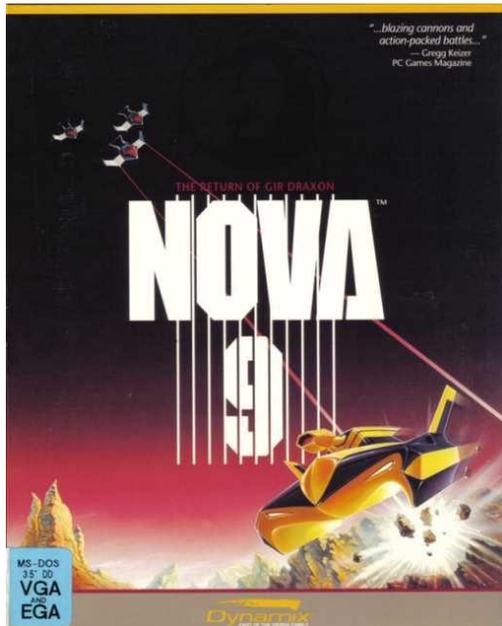
Amiga System Programmers Guide, Abacus:
https://archive.org/details/Amiga_System_Programmers_Guide_1988_Abacus

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

Amiga Forever package sold by Cloanto
www.amigaforever.com



NOVA9: IL RITORNO DI GIR DRAXON



GIUDIZIO SUL GIOCO

GIOCABILITA'

80%

Sebbene Novag sia un parente prossimo di Battlezone. Sebbene sia diretto erede di Stellar7. Siamo arrivati ad un videogioco che sfrutta la traslitterazione della grafica vettoriale su nuovissime schede VGA in ambiente DOS. Il tripudio di colori soddisferà i tanti giocatori. Meno anima di gioco, stessa trama, stessi obiettivi ma molti più colori, più grafica, più effetti speciali. Tutto ciò però non può incrementare in modo spiccato il proprio valore, sebbene io abbia giocato con estrema passione a questo titolo degli anni 90. Il gioco è stato premiato con un bel 4,5 su 5 nella rivista americana Dragon. Addirittura la rivista Computer Gaming World, nel mese di Aprile del 1992, ha definito "mozzafiato" la grafica e ha elogiato il gioco definendolo "un'eccellente scelta per gli appassionati di sale giochi di tutto il mondo, grazie ad uno sforzo di programmazione assolutamente professionale".

LONGEVITA'

70%

Sostanzialmente, sono veramente pochi i miglioramenti di trama e le caratteristiche intrinseche del gameplay rispetto a Stellar7. Ricordiamo però che questo passaggio a grafiche fantascientifiche ha fatto sognare i fanciulli del tempo. Suoli esotici, anzi, marziani, macchè, al di fuori del sistema solare. No, in realtà ben di più, perché quei colori facevano uscire la nostra fantasia addirittura dall'universo conosciuto. Quindi, tolgo il cappello anche a questo grandioso titolo.

Novag: il ritorno di Gir Draxon

Dynamix - Anno 1991 - Piattaforma PC

Sono passati due anni dalla famosa guerra di Stellar7, il nemico Gir Draxon non è stato sconfitto dato che non è stato trovato. Ecco a voi Novag, il famoso seguito che viene alla luce per via dell'improvvisa ricezione di una chiamata di emergenza dallo spazio.



Applaudiamo al ritorno del Capitano John Alex che dovrà pilotare il nuovo carro armato iper tecnologico chiamato Raven2 per indagare sulla chiamata di soccorso.



Il bellissimo Novag, come annunciato nel precedente capitolo, è stato curato dalla Dynamix e ha mantenuto le pregiatissime prospettive, nonché caratteristiche grafiche, estremamente dettagliate.

La visuale del gioco permane con la medesima prospettiva in prima persona, dentro la cabina di guida. L'obiettivo rimane identico, cioè eliminare i nemici dei nove pianeti che dovremo raggiungere.

L'aggiunta di nuove armi, moduli di potenziamento, ricariche e scanner a lungo raggio saranno essenziali per poter proseguire incolumi nei vari scenari. E' stato criticato, da alcuni giocatori, l'inserimento di elementi a "puzzle" nei vari scenari, per movimentare un poco la storia e differenziarla dal predecessore Stellar7. De gustibus...

In questo titolo siamo approdati a piattaforme quali Amiga e DOS. L'ambiente grafico ha perso la magica atmosfera vettoriale, pura, fredda, perfetta. Abbiamo acquisito tantissime tonalità grafiche, risulta particolarmente degna di nota l'illuminazione

nei vari ambienti: con lo scorrere del tempo, nei rispettivi pianeti, il turnover stellare modifica lo scenario facendogli assumere gamme cromatiche e dinamiche molto più ricche.

Quali elementi sono rimasti a sottolineare la purezza del gameplay, a prescindere dall'abbandono dell'ambiente vettoriale? Possiamo annoverare un collocamento "Sci-fi" futuristico, permane il genere d'azione, si fregia nuovamente di una prospettiva in prima persona con una grafica di plancia notevolmente raffinata, infine il gameplay appartiene ancora una volta agli "shooter", d'altronde siamo a bordo di un carro armato. La pubblicazione è gestita dalla Sierra On-Line inc. mentre lo sviluppo è stato curato dalla nostra carissima Dynamix.

I commenti del pubblico di quegli anni sono stati a dir poco esuberanti. Il suono è stato definito squisito, soprattutto per i possessori di schede Roland. I sette pianeti di Stellar7 sono stati definiti abbastanza scontati, mentre i nove pianeti di Novag sono stati commentati come "pieni di sorprese".

La fluidità era appena accettabile con processori 286 mentre era pressochè perfetta su 386. Meglio ancora nei 486. La grafica 3D, per quei tempi, è stata amata tanto da definire "s sofisticate e dettagliate" le immagini che viaggiavano in risoluzione a soli 256 colori.



Alcuni commenti del pubblico, purtroppo, sono dolorosi, soprattutto per me, poiché i giocatori dei primi anni 90 erano stanchi dei giochi di grafica vettoriale. Non vedevano l'ora di immergersi nell'oblio dei colori, abbandonando la fredda purezza vettoriale costretta a possedere un'anima (relativa al gameplay) ricca di fervida immaginazione al fine di conquistare il pubblico ed ottenere vendite proficue.

Durante lo scorrere degli anni 90, il giocatore scalpitava con energia, pur di gettarsi tra le



braccia dell'evoluzione grafica, spesso a scapito di trama e storia più ricche.

Siamo sicuri di ricordare con precisione la corsa sfrenata della tecnologia x86? Accidenti, all'improvviso era arrivato il Pentium, un potentissimo factotum tanto desiderato, esso poteva fare tutto, ovviamente dalla prospettiva vissuta negli anni 90. Nel bene e nel male questa corsa lanciata a perdifiato in una prateria di risoluzione grafica dettagliata, ci ha portati all'8K odierno! Durante questa corsa virtuale nella prateria della programmazione, ecco spuntare come funghi noi Retrogamer... Non siamo meramente nostalgici della ricca anima del passato, ma soprattutto sostenitori di videogiochi che mirano a possedere storia, trama e personalità variegata come le mille sfaccettature di un diamante, apparentemente freddo, puro, perfetto.

Vi sto quasi convincendo a giocare una partita? Vi allego il link e vi consiglio di dare una chance anche al bellissimo Stellar7:

<https://classicroload.com/nova-9-return-of-gir-draxon.html>

Novag è l'evoluzione di Stellar7? Oppure è una involuzione? La grafica, a dir di molti, è progredita tanto da definirlo un gioco "onirico"? Oppure ha mantenuto scelte costruttive (a partire dalla plancia della carro armato fino agli scenari dei vari pianeti) abbastanza scontate? Impossibile scindere il tutto, la magia grafica negli anni 90 ci aveva fatti innamorare della ricchezza del gioco e, chiaramente, ciascun amore non ci istiga alla

ricerca di difetti, poiché diventa parte indissolubile del nostro vissuto.

Personalmente parlando, reputo importante analizzare a mente fredda e cuore sereno, a distanza di anni, ciò che realmente è accaduto al tempo. Per questo ringrazio infinitamente RM: quale splendida opportunità si può avverare attraverso questa rivista! Ringraziamo a questo punto anche il designer del gioco, Paul Bowman, a lui vanno tutti i ringraziamenti per un successo sintonizzato all'evoluzione grafica dell'epoca.

L'art director è stato Robert Caracol. I principali programmatori: ovviamente Paul Bowman e Nels Bruckner. Infine la grafica 3D è stata curata da Damon Mitchell e Cyrus Kanga. Un ringraziamento va anche ai numerosi programmatori e artisti che hanno seguito gli sviluppi di VGA, EGA, World design, Background Painting. Una squadra veramente agguerrita.

Vi allego infine, in calce alla recensione, una bellissima locandina. Vorrei indurre in voi una riflessione: possiamo notare il dettaglio grafico del velivolo disegnato dall'artista. E' bellissimo. Come possiamo giudicare invece i velivoli a modica gestione poligonale presenti nel videogioco? Al tempo il giudizio è stato sicuramente "a pieni voti", purtroppo oggi, li consideriamo "scadenti". Cari amici, negli anni 90 tutti sognavamo una grafica presente solo in questi anni. Le locandine ci hanno indotto a fantasticare, correre, ricercare il dettaglio grafico odierno di una ipotetica Lara Croft (p.e. l'amato Shadow of the Tomb Raider) che prende vita in un ambiente quasi identico alla realtà obiettivamente conosciuta. Anzi, spesso la composizione fotografica virtuale delle locandine odierne supera di gran lunga la realtà ambientale. La grafica vettoriale è lontana dagli schermi del 2018 e il sacro Vectrex è una parola tabù, però possiamo stare tranquilli: nei cuori dei Retrogamer il passato è tutto scritto a fuoco.

Con queste ultime riflessioni cari lettori ci accomiatiamo. A Febbraio in RM13 potrete leggere la mia prossima recensione. Questa volta non anticipo volontariamente nulla. Auguro a tutti voi Buone Feste, un abbraccio!

Di Michele Ugolini



"Sierra arcade games...from electrifying interplanetary battles to madcapped maze games to a multi-player zany romp through life 'in the fast lane'."

ARCADE ACTION FOR THE WHOLE FAMILY



Nova 9™ Draxon's Revenge
Space warriors, get ready! Nova 9 is a supercharged adventure from the designers of Stellar 7.

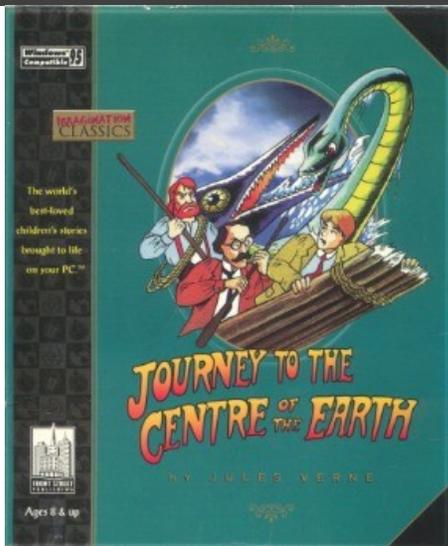
The dazzling hand-painted and digitized graphics with 3-D enhanced animation will explode on your computer screen, pitting you against evil aliens with an arsenal of futuristic high-tech weaponry.

You'll blast off to far-away galaxies and battle Draxon's forces on nine worlds of ever-increasing danger and difficulty. The music-card compatible stereo soundtrack will have you cruisin' into space combat in high style.

You'll need speed, cunning and lots of guts to defeat Draxon's alien forces.

NEW!

JOURNEY TO THE CENTRE OF THE EARTH



Copertina

Almeno a giudicare dalla copertina è difficile non trovare riferimenti al romanzo di Jules Verne "Viaggio al Centro della Terra" vero?



Un gioco "tosto"

Il sistema di caverne è esteso e basta il minimo errore per perdere una vita preziosa. Inoltre, è necessario tornare in superficie spesso per liberare spazio all'interno dell'inventario.

GIUDIZIO SUL GIOCO

GIOCABILITA'

85%

E' abbastanza semplice capire come procedere all'interno del gioco. Interfaccia utente intuitiva e personaggio sufficientemente reattivo. Per gli amanti del genere è un gran bel gioco da giocare!.

LONGEVITA'

65%

Data la difficoltà oggettiva di esplorare tutto il sistema di caverne per recuperare i vari tesori, riportandoli via via in superficie, è difficile che la voglia di rigiocare torni presto a partita finita.

Journey to the centre of the Earth

Ozisoft - Anno 1984 - Piattaforma Commodore 64

Si tratta di un videogioco di avventura, ambientato in un labirinto di caverne, ispirato al romanzo "Viaggio al centro della Terra" sebbene non in maniera troppo esplicita, che fu pubblicato nel 1984 per Commodore 64 dalla società australiana OziSoft. Fu ripubblicato l'anno dopo dalla società britannica CRL Group con il titolo "Journey".

Il giocatore controlla un esploratore (il Prof. Lidenbrock, presumibilmente, nome del protagonista del famoso romanzo di J.Verne) che, partendo dalla superficie, entra all'interno di un imponente sistema di caverne abbastanza intricato, alla ricerca di ben 11 tesori nascosti all'interno, l'ultimo dei quali si trova alla massima profondità, difeso da un troll il cui contatto risulta mortale.

L'ambiente è un labirinto a 2 dimensioni composto da cunicoli e passaggi, con scorrimento in tutte le direzioni. Molti di questi cunicoli sono piattaforme orizzontali, ma ci sono anche passaggi verticali e soffitti inclinati.

L'esploratore può camminare in orizzontale, arrampicarsi sui lati dei cunicoli verticali, e saltare in alto o in lungo a seconda della pressione breve o prolungata del pulsante fuoco.

Si può anche aggrappare al volo alla parete mentre cade, se non ha ancora preso troppa velocità.

Le cadute da una altezza eccessiva causano la perdita di una vita e si possono incontrare molti altri pericoli letali al contatto, come ad esempio pipistrelli, vampiri, pozze acide, fiumi sotterranei, massi cadenti, draghi, oltre al già citato Troll.

Sparsi all'interno del labirinto si possono trovare vari tipi di oggetti utili da raccogliere, tra cui le pistole, utili per abbattere i pipistrelli, oppure la dinamite per far saltare muri aperti parzialmente (ma servirà anche il detonatore per adoperarla), oppure ancora pillole di ossigeno per attraversare l'acqua, l'ombrello per proteggersi dalle rocce che cadono etc.

L'esploratore ha un inventario con spazio sufficiente per contenere fino ad 8 oggetti simultaneamente, rappresentati da icone nella parte inferiore dello schermo. L'inventario si può anche visualizzare in una schermata a parte con descrizioni in inglese delle icone. Con i tasti numerici ed un cursore si può selezionare l'oggetto da tenere attualmente in mano, per utilizzarlo al momento giusto con il pulsante del joystick.

Non è possibile né saltare né arrampicarsi quando si ha un oggetto selezionato. Ogni tesoro raccolto occupa temporaneamente un posto nell'inventario come gli altri oggetti e dev'essere depositato in superficie.

Il personaggio che pilotiamo non può saltare quando ne trasporta 6 o più, e non può salire quando ne trasporta 8. Ecco perché diventa importante tornare in superficie più volte durante il gioco per scaricare i tesori che via via raccoglierete.

Altro importante fattore da considerare è che, in caso di perdita di una vita, si riparte dalla superficie e si perdono anche gli oggetti attualmente nell'inventario, compresi eventuali tesori!

Purtroppo è presente qualche glitch, qualche errore nella grafica e tali errori si manifestano sempre di più proseguendo nel gioco.

E' un gioco "tosto" da terminare, in particolare se non si ha a disposizione una mappa dei labirinti, data la mole del sistema di grotte e cunicoli. Difficile per i ragazzini di allora fornirsi di tutti questi strumenti, ahinoi, ma tra i links utili troverete comunque la mappa :)

Curiosità:

La colonna sonora, che resta spesso impressa in maniera indelebile nelle menti di chi ha giocato a questo gioco (me compreso ovviamente), è basata su "House of fun" dei Madness magistralmente interpretata <https://www.youtube.com/watch?v=GJ2XgS ANsME>

Suggerimenti per aggirare il Troll? Occorre evitarne il contatto ed avvicinarsi velocemente al tesoro dietro di lui approfittando dei brevi istanti in cui il Troll non è visibile.

Riferimenti

https://it.wikipedia.org/wiki/A_Journey_to_the_Centre_of_the_Earth
http://ready64.org/giochi/scheda_gioco/id/1094/journey-to-the-centre-of-the-earth-a
https://www.c64-wiki.com/wiki/A_Journey_to_the_Centre_of_the_Earth

di Marco Pistorio

CIVILIZATION



La schermata dell'Attitude Survey, che ci dice come stanno i nostri cittadini.



Abbiamo scoperto la ruota, e ora le nostre bighe faranno strage di nemici

GIUDIZIO SUL GIOCO

GIOCABILITA'

90%

Il gioco miscela in maniera egregia elementi molto differenti tra loro, e non si ha mai la sensazione di non sapere cosa fare. Certamente richiede una dose notevole di dedizione, e risulterà quindi molto complesso per chi è abituato a platform e puzzle.

LONGEVITA'

99%

Un gioco che non "finisce" sostanzialmente mai, e che anzi regala soddisfazioni sempre maggiori con il passare del tempo. Il giocatore occasionale non troverà nulla di interessante in Civilization, mentre l'appassionato lo abbandonerà solo all'uscita di Civilization II.

Civilization

Microprose Software - Anno 1991 - Piattaforma PC

"Civilization è un gioco a cui la parola GRANDE potrebbe adattarsi solo se fosse scritta a lettere cubitali alte 100 metri su un neon lampeggiante posto sulla sommità di un palazzo di cinquanta piani." Con queste parole la redazione di K inizia a descrivere Sid Meier's Civilization sul numero 34 del dicembre 1991. E leggendo queste parole mi torna alla mente l'impressione che ebbi quando iniziai a giocare con questa pietra miliare dei videogiochi: immensità.

Il 1991 è stato un anno sicuramente non avaro di capolavori, anche se ad essere onesti in un ambito affollato come quello dei videogiochi probabilmente nessun anno dopo il 1978 lo è stato. Lemmings, Street Fighter II, Sonic, Monkey Island 2, Another World, tutti titoli che hanno segnato un cambio di passo nei rispettivi generi, sia sul fronte tecnico che su quello della giocabilità.

MicroProse, nel settembre 1991 se ne esce con l'ultima fatica di Sid Meier, che il programmatore canadese aveva sviluppato assieme a Bruce Shelley, poi iniziatore della saga di Age of Empires. Meier aveva già pubblicato 22 titoli al tempo, e si era fatto conoscere specialmente per F-19 Stealth Fighter e Railroad Tycoon.

È indubbio che Civilization sia un punto di svolta nel mondo dei videogiochi, su molti fronti, ed in effetti una delle sue caratteristiche principali è proprio questa natura sfaccettata. Civilization è un wargame, ma manca di un obiettivo specifico, se escludiamo la vittoria per "conquista universale". È una simulazione diplomatica, che però va ad intrecciarsi con l'aspetto militare in quanto i nemici reagiscono alla presenza delle nostre truppe sul loro suolo. È una corsa all'avanzamento tecnologico e al progresso scientifico, che però dipendono fortemente da come gestiamo le città, che forniscono "cervelli" alle nostre scuole e università. È un manageriale, visto che le risorse delle città vanno amministrate con ocularità per evitare problemi di carestia o di stasi produttiva.

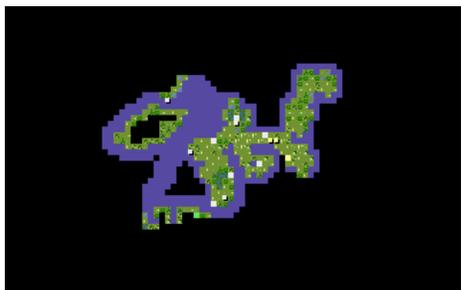
Va da sé che il dettaglio con cui i singoli aspetti sono affrontati è minore di quello che hanno (o avranno) simulazioni dedicate come ad esempio Sim City, ma la mole di fattori di cui tenere conto è veramente alta e nonostante questo il gioco rimane estremamente coinvolgente e divertente. Questo è sicuramente ciò che fa di Civilization non solo un gioco vasto, ma un grande gioco, dove l'apparente povertà di azioni dei coloni iniziali si tramuta in un universo di possibilità, restando però nei confini del gioco, senza diventare un puzzle cervellotico.

Ed è infatti con una carovana di coloni che ci troveremo ad iniziare il gioco, dopo aver assistito alla presentazione "cosmica" ed aver scelto la civiltà che vogliamo impersonare, il livello di difficoltà e la geografia del mondo. Il gioco non include ancora il concetto di fog of war, limitandosi ad annerire completamente le parti di mappa che non abbiamo visitato. All'inizio del gioco, pertanto, abbiamo visibilità solamente sulle 8 tile che circondano i coloni. Se siamo fortunati il gioco assegnerà alla nostra tribù un avanzamento tecnologico gratuito (ad esempio l'alfabeto o la ceramica) e/o un'unità aggiuntiva, che nelle prime fasi di esplorazione e difesa può decisamente fare la differenza.

Il gioco poi prosegue a turni in cui muoveremo tutte le unità libere, ma non c'è fretta in Civilization. A fine turno abbiamo tutto il tempo di visitare le città che fonderemo e di controllarne l'andamento, di pianificare la produzione, di rinforzarne le difese. E dovremo prendercelo, questo tempo, visto che di lì a poco cominceranno i contatti con le civiltà vicine, talvolta avidi di rapporti amichevoli, talvolta solo desiderose di vedere il nostro impero schiacciato sotto i loro piedi primitivi.

Le relazioni con gli altri imperi saranno un fattore costante del gioco e una notevole spina nel fianco in certi momenti cruciali. C'è sempre una città straniera nel luogo dove vorremmo costruire la nostra nuova capitale, c'è sempre un'unità nemica in una lingua di terreno che porta a lande inesplorate, impedendo alle nostre unità di passare a meno di dichiarare guerra, c'è sempre una nazione avversaria che scopre la bomba atomica prima di noi o che inizia a costruire navi spaziali quando noi abbiamo appena scoperto la polvere da sparo. Non si riposa sugli allori, in Civilization, e se lo si fa è meglio prepararsi a sentire la puzza di bruciato, perché qualcuno che vuole dare fuoco al nostro letto di gloria c'è sempre.

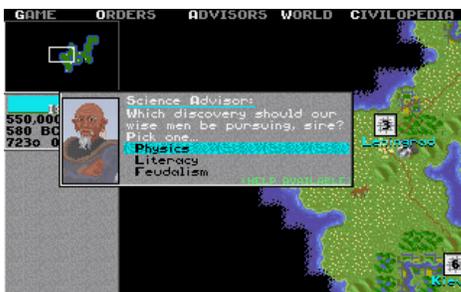
A meno che, gestendo con incredibile maestria i fattori militari, economici, scientifici, politici e sociali del gioco non arriviamo ad costruire un impero veramente potente, ma ci si renderà velocemente conto di quanto tirare tutti gli angoli della tovaglia risulti non banale e che anche l'impero più potente, se attaccato in modo oculato, può iniziare a zoppicare. La perdita di una città, ad esempio, significa la perdita della produzione militare, della ricchezza e del contributo scientifico, ma anche l'interruzione delle linee di comunicazione del nostro impero, o magari l'agognato accesso all'oceano.



Non si può governare un impero su scala planetaria senza una buona mappa



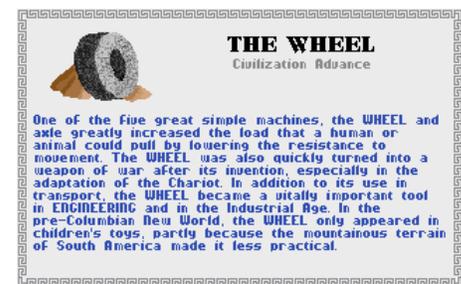
Fondiamo Leningrado, la prateria si riempirà presto di case, costruzioni e meraviglie del mondo



È ora di decidere in cosa investire il proprio tempo. Fisica o letteratura? Cannoni o diplomatici?



I nostri sforzi per tenere alto l'onore dell'impero sono ripagati da una modesta dimora nella campagna di Mosca



La Civopedia descrive in dettaglio le scoperte scientifiche a cui possiamo accedere.

L'esplorazione è un tema importante in Civilization, in quanto trovare il luogo giusto dove fondare una città può fare una grande differenza, in particolare se si riescono a sfruttare i terreni speciali, che quindi producono più del normale, soprattutto se adeguatamente trattati dai nostri fidi coloni. L'esplorazione è però connessa, come anche la potenza militare, al progresso scientifico. Non è sufficiente inventare il trirème per poter solcare l'oceano, tanto che queste unità devono terminare il turno a fianco alla costa o andranno perse. Sarà necessario scoprire la navigazione per potersi avventurare in mezzo alle distese di acqua che ci separano da terre in cui la nostra nazione potrà prosperare. O almeno questa è la nostra speranza quando carichiamo coloni e un paio di unità militari sul nostro guscio di noce, visto che all'inizio non sappiamo cosa ci sia al di là dell'acqua, e nemmeno DOVE sia questa fantomatica destinazione, ammesso che ci sia. Questo ovviamente a meno di giocare su una mappa conosciuta, ma anche lì, visto che la posizione iniziale nostra e dei nostri avversari è casuale, potremmo felicemente sbarcare in America per scoprire che il glorioso impero russo ha già costellato il continente di città dai nomi decisamente diversi da quelli che i Padri Pellegrini si sono portati dalla madrepatria.

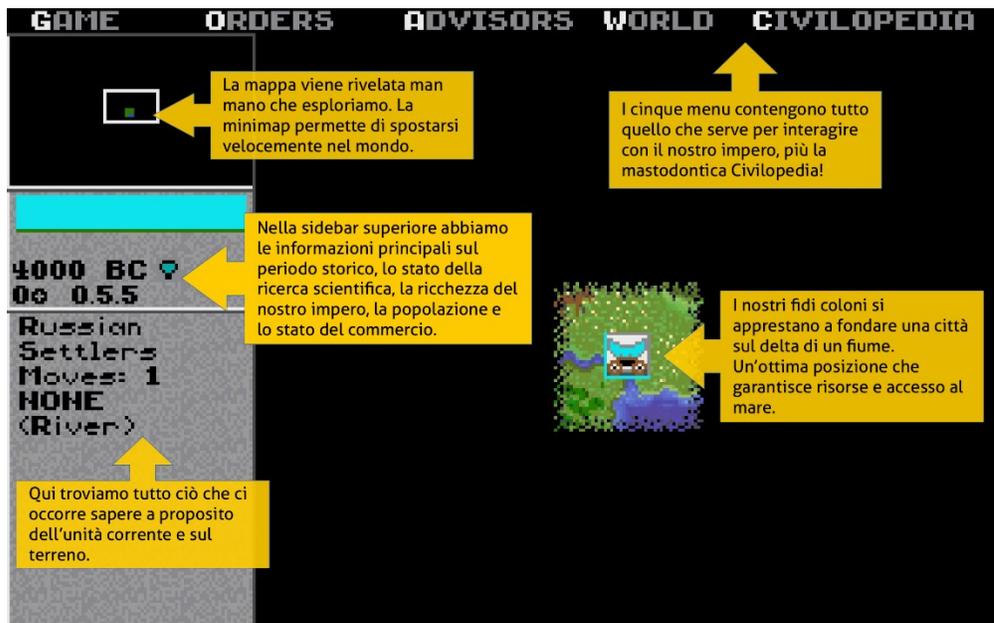
L'avanzamento tecnologico è quindi una trave portante dell'edificio che costruiremo accettando la sfida di Civilization. L'albero delle scoperte scientifiche è decisamente fitto e, se all'inizio del gioco le scelte possono sembrare in numero ridotto, ancora una volta basterà poco tempo per trovarsi di fronte a dilemmi tipici tra lo sviluppo di qualcosa che ci permette di migliorare le condizioni della popolazione, come una nuova forma di governo o un processo produttivo, e la nuova tecnologia militare che finalmente metterà tutti i nemici sotto i nostri piedi. D'altronde, avere città prosperose ma non protette è inutile quanto dominare il mondo ed avere una popolazione allo stremo.

Pianificare per tempo dove si vuole arrivare è quindi fondamentale, pertanto sarà necessario consultare spesso le descrizioni delle scoperte scientifiche e delle unità. E parlando di descrizioni non si può non citare il mastodontico manuale in linea. La documentazione a disposizione del giocatore non è solo ben fatta, ma copre ogni singolo aspetto del gioco, e documenta dettagliatamente ogni unità, infrastruttura cittadina o scoperta scientifica a cui è possibile accedere. Nel 2018, ormai viziati da Wikipedia e Google, siamo abituati a poter

accedere a basi di conoscenza che superano di molti ordini di grandezza quanto immaginato 30 anni fa, ma nel 1991 la Civopedia era un'opera monumentale, che contribuiva non poco a quell'impressione di immensità che si di fronte a Civilization.

Si scopriranno così i numerosi edifici che è possibile costruire nelle nostre città, ognuno con degli specifici vantaggi sul fronte produttivo o sociale. Tra questi bisogna menzionare le 21 meraviglie del mondo, che, una volta costruite in una città, portano un vantaggio globale a tutto il nostro impero. Queste vanno dai Giardini di Babilonia, che aggiunge un cittadino felice in ogni città, al Programma Apollo che permette i viaggi spaziali e rivela tutta la mappa. È decisamente deprimente quando, a due turni dalla fine della costruzione delle agognate Piramidi, uno dei leader avversari annuncia al mondo di averle appena terminate, costringendoci quindi a cambiare piani in corsa, solitamente buttando al vento turni spesi nella loro produzione. Anche le unità militari sono molto varie e beneficiano delle scoperte scientifiche. Scoprire la lavorazione del ferro permette di iniziare a costruire ponti ma anche di produrre le temute Legioni, e sfruttare adeguatamente l'ultima scoperta dei nostri scienziati può essere la chiave verso la vittoria.

Su questo fronte Civilization si permette di mettere in campo situazioni abbastanza paradossali, quando un impero si trova ad essere notevolmente avanzato rispetto ad un altro. Raramente questo raggiunge livelli estremi, in quanto imperi deboli e poco progrediti vengono velocemente mangiati dai più potenti vicini, ma in qualche occasione è possibile vedere catapulte in legno affrontare carri armati e in casi di questo tipo il sistema di combattimento scricchiola un poco. Le unità hanno infatti dei parametri di attacco, difesa e movimento che sono paragonabili ad altre unità dello stesso periodo ma non ad unità di periodi storici differenti. Una catapulta ha un valore di attacco di 6, mentre una artillery ha una difesa di 2. Questo significa fondamentalmente che è possibile disfarsi di un mezzo robotizzato armato di missili terra-terra tirando delle pietre con un braccio di legno, o che un gruppo di cavalieri medioevali potrebbe tenere sotto assedio una città moderna. Ad ogni modo, come già detto questi aspetti sono marginali e non inficiano minimamente l'esperienza di gioco. Sarà difficile che i vostri coloni si trovino ad affrontare un caccia bombardiere tedesco, in quanto Federico il Grande vi avrà già



Gli infidi Egiziani attaccano Odessa a sorpresa (beh, non tanto). La città ha appena perso la sua falange difesa e verrà conquistata. Preparati, Ramses, la vendetta arriverà!

From the past

Civilization uscì in America nel settembre 1991 per PC ed arrivò in Italia di lì a poco, non tradotto. **K Magazine** recensì il gioco nel dicembre 1991 (**K 34**), assegnando un voto di 930/1000 e conferendogli lo status di K-gioco nel settore strategia, ossia la pietra di paragone per tutti i giochi di quel tipo. **The Games Machine**, in quegli anni molto più polarizzata sull'Amiga che sul PC, recensirà il gioco nel febbraio 1992 (**TGM 39**) assegnandogli un 94% e la palma d'oro nell'innovazione. Il gioco fu poi convertito per Amiga, Macintosh e Atari ST, e approdò anche sulle console Sega Saturn, Super NES e Playstation. Nell'aprile 1992 il gioco era acquistabile per 79000 Lire (circa 70 Euro attuali), mentre nel dicembre 1994 il prezzo era di 99000 Lire (circa 80 Euro).

conquistati appena scoperta la polvere da sparo. Per 5 anni, fintanto che MicroProse non ha pubblicato il bellissimo seguito, Civilization ci ha messo in mano lo scettro del comando totale, ci ha fatto sognare di creare imperi "dove non tramonta mai il sole" e di raggiungere le stelle con le nostre astronavi spaziali, pronti a piantare la nostra bandiera su un nuovo pianeta. Grazie Sid e Bruce, ci avete regalato anni di divertimento assoluto, di delicate decisioni sociali e militari, di assoluta e percettibile "immensità"!

Requisiti tecnici

Il comparto grafico e sonoro non sono certamente il punto forte di Civilization, aspetto che i successivi capitoli della serie cureranno maggiormente, ma non se ne sente una grossa mancanza. L'interfaccia è pulita e ordinata e i numerosi shortcut che le unità mettono a disposizione permettono di effettuare la maggior parte del lavoro senza dover navigare tra menu o comunque usare il mouse. Questa periferica è comunque necessaria per apprezzare appieno il gioco, ma questa raccomandazione non è necessaria nel 2018.

Bugs

Come ogni software, specialmente di queste dimensioni, Civilization non è esente da bug, ma la cura generale del prodotto è ottima, tanto che gli errori si scovano solamente in particolari successioni di eventi del gioco. Un errore molto famoso in cui è facile imbattersi è quello che vede il pacifico leader indiano Gandhi mutare atteggiamento in maniera drastica da amichevole a guerrafondaio implacabile. L'origine di tale bug si trova nel (semplice) sistema di gestione

dell'aggressività di un leader, che è rappresentata da un numero intero tra 0 (pace assoluta) e 255 (guerra totale), evidentemente un intero a 8 bit. Specifici eventi del gioco riducono l'aggressività del governo di una nazione, e nello specifico il passaggio alla democrazia riduce l'aggressività di 2. I programmatori, però, si sono dimenticati di controllare la soglia inferiore di questo valore e Gandhi passa in quel frangente da 1 a -1, un numero negativo che si tramuta in 255 (dato che un integer ospita solo numeri positivi), trasformando la nazione indiana in un covo di assassini sanguinolenti che vogliono solamente vedere le nostre città rase al suolo e la nostra popolazione passata a fil di spada. Quando si dice che basta un nonnulla per scatenare una guerra...

Una famiglia numerosa

Quest'anno la Firaxis ha pubblicato Civilization VI, testimone che l'idea iniziale è stata brillante tanto da resistere per 30 anni nel ricco panorama videoludico. Ovviamente adesso siamo di fronte ad un prodotto del tutto diverso dal punto di vista grafico e sonoro, e anche l'algoritmica alla base del gioco è mutata, specialmente per quanto riguarda l'intelligenza degli avversari. Queste incarnazioni moderne, però, nonostante l'ovvio adeguamento visuale e la possibilità di sfruttare la potenza dei computer odierni, non si discostano in modo particolare dallo spirito del capostipite, che, con la povertà di mezzi del tempo, aveva già messo in mano ai giocatori tutti i mezzi per soddisfare i propri sogni di grandezza.

Gioco online

Oggigiorno è impossibile pensare ad un gioco come Civilization senza subito figurarsi frotte di giocatori che si affrontano online, ma nel 1991 questa possibilità non era contemplata, anche perché avrebbe oggettivamente riscontrato il favore di pochi giocatori fortunati possessori di un modem e del budget necessario ad accedere alle reti di telecomunicazioni che possiamo tranquillamente definire primitive a confronto di ciò che abbiamo ora. Quattro anni più tardi il panorama dell'accesso telematico era cambiato e MicroProse fece uscire CivNet, una versione di Civilization che permetteva a 8 giocatori di affrontarsi in rete o sullo stesso computer. Questa versione del gioco, però, uscì troppo a ridosso di Civilization II, pubblicato l'anno dopo, e si diffuse poco.

di Leonardo Giordani

Black Tiger

Capcom - Anno 1987 - Piattaforma Arcade

BLACK TIGER



Dungeons che passione!

L'ingresso di una delle missioni aggiuntive che potevamo svolgere per aumentare il nostro score



Shopping che passione

Una delle novità di Black Tiger fu l'introduzione dello shop per spendere gli zenny trovati per strada



Il riposo del guerriero

Il mondo è ormai salvo, possiamo finalmente riposarci e tenere tra le braccia la nostra amata

GIUDIZIO SUL GIOCO

GIOCABILITA'

90%

LONGEVITA'

90%

Sulla scia del grandissimo successo di Ghost 'n Goblins uscito nel 1985, la mitica casa giapponese Capcom lancia nel mercato arcade due anni dopo Black Tiger, platform in perfetto stile fantasy nel quale vestiamo i panni di un guerriero barbaro dotato di armatura e di una potente catena chiodata, che ricorda da vicino Rygar della Tecmo, il quale deve liberare il mondo dalle forze nemiche che lo tengono in ostaggio.



Se la storia non è certo un capolavoro di inventiva, in poco tempo Black Tiger è riuscito però a far breccia nel cuore dei video giocatori sia per il suo fantastico gameplay che per alcune innovazioni importanti che non erano presenti nei suoi illustri predecessori e che poi condizioneranno alcuni titoli successivi.

Innanzitutto i livelli di gioco sono stati pensati dagli ideatori della Capcom non più nella classica struttura lineare (da sinistra a destra per intenderci) ma con la possibilità anche di muoversi in verticale. Inoltre si possono esplorare aree nuove o decidere di tralasciare determinati percorsi per giungere subito alla conclusione dello stage di turno. Tutto questo contribuisce a creare nel giocatore un senso di libertà e di profondità prima sconosciuto.

Seconda importante novità è stata l'introduzione di un negozio all'interno del quale è possibile potenziare la propria armatura, la catena oppure acquistare pozioni curative o chiavi per aprire forzieri ricchi di tesori. Per poter acquistare questi bonus possiamo spendere una moneta, chiamata zenny che poi ritroveremo anche in altri titoli, che viene lasciata dai nemici che sconfiggiamo o da alcuni saggi anziani che possiamo liberare lungo il nostro percorso.

Terza novità è stata l'introduzione dei famosi dungeons, ovvero delle piccole missioni aggiuntive all'interno dello stage che possiamo scegliere di svolgere o meno, con le quali possiamo comunque raccogliere monete o altri bonus e aumentare il nostro score finale.

I nemici che cercano di sbarrarci la strada sono vari e numerosi, tutti appartenenti al più classico dei repertori fantasy: orchi armati fino ai denti, scheletri che spuntano dappertutto (proprio come gli zombi di Ghost 'n Goblins) demoni infuocati e serpenti pronti a morderci. Non sono da meno i potenti boss che ci aspettano alla fine di ogni stage: si parte dalle pietre animate che saltellano fino ai diversi dragoni via via più potenti e pericolosi. Inoltre il nostro cammino è costernato di trabocchetti e trappole come massi cadenti, gocce velenose o spuntoni nascosti che dobbiamo evitare a tutti i costi per non perdere punti vita.

Ad arricchire un repertorio già molto vasto, gli sviluppatori di Black Tiger hanno pensato bene di seminare in vari punti segreti delle mappe alcuni bonus o addirittura vite aggiuntive che è divertente cercare di trovare. Dal punto di vista grafico non c'è assolutamente nulla da dire perché ogni sprite è ben curato nel dettaglio e ricco di colori e il sonoro accompagna in modo egregio le varie fasi del gioco. Ma il punto di forza di Black Tiger, a mio avviso, è senza dubbio la sua giocabilità infatti è davvero piacevole muoversi e saltare con il nostro eroe tra i vari livelli e dungeons di questo magnifico platform.



Come tutti i successi arcade, anche questo titolo Capcom ha avuto diverse conversioni per i computer e console casalinghe ma purtroppo il lavoro fu affidato ai Tiertex della U.S. Gold che all'epoca erano famosi per conversioni approssimative o difettose e Black Tiger non sfuggì a questo destino. L'unico adattamento degno di nota è stato quello per Amiga, quasi identico nella grafica, ma ostico nella sua giocabilità.

Voglio salutarvi con un riferimento trovato in rete. Sembra infatti che Black Tiger sia il primo dei tanti easter egg nascosti da Cline nel suo libro *Ready Player One* (diventato film con Spielberg) poiché il cabinato invitava i giocatori ad inserire le monete ed entrare nel suo fantastico mondo.

di Querino Ialongo

SPACE MANBOW



GIUDIZIO SUL GIOCO

GIOCABILITA'

98%

Sarebbe perfetta se non fosse per la difficoltà estrema.

LONGEVITA'

99%

Quasi infinito, richiede mesi di pratica per finirlo.

Per giocare a Space Manbow occorre un computer MSX2/MSX2+ (o un emulatore come BlueMSX e OpenMSX per Windows o CocoaMSX per Mac), la ROM del gioco (scaricabile in rete) o la cartuccia originale, purtroppo disponibile a costi inviccinabili sul mercato dell'usato. Qui troverete il mio longplay di Space Manbow, soluzione del primo loop del gioco senza perdere vite: <https://www.youtube.com/watch?v=LeBRLHbtOH8&t=8255>

Space Manbow

Konami - Anno 1989 - Piattaforma MSX2/MSX2+

Space Manbow è uno sparattutto per MSX2/MSX2+ realizzato dalla Konami nel 1989 e commercializzato solo in Giappone. Nato come porting casalingo del bellissimo arcade Thunder Cross, il titolo si è evoluto in un capitolo autonomo, diventando uno dei giochi più spettacolari dell'era a 8 bit (il titolo originale del progetto era "Egzart"). Space Manbow si svolge nella regione spaziale Alfred-4 (!), i cui abitanti scoprono i resti di un'antica civiltà. I malcapitati, forse spinti dalla curiosità, attivano inavvertitamente il sistema di difesa messo in atto dagli antichi, vale a dire l'enorme astronave Sun-Fish, che il giovane pilota Clever Mew dovrà sconfiggere a bordo della navetta Mambo-J, un temibile vascello spaziale a forma di pesce mambo (!). Space Manbow sviluppa il paradigma ludico di Gadius aggiungendo una struttura complessa, grande varietà e un singolare comun denominatore, vale a dire la natura "acquatica" di gran parte dei boss finali. Infatti, durante l'avventura dovremo affrontare boss che ricordano aragoste giganti, strane conchiglie, pesci minacciosi e via dicendo, un elemento che aggiunge un pizzico di stranezza tutta nipponica.

L'astronave dispone di un armamento molto vario. Il colpo di base è un laser di cui è possibile incrementare la potenza raccogliendo i bonus rossi, con l'aggiunta di due "pod" indistruttibili che consentono di sparare in diverse direzioni, selezionabili con il secondo pulsante del joystick. Non provate a giocare a Space Manbow con un joystick a un solo tasto, amici. Il sadismo degli sviluppatori Konami ha prodotto un sistema diabolico: la potenza del colpo, che è possibile incrementare faticosamente a suon di bonus, tenderà a ridursi con il passare del tempo, aspetto che costringe a raccogliere continui bonus e spinge il giocatore a compiere spericolate acrobazie. Si tratta di uno dei principali elementi strategici del gameplay, a cui si aggiungono i due tipi di colpi disponibili (diretto e a tripla gittata) e la smart bomb (bonus blu), che consente di eliminare ogni elemento nemico su schermo.

Space Manbow è un gioco dal concept impressionante e in grado di rivaleggiare con i tipici giochi arcade del periodo. I livelli del titolo Konami sono molto vari: si parte dall'assalto a una fortezza cingolata, protetta da un boss dotato di braccio meccanico, per poi passare a un quadro indoor in cui occorre attraversare un budello claustrofobico contenente ogni sorta di nemico. Da qui passeremo allo spazio: un lungo canale di roccia sulla superficie di un pianeta misterioso in cui blastare ogni elemento visibile. In

questo livello dovremo farci strada fra terribili nemici, fra cui i malefici cursori che disegneranno, in tempo reale, un labirinto intorno alla nostra nave, per poi affrontare un mega paguro. Quindi, torneremo in un quadro irto di girandole e nemici e custodito da un'enorme aragosta meccanica. In seguito dovremo attraversare a velocità supersonica un cunicolo rosso fuoco ispirato all'attacco alla morte nera di Guerre Stellari, lanciarci in un lungo livello pullulante di nemici biomeccanici, affrontare un esercito di coriacee formiche robot e sconfiggere il boss finale del gioco dopo un temibile campo di asteroidi. La difficoltà del gioco è atroce: come se ciò non bastasse, gli sviluppatori hanno inserito pochissimi checkpoint, aspetto che complica ulteriormente la missione. Alcuni livelli, come ad esempio quello delle formiche robot, sono di una difficoltà quasi impossibile e richiedono un allenamento indefesso. I boss sono di difficoltà altalenante, passando dal ridicolo guardiano del penultimo livello alla sfera meccanica posta alla fine del tunnel laser, quasi impossibile da giustificare senza una smart bomb. Anche la lunghezza dei quadri è incostante: si passa dal brevissimo ma intenso livello delle formiche robot fino all'interminabile terzultimo quadro.

In generale, Space Manbow è uno degli sparattutti più spettacolari dal punto di vista tecnico e artistico dell'intera epoca a 8 bit. Si tratta di un prodotto che, escludendo qualche sfarfallio e un frame rate non cristallino, non avrebbe nulla da invidiare anche agli arcade più blasonati. Il gioco Konami è dotato di un livello impeccabile di cura, varietà e attenzione ai particolari: ogni elemento, pixel, nemico e pattern è accuratamente posizionato, animato e giustificato, rendendo Space Manbow un titolo in grado di umiliare qualsiasi sparattutto a 8 bit (e anche diversi prodotti a 16 bit) per il mercato home. Lo scrolling del gioco è abbastanza fluido, anche se non a 50/60 Hz (MSX2 non dispone di scrolling orizzontale con assistenza hardware), e l'azione di gioco scorre senza intoppi nonostante qualche sporadico rallentamento. La versione SCC del gioco vanta una delle colonne sonore più evocative di sempre, che offre melodie accattivanti, ritmi serrati ed effetti sonori di qualità immensa. Il sonoro del gioco è opera di Tsuyoshi Sekito, musicista che ha contribuito anche a Brave Fencer Musashi della Square.

Adriano Avecone

RetroGiochiAmo: The Official Father Christmas

di Daniele Brahimi

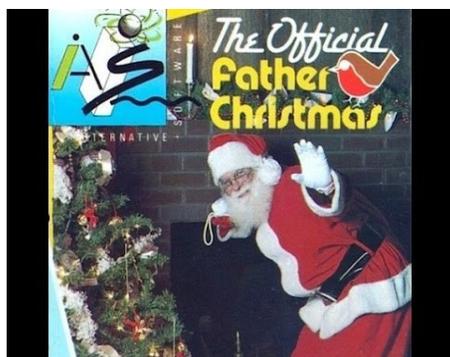
La nostra rubrica **RetroGiochiAmo** continua imperterrita anche grazie al nostro affezionato lettore **Daniele Brahimi**, che ha ormai fatto dello scrivere articoli per la nostra rivista un appuntamento pressoché fisso.

La cosa non può che farci piacere e vogliamo ribadire l'invito a tutti voi a seguire le orme di Daniele per invitarci a scoprire i giochi a cui siete particolarmente legati.

Dopo lo stupendo numero del mese scorso con lo speciale Halloween, rieccoci di nuovo qui con questo nuovo numero anch'esso speciale. Dico anch'esso speciale per dei semplici motivi.

Primo perché siamo sotto le feste natalizie e sappiamo tutti che momento magico è ed è stato per noi retrogamer, grazie alle uscite dei capolavori videoludici che trovavamo sotto l'albero ed anche computer: il mio Commodore 64 lo trovai proprio sotto l'albero nel 1987.

Secondo perché anche in questo numero ho voluto rimanere in tema tirando fuori dal cappello un bel videogioco che in pochi conoscono ma che comunque valeva la pena di essere giocato soprattutto in questo periodo. Sto parlando di **The Official Father Christmas!**



Quando il tutto iniziava si veniva subito accolti dalla classica canzoncina di Natale che si sentiva ovunque e già da lì si sentiva la magia del gioco in cui si poteva venire immersi premendo subito fire e aiutare così il nostro amato Babbo Natale a consegnare i regali in

tempo a tutti i bambini del mondo noi compresi.

Il gioco inizia con il nostro Santa Klaus in una casa infestata da folletti dispettosi e il nostro scopo è quello di recuperare tutti i pezzi della slitta necessari per poter farla partire; apparentemente non ci dovrebbero essere grosse difficoltà a parte gli odiosi folletti che ci rubano il pezzo facendolo ritornare al punto in cui è stato preso e si spera sempre che i pezzi siano recuperabili comodamente in un punto vicino anche perché il tempo è limitato e lo avremo fino al sorgere del sole, raffigurato in basso allo schermo. Una volta raggruppati tutti i pezzi della slitta andremo così a consegnare i regali nelle località di tutto il mondo, dall'Europa all'Antartico!



Uno dei particolari che mi ha incuriosito del gioco è stata la scelta dei regali da consegnare prima di partire e ce ne sono per tutti i gusti: un televisore, un orsacchiotto di peluche, un libro ecc.. Ah ed anche un computer game guarda caso somigliante al Commodore 64!



Forse questo gioco non ha avuto il successo che molti meritavano e non era nemmeno facile trovarlo se non in qualche cassetta da

edicola, ma come ogni altro gioco superiore o inferiore che sia, vale la pena di essere vissuto e massimo rispetto come sempre per gli ideatori, sviluppatori e via discorrendo.



Se alcuni di voi sono diventati papà ed hanno figli piccoli, consiglio di far provare loro questo simpaticissimo gioco. E provatelo anche voi nostalgici delle feste natalizie di allora, in cui si stava tutti in famiglia e non si aspettava altro che correre sotto l'albero la mattina di Natale a scartare i regali.

Detto questo ragazzi e ragazze (ebbene sì, sono sicuro che ci siano anche lettrici donne di questa rivista e che la parità dei sessi sia arrivata anche qui) come anticipato sopra, visto lo scorso numero di Halloween che è stato un capolavoro assoluto, senza nulla togliere ai precedenti ovviamente, ho voluto che anche questo numero sia speciale grazie ancora una volta a questo mio contributo ripescando questo gioco, anche perché le feste si avvicinano e spero che tutti possiate godervelo tra la neve, le illuminazioni e gli addobbi, possibilmente al caldo, davanti al camino.

Vi auguro buone feste a tutti voi e a tutta la redazione.

Grazie Daniele, da parte di tutta la Redazione di RetroMagazine Buone Feste a te ed a tutta la tua famiglia!

RetroGiochiAmo: Gabriel Knight - Sins of the Fathers

di Dante Profeta



Sin da ragazzino sono rimasto sempre affascinato dai giochi di avventura, inizialmente testuali, poi da quelli del genere "punta e clicca". Avevano il fascino dell'esplorazione, dell'apparente mancanza di confini per cui tutto può accadere e ogni luogo è da visitare ed analizzare con l'attenzione meticolosa dell'investigatore.

Gli Adventure mi hanno profondamente segnato, soprattutto quelli testuali: eccomi lì, dodicenne, davanti al C64 a giocare, o meglio a studiare The Hobbit della Melbourne House, Spider-Man della Questprobe..., col pesantissimo dizionario Ragazzini e un quaderno a quadretti accanto, intento a tradurre e decifrare l'inglese ancora prima che gli enigmi del gioco. Se ho imparato la lingua inglese ai livelli professionali a cui sono giunto negli anni lo devo proprio all'input avuto dalle avventure testuali di quando ero teen-ager.

Poi per me venne Labyrinth sempre sul C64, forse la prima avventura grafica del genere "punta e clicca", e tutto, o quasi, cambiò: le avventure testuali persero a poco a poco terreno, ma prima raggiunsero l'apice col raffinatissimo parser della Magnetic Scroll che nel 1987 sembrava aver dotato Amiga della capacità di superare il Test di Turing. Surclassate da SCUMM, il motore grafico a scripting della LucasArts, celeberrimo per via

di masterpieces come Manic Mansion, Monkey Island, e altri.

Infocom, Questprobe, Melbourne House, Sierra On-Line, Magnetic Scroll..., ormai nomi dimenticati, sono stati i pilastri dell'industria dei giochi di avventura sugli home computer e sui pc degli anni '80 e '90. Tra questi, proprio agli inizi degli anni '90, la Sierra On-Line propone uno di quei giochi che sarà destinato ad entrare a pieno titolo negli annali, per il coinvolgimento della storia, per l'ambientazione dark noir, per il cast di doppiatori degno di un film Hollywoodiano, rivolta ad un pubblico di giocatori ormai non più adolescenti, emotivamente coinvolgente al punto da diventare letteratura: **Gabriel Knight: Sins of the Fathers**.

Il gioco si apre con una sequenza di fotogrammi statici animati in zoom che descrive un orrendo delitto a danno di una presunta strega data al rogo in un tempo remoto, di un medaglione, e si conclude con le immagini dell'impiccagione di Gabriel, apparentemente un suicidio, avvenuto dopo qualche tempo dal rogo. Incubo ricorrente del protagonista.

I particolari sono il sale del coinvolgimento emotivo di Sins of Fathers, e l'azione comincia in una strada che prende il nome di Bourbon Street, nel cuore del quartiere francese di

New Orleans, proprio con un paperboy che lancia il giornale del giorno sulla soglia del negozio di libri rari di Gabriel, dall'evocativo nome St. George's Books. La routine del paperboy che lancia il giornale si ripete instancabile giorno dopo giorno. Il giornale, importante rivelatore di indizi, è parte integrante dell'atmosfera di gioco, e attraverso l'oroscopo ci farà capire quanto è profonda la tana del bianconiglio in cui assieme a Gabriel ci stiamo per inabissare.

Di lì a poco la sua brillante assistente e segretaria Grace, nonché malpagata impiegata, arriverà nel suo sventurato posto di lavoro, raccoglierà il giornale e, come da consuetudine, darà inizio alla routine giornaliera di apertura e gestione, senza nessun altro aiuto necessario, della sempre deserta bottega di libri rari di Gabriel.



Nel frattempo, al piano di sopra, Gabriel si sveglia trafelato in un bagno di sudore per via dell'incubo avuto durante la notte.

Giù, l'integerrima e incorruttibile Grace, preso posto dietro al banco, con poche battute al telefono, recitate evidentemente con una irata fiamma della notte prima del protagonista, lascia intuire che razza di sciupafemmine sia il biondo e aitante eroe, dotato della profonda e cavernosa voce del grandissimo Tim Curry.

Ogni nostra azione viene scandita dalla suggestiva voce narrante rauca della indimenticata attrice Virginia Capers, che con una buona dose di humor dark, sottolinea le azioni di Gabriel e scandisce il passare del



tempo aggiungendo un contributo denso di emotività e chiaroscuri sul nostro concedere.

Gabriel, ormai sveglio, ma tutt'altro che riposato, sceso in negozio si dedica alla sua consueta tazza di infuso di caffè, come piace agli americani, e alla lettura dell'unica pagina simulata, la prima, del New Orleans Times. Capiamo a questo punto che la vicenda si svolge a New Orleans, città famosa in tutto il mondo per il Voodoo e la magia nera. Il titolo dell'articolo in prima pagina infatti non lascia dubbi sull'ambientazione noir dell'avventura, densa di esoterismo, "Voodoo Murders Terrify Residents", e procede descrivendo i ben sei omicidi in due mesi, tutti irrisolti e apparentemente legati a rituali Voodoo che atterriscono l'intera cittadinanza.

Gabriel è un incipiente scrittore senza fortuna, intento al momento della narrazione nel raccogliere materiale per scrivere un libro che non sembra riuscirci troppo bene a giudicare dai fogli accartocciati e buttati nel cestino dell'immondizia sotto la scrivania della sua stanza da letto/studio.

L'avventura è ritmata da una musicchetta ossessiva, ma perfetta per l'atmosfera del gioco, che non ci abbandonerà mai neanche dopo venticinque anni dalla prima immersione nell'immaginifico mondo di Gabriel.

È venerdì, 18 Giugno del 1993. Durante il primo giorno faremo la conoscenza del caro amico Mosley, ispettore di polizia, con a sorpresa la voce Starwarsiana di Mark Hamill, che lo metterà sulle giuste tracce per la discesa verso gli inferi a cui sembra essere inevitabilmente destinato, come in ogni trama noir che si rispetti. E in effetti scopriremo assieme a Gabriel che dietro la storia della sua famiglia si celano indicibili segreti, the Sins of the Fathers appunto, che verranno a poco a poco messi sempre più a fuoco man mano che ci addentriamo nell'avventura, per diventare nitidi nei due

capitoli successivi, fino ad arrivare ai tempi di Cristo, alla crocefissione narrata attraverso la rilettura dei vangeli apocrifi... ma questa, come si dice, è un'altra storia.

La ricerca storica dell'autrice Jane Jansen è estremamente accurata e conferisce a Sins of the Fathers, come ai successivi due capitoli, una profondità e un'immersività ricca di particolari portandoci nel cuore del Voodoo, il Voodoo di New Orleans, della sua storia e delle sue tradizioni, alla narrazione di Marie Laveau, anche nota come la vedova francese, considerata l'ideatrice del Voodoo di New Orleans, ai rituali carnefici che permeano l'intera trama dell'avventura.

I giorni si susseguono cupi e intrisi di mistero. Ovunque Gabriel vada lascia una scia di sangue e ad ogni indizio se ne aggiungono altri provenienti dalla vecchia Europa.

In un paese della Germania rurale, dove il protagonista rintraccerà le sue radici, capirà chi egli è in realtà e qual è il destino che ha ereditato dai suoi avi. Scoprirà presto di essere uno Schattenjäger e di avere un rito iniziatore da compiere su se stesso per ricoprire il ruolo che lo attende, e prendere su di sé il pesante fardello che suo zio Wolfgang Ritter, il fratello di suo padre, ha portato sulle spalle prima di lui. Il nonno di Gabriel infatti, sperando di spezzare la maledizione della sua famiglia, emigrò negli Stati Uniti d'America, a New Orleans appunto, e cambiò nome da Ritter a Knight, ma le colpe dei genitori, come nella migliore tradizione letteraria sin dai tempi classici, ricadono sempre sui figli, e gli dei non dimenticano né perdonano. Sins of the Fathers.



Il puzzle dell'orologio – Alert: spoiler ahead

Uno dei puzzle che mi ha dato più filo da torcere è stato senza alcun dubbio quello dell'orologio degli antenati nella soffitta della nonna di Gabriel, perché è fondato su indizi indiretti, basati su immagini, oggetti e parti di dialogo che non verranno ripetute nel gioco.

In primis la statua del drago di fianco al bancone della libreria. Il simbolo del drago appare anche nel libro di poesie "Drei Drakhen" (I Tre Draghi) di Heinz Ritter. Il libro dei serpenti poi apre un dialogo con Grace che sottolinea come le storie sui draghi siano sin dall'antichità in vero basate sui serpenti, e quindi il serpente indica il drago. E ancora, il dipinto del padre di Gabriel "Three Snakes in a Skull" (Tre Serpenti nel Teschio). E poi nel sogno ricorrente, il coltello che genera per metamorfosi tre serpenti, tre draghi appunto.

Infine, il numero tre. I serpenti e i draghi sono sempre tre. Il numero tre è il numero ricorrente in Sins of Father. Nel capitolo successivo diventa invece il numero dodici... ma questa è un'altra storia... o meglio, la stessa storia. La storia di Gabriel che si espande con un altro tassello di consapevolezza dell'atroce passato del suo maledetto casato.



Il puzzle del dispositivo di tracciamento (GPS) - Alert: spoiler ahead

Un altro puzzle complicato da risolvere, soprattutto per chi non ha un'addeittura ottima dimestichezza con la lingua inglese, è il puzzle del device tracker.

Una volta recuperato il dispositivo di tracciamento dal cassetto della scrivania di Mosley, e dopo averlo depositato nella bara da rito Voodoo del museo, occorre convincere il Dottor John, dotato della possente voce Klingoniana di Mike Dorn, a portare quest'ultima con sé al rito che si terrà nella notte di St. Joseph.

Per far questo sarà necessario scrivere sulla parete della tomba di Marie Laveau i simboli che compongono la frase DJ BRING SEKAY MADOULE

La difficoltà nasce dal fatto che bisogna comporre la frase a "mano libera" senza poterla ricopiare meccanicamente da alcun indizio specifico.

SPRINT 2076

Intervista a Fabrizio Radica: Cinema, VR, Amstrad, Sprint2076 - terza puntata

di Michele Ugolini



La nostra città, realtà o sogno?

Difficile rimanere indifferenti all'arte: reale o astratta che sia, veste sempre indumenti riconoscibili a prima vista.



Evoluzione è una parola riduttiva!

Un tramonto in lontananza e una pavimentazione intrigante, manca solo la vettura ...



... per smarrirsi dentro un prodigioso coinvolgimento reale, legato indissolubilmente al virtuale!

Bentornati cari amici, in questa terza intervista a Fabrizio Radica potremo addentrarci nella sua bellissima opera Sprint2076 e scopriremo dettagli particolareggiati riguardo tutta la sua sfera multimediale. La passione Amstrad sarà un ingrediente piccante. Gli aspetti cinematografici legati al VR saranno un ottimo primo piatto. Infine la dedizione per il lavoro che svolge, sarà il vino che ci accompagnerà lungo tutto il pasto. Questo numero RM11 è in gran forma, dopo le allucinazioni di RM10 uscito ad Halloween. Come promesso, il lavoro sarà serio, pasteggeremo lentamente e scandiremo doverosamente il servizio a tavola. Tutti pronti? Via!

RM: Bentornato Fabrizio, la rubrica è alla terza puntata e ci sono grandi novità in arrivo. Recentemente sei stato relatore al simposio sulla fantascienza a Roma, hai discusso temi attualissimi e tremendamente interessanti come cinema e VR, ti andrebbe di parlarne?

FR: Sono stato invitato dall'Accademia Achille Togliani nella persona di Adelmo Togliani che saluto con affetto. Abbiamo parlato della possibile evoluzione del cinema (in questo caso italiano) grazie alle nuove tecnologie.

La Realtà Virtuale è un'ipotesi molto interessante, che va studiata ed elaborata. Come capirai, va cambiato un poco tutto, dalla sceneggiatura alle riprese, ma il risultato potrà essere di sicuro impatto.

Lo spettatore passa dalla situazione di "Passività" a quella di interazione con la scena. Ci sono esperimenti ed attività già in essere, pertanto se avrete voglia, ne parleremo in seguito.

RM: Passione Amstrad, saremmo veramente onorati di saperne di più e creare una sorta di sotto-rubrica dedicata a questo meraviglioso home computer, sia in questa, che nelle prossime interviste. Ti andrebbe di parlarci riguardo le tue esperienze Amstrad?

FR: Certo! Cercherò di essere sintetico, ma ci sarebbe tanto da parlare, tra aneddoti e curiosità! Amstrad è una macchina meravigliosa. Purtroppo non ha preso piede qui in Italia per via della tardiva distribuzione.

Parliamo degli ultimi anni 80. In Inghilterra, Francia e Spagna ha avuto molto più successo essendo stata una macchina completa. In alcune versioni compatibili al CP/M (sistema simile e precedente al DOS).

Ho avuto la fortuna di conoscere l'Amstrad CPC6128 nel 1989 quando tutti possedevano il Commodore64, io piangevo in un angolo e mi programavo i giochi (scherzo ovviamente).

Negli ultimi anni ho realizzato dei mini corsi di programmazione in Locomotive Basic e sviluppato anche giochi completi. Credetemi, il potente Basic permette di realizzare cose incredibili.

La famiglia Amstrad CPC si basa sul processore Z80, molto usato anche su altri computer del periodo, come MSX o Spectrum. Amstrad e Spectrum condividevano molto, basti pensare alla quantità di conversioni dirette tra il computer Sinclair e il CPC. Per certi versi giovarono al parco ludico, ma per altri, non sfruttarono appieno le potenzialità grafiche e cromatiche della macchina.

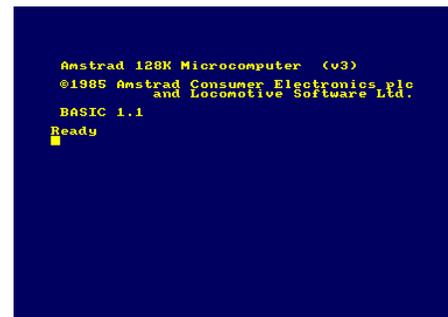
Cosa analogo successa con l'avvento dei 16bit, tra Amiga ed Atari ST, dove la prima per un breve periodo riceveva conversioni dalla seconda. La Serie CPC aveva dalla sua parte (grazie alla versione da 128k), la possibilità di accedere anche al CP/M e poter essere così utilizzata in ambiti professionali.

Ciò che più amo degli Amstrad è il design, soprattutto quello del CPC6128: è molto professionale. Non dobbiamo sottovalutare neppure potenza e velocità del Basic, che permettevano di accedere completamente all'hardware, a differenza del Commodore64, o delle altre piattaforme ad 8bit.

Il Locomotive basic permetteva di creare un gioco complesso, gestire, elaborare e definire sprite. Suoni complessi grazie al processore Yamaha (uguale a quello dell'MSX, Spectrum 128k ed Atari ST).

Accesso al disco e gestione degli interrupt. Potete immaginare voi stessi la qualità del software che era possibile sviluppare comodamente a casa.

Successivamente Amstrad acquisì Sinclair, difatti il +2 e +3 assomigliano moltissimo al CPC 464 e CPC 6128.



Nei primi anni 90 la serie CPC si era evoluta in CPC+ e "donava" a tutta la serie: sprite, scrolling hardware (blitter) e una palette di 4096 colori con risoluzioni simili a quelle dell'Amiga.

Purtroppo però, era troppo tardi.

Se avete passione e voglia di sperimentare, consiglio di provare a programmare il Locomotive Basic, non ve ne pentirete.

Chiaramente potete postare informazioni e/o richieste sul gruppo **RetroAcademy**. Sarò ben lieto di aiutarvi.

RM: Per la gioia del mondo "Amstradiano" potresti spronare i numerosi fans, iniziando con un bellissimo regalo natalizio? Un listato a tua scelta, creato proprio da te, il numero 03, libero per la rete ma imbrigliato nero su bianco in questo RM11!

FR: Certamente! (NdR. Si veda sorgente a pie' pagina). In questo semplice sorgente è possibile disegnare un livello di un ipotetico gioco.

RM: La rete, incluso io, siamo tutti impazienti di conoscere le novità di Sprint2076, il taglio grafico è nettamente maturato ed evoluto, lo stile Neon-city si è coricato? E' in fase di maturazione? E' stato abbandonato? Oppure la modalità grafica di Sprint sarà selezionabile in base ai propri canoni di preferenza?

FR: Assolutamente NON è stato abbandonato, anzi, a Dicembre io ed il mio collega **Igor Imhoff**, affronteremo più concretamente il game design e la conclusione della prima, vera, pista di gioco. Ho personalmente provato stili diversi e "pompato" di più l'engine, crediamo si possa veramente raggiungere una qualità molto elevata, sia visiva, che di contenuti.

RM: Previsioni di uscita di Sprint?

FR: When is done!

RM: Hai altri progetti dedicati al mondo dell'informatica?

FR: Sì, ho progettato una "puntata zero" di una mostra dedicata alla PixelArt all'Arte digitale in generale in quel di Albignasego (PD) a metà Dicembre. Questo progetto (che ancora non ha un nome), non è necessariamente legato alla nostra passione, ma a tutto ciò che la nostra mente è in grado di creare con un computer: Quadri, Video, Giochi, Demo etc. In ogni caso trova ispirazione dai vecchi elaboratori, infatti verranno esposti ed utilizzati alcuni retro computer. Per quel che mi riguarda, credo che il 2019 sarà un anno di cambiamento specialmente per la creazione di eventi dedicati all'arte digitale, in tutte le sue forme. Adesso basta solo trovare un nome idoneo.

Eccoci qua ragazzi, viste le grandi novità trattate, conviene rimanere sintonizzati con questa rubrica. Rinnovo i ringraziamenti alla Redazione di RM che permette questa golosa



opportunità di recensire un lavoro tutto italiano. Ringrazio di cuore Fabrizio per assecondare le numerose domande che gli pongo ad ogni intervista.

Come una lastra di Raggi X che porta alla luce tantissime informazioni... Chissà! Forse arriveremo ad un dosaggio di radiazioni tale da assumere qualche potere? Il rischio per me

potrebbe essere concreto: vi anticipo che a Febbraio, nella quarta puntata, porterò novità anche dal Sol Levante!

Bene, da parte mia e di Fabrizio, tanti Auguri di Buone Feste a tutti voi: Lettori, Redazione, Staff, un abbraccio!

```

5 'TileMap Example By Fabrizio Radica
10 cls: maxx%=20:maxy%=11:DIM map%(maxx%,maxy%)
20 'Read Map
30 FOR y%=1 TO maxy%
40 FOR x%=1 TO maxx%
50 READ a%:map%(x%,y%)=a%
60 LOCATE x%,y%:if map%(x%,y%)=1 then print CHR$(127);:
70 NEXT x%
80 NEXT y%
90 'Map Data
100 DATA 0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0
110 DATA 1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1
120 DATA 1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1
130 DATA 1,0,0,0,0,0,0,1,0,0,0,1,1,0,0,0,0,1
140 DATA 1,0,0,0,0,1,1,1,0,0,0,1,0,0,0,0,0,1
150 DATA 1,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,1,0,1
160 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
170 DATA 1,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,1
180 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1
190 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1
200 DATA 0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0

```

Resoconto evento: Firenze Vintage Bit 2018



a cura dell'Associazione Firenze Vintage Bit Onlus e A. A.

Sabato 24 e Domenica 25 Novembre 2018, si è svolta a Lastra a Signa (FI) la decima edizione del Firenze Vintage Bit, una manifestazione nata nel 2009 da una idea di Walter Pugi e Maurizio Morandi, per riunire gli appassionati di RetroComputing e Storia dell'Informatica.

Nonostante il tempo atmosferico inclemente (ha quasi sempre piovuto) e il contemporaneo svolgimento della Firenze Marathon, con i conseguenti disagi alla viabilità di Firenze, i visitatori non sono mancati e ancora una volta l'evento (a ingresso gratuito e sempre caratterizzato dalla scelta di un "tema" diverso) ha registrato un enorme successo e la soddisfazione degli organizzatori della Associazione Firenze Vintage Bit Onlus presieduta da Walter Pugi.

I membri della Associazione, alcuni dei quali seguiti anche dai propri familiari, hanno dato ciascuno un aiuto per preparare le tre grandi sale all'interno del suggestivo ambiente medioevale dell'Antico "Spedale di Sant'Antonio" di Lastra a Signa, costruito tra il 1416 e il 1421 per volere dei Consoli dell'Arte della Seta, con lo scopo di accogliere quei viandanti che percorrevano la Via Pisana.

Situato in pieno Centro Storico, è oggi sede di convegni ed esposizioni e per l'occasione è stato concesso gratuitamente dal Comune alla Associazione Firenze Vintage Bit Onlus (con Atto di Patrocinio del 12 ottobre 2018).



Oltre a manifesti, targhette e badge di riconoscimento, gli organizzatori hanno provveduto a ottenere dalle competenti Autorità Comunali i permessi necessari per accedere alla Z.T.L. ed effettuare il carico/scarico dei materiali di proprietà degli espositori.

Tramite il sito www.retrocomputer.org, i partecipanti potevano ricevere notizie relative al programma, come raggiungere la destinazione e suggerimenti riguardo alle

strutture ricettive per pernottare nelle vicinanze.

Inoltre nella Sala 3 era stato allestito un angolo "officina" per il gruppo "Restarters Firenze" al fine di garantire piccole riparazioni di emergenza.



La giornata di Sabato 24 era riservata solo agli espositori per installare le macchine e provarle. Tuttavia (previa autorizzazione degli organizzatori) è stato concesso l'ingresso a qualche curioso visitatore che si affacciava per osservare la preparazione delle aree espositive.

Terminato il previsto orario di apertura dei locali della manifestazione, si è proseguito con la tradizionale cena di benvenuto alla Accademia del Coccio di Ponte a Signa, una



occasione per conoscersi meglio e raccontare qualche aneddoto da "lupi del mare informatico".

La giornata del 25 è stata quella aperta al pubblico, che ha potuto osservare una incredibile varietà di cimeli, computer, consolle giochi e sintetizzatori musicali.

Per facilità di spostamento, la maggioranza degli espositori proveniva dalla Toscana (alcuni dei quali appartenenti alla Associazione Firenze Vintage Bit Onlus, Vicoretrò, Museo del Calcolatore "Laura Tellini" di Prato). Il resto d'Italia era rappresentato da espositori provenienti da Torino, Parma, Verona, Vicenza, Roma, Ancona, Perugia.



Segnaliamo chi si è distinto per la presentazione di macchine particolari: "Tania Borealis" (VR) e "Videobros" (VI) che hanno portato varie schede Arcade, Fabrizio Beneforti (PO) con un pesantissimo sistema "American Laser Game" tirato fuori dal garage, "Retroforce" (TO) con gli MSX, Federico Gori (FI) con diversi sistemi video degli anni '80, Robert Swiderski (FI) con una collezione Apple, Giancarlo Oneglio (TO) con i Sega SC-3000, Sergio Massaccesi (AN) con il suo BBC Microcomputer, "8 Bit Inside" con

alcuni sintetizzatori Yamaha, Luca Cusani (PI) con un raro VIC 1001, Marco Fanciulli (GR) che metteva a disposizione attrezzature per Esperienze in Realtà Virtuale del modulo di comando (statico) e dell'allunaggio (dinamico) delle navicelle Apollo.

L'elenco completo degli espositori (più di venti!!!) e le macchine esposte si trova sulla pagina:

<https://www.retrocomputer.org/index.php/fv/b/edizione-2018/elenco-espositori-2018>.

Ad "AlexTheLionet" (assente giustificato) gli auguri per una completa guarigione.

Sempre nella stessa giornata vi sono state due interessanti conferenze, una al mattino e l'altra nel pomeriggio (dopo il pranzo riservato agli espositori).

Quella del mattino ha avuto per tema "10 anni insieme". Dopo il doveroso ringraziamento in pubblico al Sindaco Angela Bagni e all'Assessorato alla Cultura per la concessione dei locali e il patrocinio della manifestazione, Walter Pugi ha illustrato brevemente la storia della manifestazione e i suoi valori: trascorrere insieme momenti di sano divertimento e perpetuare la memoria storica delle macchine che abbiamo utilizzato in passato.

Il resto della conferenza è stato a cura di Leonardo Vettori che, con le sue proverbiali doti di brillante narratore, ha esposto i principali aspetti del variegato mondo del RetroComputing. Con la metafora dei raduni delle auto d'epoca, si va dal "tuning" (ossia la personalizzazione con colori e accessori di più o meno buon gusto) fino alle repliche attuali di vecchi modelli. Un accenno è stato dato agli emulatori che ora si trovano anche sugli smartphone e che permettono anche ai più giovani di sperimentare l'uso dei computer "storici" e il loro software.



Nel pomeriggio Marco Fanciulli ha mostrato un progetto di riproduzione dell' Apollo Guidance Computer, il sistema di navigazione delle navicelle Apollo impiegate per arrivare all'orbita lunare. Gestito al momento dal solo Marco Fanciulli, tramite la documentazione della N.A.S.A. pazientemente cercata e ottenuta, questo titanico progetto vuole raggiungere i seguenti obiettivi: replicare "fisicamente" la postazione di guida di una navicella Apollo con i sistemi computerizzati e offrire esperienze "virtuali" di navigazione simulata (grazie anche alle tracce audio registrate delle conversazioni tra gli astronauti e il Controllo Missione che sono state rese pubbliche).

Per dare un'idea delle difficoltà affrontate, sappiate che l'A.G.C. aveva le memorie a nuclei di ferrite, memorie che Marco Fanciulli ha provato a riprodurre personalmente nel corso del suo progetto giunto al terzo anno.



Dopo la consueta foto di gruppo degli espositori, non poteva mancare il taglio della torta del decennale, decorata con una riproduzione a colori da pasticceria del manifesto 2018 del Firenze Vintage Bit.

Con i brindisi finali, l'augurio di ritrovarsi presto ad altre manifestazioni simili che incominciano ad essere organizzate in altre parti d'Italia, segno di un crescente interesse per la memoria storica dell'Informatica, qualcosa che solo pochi anni fa era considerata una stranezza guardata con un po' di sospetto.

Alla scoperta del “Museo dell’Informatica Funzionante”

di Marco Pistorio

Foto di Luisa Civardi per MusIF

Abbiamo recentemente scoperto che esiste il “Museo dell’Informatica Funzionante”, grazie ad Andrea Milazzo che ci ha messo in contatto con Gabriele “Asbesto” Zaverio, il curatore del Museo. Sicuramente i nostri lettori vorranno saperne di più in proposito, quindi propongo subito a Gabriele alcune domande:

Quando e come è nata questa idea del “Museo dell’Informatica Funzionante”?

G. L’idea parte da molto lontano. Negli anni ‘90 la nostra idea di creare un laboratorio di informatica libero ed aperto al pubblico (ilFreaknet Medialab) e’ stata realizzata grazie all’ospitalita’ del Centro Sociale AURO di Catania, che ci ha fornito una sala dove organizzare una rete di vecchi computer, recuperati da donazioni o discariche. Siamo stati forse tra i primi in Europa a creare un laboratorio libero in cui chiunque poteva usare i computer ed accedere ad internet! Alcuni di questi computer erano talmente vecchi che l’idea di conservarli e farne un Museo e’ nata spontaneamente.

Come siete strutturati? Chi (e/o quanti) sono gli elementi del tuo staff? Ti va di parlarci un po’ di te e di tutti loro?

G. Il gruppo e’ molto vario ed i suoi membri sono sparsi per tutta Europa, poiche’ molti hanno dovuto emigrare per questioni di lavoro. Nonostante la distanza la nostra community è rimasta unita e in molti continuano a supportarci e quando possono partecipano alle attivita’ nella nostra sede. Alcuni membri storici provengono dall’esperienza del Freaknet Medialab di Catania da cui e’ nata in seguito l’associazione culturale “FreakNet” (che oggi gestisce il museo) altri si sono aggiunti man mano. Le competenze sono tra le piu’ disparate: elettronica, informatica, design, ma anche archeologia, chimica e fisica. Abbiamo un comitato scientifico e le decisioni vengono sempre prese in maniera collettiva. Inoltre, da svariati anni siamo gemellati con il Museo Interattivo di Archeologia Informatica (MIAI) di Cosenza curato dalla Ass. Cult. “Verde Binario”: ogni iniziativa o attivita’ che realizziamo e’ progettata e realizzata insieme, tanto che ormai ci consideriamo un Museo unico, con una collezione unica ma conservata in due sedi diverse.

Nel Museo quali e quante macchine ospitate? Potresti parlarci in particolar modo delle macchine più rare?

G. Tra noi e il MIAI di Cosenza ormai conserviamo piu’ di seimila pezzi tra computer, periferiche, schede ed altri oggetti. Inoltre abbiamo una biblioteca di quasi 8000 documenti, parte dei quali censita all’interno del Servizio Bibliotecario Nazionale (SBN). Tra i pezzi piu’ rari o particolari c’e’ sicuramente il General Electric GE-120, un sistema informatico completo che e’ attualmente conservato a Cosenza. Si tratta di un enorme computer del 1969, originariamente in uso presso l’aeroporto di Zurigo, il cui recupero e’ stato possibile grazie ad un crowdfunding ed ha coinvolto 2 camion per trasportare le 5 tonnellate di hardware dalla Svizzera al sud Italia. Altri pezzi interessanti sono sicuramente un sistema Nuova Elettronica del 1980, e la nostra ricostruzione funzionante del computer APPLE 1, realizzato con componentistica d’epoca.

Il Museo si limita solo a raccogliere e catalogare hardware oppure no? Ci spieghi meglio in cosa consiste l’attività del Museo, più a 360 gradi, come funziona?

G. L’attivita’ e’ quella di un Museo tradizionale ma moderno. Ovviamente, lo scopo principale di un Museo e’ quello di conservare



i pezzi nel miglior modo possibile e curarne l’esposizione. Nel nostro caso questo avviene per le due collezioni di Palazzolo Acreide e di Cosenza e per mostre temporanee che allestiamo in giro per l’Italia e nel resto del mondo. L’ultima mostra che abbiamo organizzato, una retrospettiva sulla computer art che abbiamo chiamato “bin/art” (<https://binart.eu>), si e’ svolta a Roma lo scorso Maggio. Per le nostre esposizioni abbiamo sempre scelto di evitare il classico allestimento con pezzi “spenti” dietro una teca di vetro, preferendo un approccio interattivo: i pezzi esposti sono solitamente funzionanti, utilizzabili dai visitatori, seppur entro certi limiti e con la nostra supervisione. L’approccio della nostra ricerca e’ multidisciplinare, non si tratta soltanto di “collezionare” computer: siamo interessati a ogni aspetto della storia delle tecnologie informatiche, in particolare al lato antropologico, a tutto cio’ che riguarda il contesto storico, le esperienze di chi ha usato i pezzi in questione, le eventuali personalizzazioni, note, aneddoti, appunti, software, materiale fotografico, interviste

audio/video eccetera. Tra le attivita’ piu’ importanti ovviamente ci sono i lavori di restauro, eseguiti sempre con precisi criteri filologici (senza assolutamente snaturare il pezzo da restaurare, cercando di





"conservarlo" nello stato attuale anziché riportarlo alle condizioni originali) che riguardano non solamente i computer e le loro periferiche, ma anche la documentazione ed il recupero dati dai media originali (dischi, nastri, schede di memoria, etc). Altri aspetti delle nostre attività di ricerca riguardano gli effetti dell'invecchiamento chimico/fisico sulla componentistica elettronica, gli studi chimici sul deterioramento delle plastiche ed eventuali contromisure, nonché le ricostruzioni storiche di computer, periferiche o software.

Raccontaci degli obiettivi che avete già raggiunto e quali sono i vostri prossimi obiettivi, i vostri programmi a breve, a medio ed a lungo termine.



G. Al momento l'obiettivo primario è trovare una sede adeguata ad ospitare entrambe le collezioni del MusIF e del MIAI di Cosenza. Le speranze al momento sono rivolte alla Calabria, forse qualcosa si muoverà presto.

Quali sono le spese che è necessario fronteggiare per portare avanti il Museo?

G. Tutte le spese vive sono quelle relative all'affitto degli stabili (che abbiamo dovuto affittare per nostro conto dopo aver cercato per anni di ottenere aiuti istituzionali che non sono mai giunti) nonché tutte le utenze necessarie (elettricità, linea dati, acqua, riscaldamento, tasse comunali etc.)

Come fronteggiate tali spese per mantenere il Museo attivo e fruibile?

G. Al momento non abbiamo nessuno sponsor né alcun aiuto istituzionale, per cui le nostre uniche entrate derivano da collette che facciamo mensilmente tra i membri delle associazioni FreakNet e Verde Binario, e da donazioni spontanee che ci vengono di tanto in tanto fatte, specie in periodi di crisi economica. Inoltre, periodicamente produciamo dei gadget che distribuiamo a chi ci rilascia una donazione liberale: attualmente chiunque può sostenerci richiedendo una copia del nostro calendario fotografico 2019, interamente autoprodotta con foto originali dei pezzi esposti nella mostra 'bin/art' di cui dicevamo prima, il pdf del calendario e le istruzioni per ricevere una copia cartacea si trovano all'indirizzo di seguito indicato: <https://binart.eu/calendario-2019/>

Cercate nuovi collaboratori? Chi collabora con voi viene pagato oppure si tratta di collaborazioni a titolo volontario e gratuito?

G. Si tratta, purtroppo, sempre di volontariato perché al momento non possiamo permetterci di retribuire nessuno. Le collaborazioni nascono spontaneamente dall'entusiasmo relativo ad una nuova sfida tecnologica.

Chi intendesse collaborare con voi come potrebbe contribuire? Fornendo dell'hardware? Effettuando riparazioni? O per quali altre attività? Documentazione? Catalogazione? È possibile contribuire a sostenere le spese del Museo, anche simbolicamente? E se sì, come?

G. Qualsiasi tipo di donazione di materiale storico è ovviamente ben accetta; per problemi di spazio ovviamente dobbiamo valutare cosa conservare e cosa invece non possiamo ricevere. Se qualcuno volesse darci una mano con le riparazioni, ovviamente è il benvenuto! Donazioni economiche sono sempre gradite perché ci permettono di pagare le bollette.

Come è possibile effettuare una visita al Museo? Le visite sono libere oppure è necessario un contributo/ticket per poter accedere ai locali del Museo? Bisogna contattare preventivamente qualcuno (e se sì come) oppure no? Avete dei giorni/orari prestabiliti per le visite?

G. Al momento le visite si effettuano solamente su prenotazione, genericamente di mattina. Non c'è un biglietto di ingresso: abbiamo preferito lasciare liberi i visitatori di donare qualcosa all'uscita del Museo, cosa che spesso si rivela molto più vantaggiosa perché chi resta contento della visita è ben felice di supportare il nostro lavoro.

Ti ringrazio per aver risposto a queste nostre domande. Ci auguriamo che il Museo dell'Informatica Funzionante continui a lungo la sua attività e che diventi un punto di riferimento sempre più importante nel panorama del Retrocomputing, sia all'interno dei nostri confini che a livello internazionale. In bocca al lupo a te ed a tutto il tuo staff!

G. Grazie, faremo del nostro meglio!

Riferimenti:

<https://museo.freaknet.org>
<https://www.verdebinario.org>

MANUALE - INTRODUZIONE ALL'ARCHITETTURA DEL C64:

PROGRAMMABLE CHARACTERS, CHARACTER SET, CHARACTER GENERATOR ROM, SCREEN MEMORY, VIC-II REGISTERS, MEMORY BANK, CIA REGISTERS

di Attilio Capuozzo - Admin Fondatore Gruppo FB "8 Bit Retroprogramming Italia"



Il presente Manuale nasce da un desiderio personale di descrivere, in maniera sufficientemente dettagliata, argomenti relativi a diversi aspetti della programmazione dell'Architettura del C64 che, solitamente, nella bibliografia specializzata dedicata al mitico 8 bit della Commodore o sono spiegati poco approfonditamente oppure per contro la trattazione risulta troppo tecnica e dunque poco chiara.

Il mio sincero auspicio è che questo mio lavoro, frutto di profonda passione, possa essere apprezzato dai lettori.

Nota Bene: A vantaggio soprattutto dei Programmatori meno esperti, nel Manuale ho cercato di sottolineare e ribadire più volte i concetti esposti a costo anche di sembrare ripetitivo.

Inoltre per quanto riguarda la terminologia tecnica, ho cercato di usare molti sinonimi sia in lingua inglese che in italiano, in modo da rendere il più agevole possibile, in particolare ai neofiti, la successiva lettura di libri e articoli dedicati alla Programmazione del C64.

Infine, per comodità del lettore ho allegato la Tabella dei Codici ASCII/PETSCII ("ASCII AND CHR\$ CODES"), la Tabella degli SCREEN CODES e il Programmable Character Worksheet presenti nella **COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE**, la Bibbia del Programmatore del C64 realizzata dalla stessa Commodore, un TESTO FONDAMENTALE e IMPRESCINDIBILE da cui ha attinto a piene mani la maggior parte della bibliografia tecnica oggi esistente sul C64 nonché la stessa C64-WIKI che in più occasioni riporta esempi ripresi tal quale dalla **PROGRAMMER'S REFERENCE GUIDE**.

Ho inoltre allegato un esempio di Programmable Character complesso ottenuto con l'affiancamento di più Caratteri in un'unica grande Griglia. L'immagine è tratta da un altro Must Have, il libro **COMMODORE 64 GRAPHICS AND SOUND PROGRAMMING** scritto da STAN KRUTE.

Infine troverete la schematizzazione della Mappa di Memoria del C64 riportata in uno dei più dettagliati libri sul BASIC del C64, **COMMODORE 64: IL BASIC** di Rita Bonelli, pubblicato dal Gruppo Editoriale Jackson.

Vi invito a segnalarmi eventuali refusi o imprecisioni a: attilio.capuozzo@gmail.com.

Inoltre mandatemi vostri contributi di Tecniche di Programmazione, dettagliamente illustrate e spiegate, relative agli argomenti del presente Manuale, in modo tale che le inserirò (citando ovviamente le fonti) in una prossima edizione del Manuale.

Iniziamo a vedere innanzitutto il corpo principale del programma in cui effettueremo:

1) Copia parziale o completa di uno dei 2 Character Set del C64 dalla Character Generator ROM a una specifica area di RAM (Character Memory)

2) Riprogrammazione dei nostri Caratteri

Successivamente analizzeremo le linee di apertura e chiusura del programma.

Prima di spiegare i punti 1 e 2, vediamo sinteticamente cosa è la ROM Caratteri:

la Character Generator ROM occupa 4K (4096 byte) a partire dall'indirizzo 53248 (\$D000) e contiene la descrizione di tutti i Caratteri stampabili del C64 (512 Caratteri suddivisi in 2 Set di 256 Caratteri ciascuno).

La descrizione (Pattern) di ogni Carattere è contenuta in una Griglia (Character Matrix) di 8*8 bit pari a 8 byte.

I bit della Matrice Carattere corrispondono ad altrettanti punti (dots) sullo schermo ossia ai pixel.

Ogni Carattere è individuato da uno specifico Codice (Screen Code) contenuto nella Memoria Schermo (Screen Memory).

Il simbolo "@" occupa i primi 8 byte della ROM Caratteri (da 53248(\$D000 a 53255(\$D007) e ha 0 come valore di Screen Code. Nei successivi 8 byte, da 53256 (\$D008) a 53263 (\$D00F), è presente il Pattern della "A" maiuscola con Screen Code 1 e così via (vedi Tabella degli SCREEN CODES allegata).

SCREEN CODES

SET 1	SET 2	POKE	SET 1	SET 2	POKE	SET 1	SET 2	POKE
@		0	C	c	3	F	f	6
A	a	1	D	d	4	G	g	7
B	b	2	E	e	5	H	h	8
SET 1	SET 2	POKE	SET 1	SET 2	POKE	SET 1	SET 2	POKE
I	i	9	%		37		A	65
J	j	10	&		38		B	66
K	k	11	'		39		C	67
L	l	12	(40		D	68
M	m	13)		41		E	69
N	n	14	*		42		F	70
O	o	15	+		43		G	71
P	p	16	,		44		H	72
Q	q	17	-		45		I	73
R	r	18	.		46		J	74
S	s	19	/		47		K	75
T	t	20	0		48		L	76
U	u	21	1		49		M	77
V	v	22	2		50		N	78
W	w	23	3		51		O	79
X	x	24	4		52		P	80
Y	y	25	5		53		Q	81
Z	z	26	6		54		R	82
[27	7		55		S	83
]		28	8		56		T	84
↑		29	9		57		U	85
←		30	:		58		V	86
→		31	;		59		W	87
SPACE		32	<		60		X	88
↑		33	=		61		Y	89
↓		34	>		62		Z	90
#		35	?		63			91
\$		36			64			92

SET 1	SET 2	POKE	SET 1	SET 2	POKE	SET 1	SET 2	POKE
		93			105			117
		94			106			118
		95			107			119
		96			108			120
		97			109			121
		98			110		<input checked="" type="checkbox"/>	122
		99			111			123
		100			112			124
		101			113			125
		102			114			126
		103			115			127
		104			116			

Codes from 129-255 are reversed images of codes 0-127.

Come vedremo meglio in seguito, ridefinire ad esempio la "@" vorrà dire sostituirla con un proprio carattere personalizzato descritto da una nuova Matrice di 8 Righe(Byte)*8 Colonne(bit/pixel) da noi specificamente creata.

Le istruzioni propedeutiche alla copia del contenuto della ROM Caratteri sono le seguenti:

bisogna disabilitare gli Interrupt di Sistema (IRQ) con:

POKE 56334, PEEK(56334) AND 254

L'istruzione va a resettare il bit 0, il contatore generatore dell'Interrupt (Timer A), del Registro 14 (56334/\$DCoE) del CIA #1 disabilitando di fatto l'operazione di scansione della tastiera effettuata 60 volte al secondo (Turn Off KeyScan Interrupt Timer).

Successivamente bisogna rendere visibile alla CPU (MOS 6510) la ROM Generatrice dei Caratteri (Character Generator ROM), a cui normalmente può accedere solo il Chip VIC-II, con l'istruzione:

POKE1,PEEK(1) AND 251

La POKE resetta il bit 2 (CHAREN bit) della locazione 1/\$0001, ossia il Registro di INPUT/OUTPUT della CPU (il Microprocessore 6510), e come conseguenza si ha uno Switch Out, dallo spazio di indirizzamento del Microprocessore, dei Registri di Controllo dell'I/O detti anche Registri di Controllo Periferiche o PCR's (ossia i Registri del VIC-II, del SID, ecc. oltre anche alle locazioni della Color RAM) e uno Switch In (nell'Address Space della CPU) della Character Generator ROM che pertanto occuperà l'area di memoria che va da 53248/\$D000 a 57343/\$DFFF (2 Character Set di 2K = 2048 byte ciascuno per un totale di 4K=4096 byte).

Dunque nell'area di memoria che inizia da 53248/\$D000 abbiamo la coesistenza sia di 4K di RAM INPUT/OUTPUT (accessibile di default dalla CPU) che di 4K "nascosti" della Character Generator ROM a cui il Processore del C64 non può normalmente accedere.

La Mappa della Memoria del C64 prevede una configurazione a "strati"(layer) dove il layer superiore nasconde quello inferiore (complessivi 20K di Memoria Nascosta) e all'occorrenza tramite un opportuno "bank switching" lo strato inferiore può essere reso visibile alla CPU 6510 del 64 (ossia inserito nello spazio di indirizzamento del

Microprocessore) tramite uno scambio logico con il layer di livello più alto.

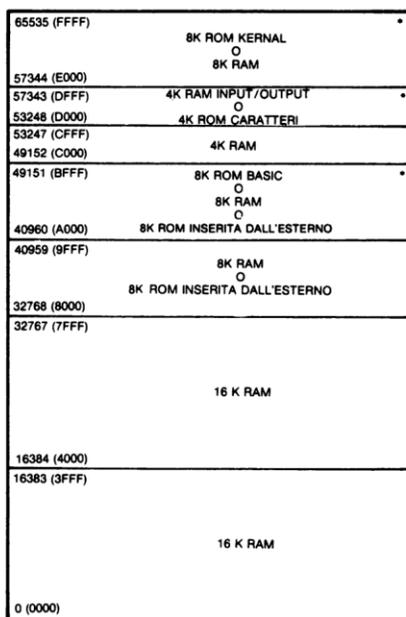


Figura 8.2 Mappa della Memoria

In questo modo si superano i limiti di indirizzamento della CPU (bus indirizzi a 16 bit) che può indirizzare massimo 64K (65536 bytes) a fronte di una Mappa di Memoria del C64 che prevede complessivamente 64K di RAM più 20K di ROM (Interprete BASIC, KERNAL, ossia il SISTEMA OPERATIVO del C64, e Character Generator ROM).

La Mappa di Memoria del C64 è schematizzata in Fig. 8.2 tratta dal libro COMMODORE 64: IL BASIC di Rita Bonelli (Gruppo Editoriale Jackson).

Quindi si va a caricare il CharSet dei Caratteri dalla ROM alla RAM per 2048 byte consecutivi scegliendo il SET 1 Maiuscolo/Grafico o il SET 2 Minuscolo/Maiuscolo e la locazione di partenza della RAM che ospiterà la copia della ROM dei caratteri.

L'Indirizzo Base di quest'area di RAM rappresenterà lo Starting Address della Character Memory RAM.

Un'ottima area di memoria RAM per salvare i Caratteri ricopiati dalla Character Generator ROM, come anche i dati per la Definizione degli Sprite (Sprite Data Pattern), nonché per lasciare al programma BASIC un adeguato spazio di memoria, è quella che inizia dalla locazione 12288 (\$3000) in poi, sempre

rimanendo nell'ambito del Banco di Memoria Numero 0 che è quello di default e che rappresenta i primi 16384 byte della RAM del C64 come spiegherò meglio in seguito.

La locazione scelta per la nuova Character Memory RAM deve essere un multiplo di 2048.

Nel mio caso ho scelto la locazione di memoria 14336 (\$3800) che è, appunto, un multiplo di 2048 ed è compresa nel range 12288/\$3000 - 16384/\$4000 ed è anche l'ultima locazione disponibile nel Banco 0 per la Character Memory (infatti 14336+2048=16384).

Tornando ai Character Set, vi ricordo che in DIRECT MODE (Modo Diretto o Immediato) per switchare da un SET all'altro basta premere contemporaneamente i tasti CBM e SHIFT.

Se si è scelto, per esempio, il SET 1 Maiuscolo/Grafico si dovrà inserire, in un opportuno ciclo FOR/NEXT da 0 a 2047, una POKE che valorizzi le locazioni RAM a partire dalla 14336/\$3800 (incrementata della variabile contatore del ciclo FOR) con il contenuto (PEEK) delle locazioni di memoria a partire dalla 53248/\$D000 (+ variabile contatore):

```
FOR J=0 TO 2047:POKE (14336+J),PEEK(53248+J):NEXT J
```

*Nella pratica sarebbe opportuno assegnare il 2 Starting Address, quello della Character Memory RAM (14336) e della Character Generator ROM (53248), ad altrettante Variabili Numeriche da dichiarare in testa al programma e da utilizzare nella POKE e nella PEEK del ciclo FOR/NEXT per un'ottimizzazione del codice.

Nel caso si fosse scelto il SET Minuscolo/Maiuscolo (SET 2), la locazione di partenza da cui copiare i profili sarebbe stata la 55296/\$D800 (53248+2048) in quanto ogni SET occupa 2K ossia 2048 byte e contiene 256 Caratteri di 8 byte ciascuno.

È ovvio che il Range dei Caratteri da ricopiare dalla Character Generator ROM alla RAM non deve necessariamente contemplare un intero CharSet da 2K (2048 byte) ma può essere scelto a piacimento.

A puro titolo esemplificativo potremmo voler ricopiare i primi 64 Caratteri del Character Set creando un loop FOR/NEXT che va da 0 a 511 per un totale di 512 byte (64 Characters*8 byte). Allo stesso modo potremmo avere la necessità di ricopiare i primi 128 Caratteri di uno dei 2 CharSet del 64 (SET 1/SET 2) per complessivi 1024 byte da copiare con un Ciclo che va da 0 a 1023 e così via (128 Caratteri*8 byte).

Dopo aver effettuato la copia dei Caratteri, bisogna mascherare di nuovo alla CPU la ROM dei Caratteri riabilitando, dunque, le funzioni di I/O con:

POKE1,PEEK(1) OR 4

che di fatto effettua l'operazione inversa alla precedente ossia Switch Out della Character Generator ROM e Switch In dei 4K di I/O RAM e poi riattivare il Timer dell'Interrupt di Scansione della Tastiera (Turn On KeyScan Interrupt Timer) con:

POKE 56334,PEEK(56334) OR 1

Ultima operazione da fare, prima di creare i propri caratteri personalizzati, è dire al Chip VIC-II di cercare i profili dei Caratteri non più nella Character Generator ROM ma a partire dal nuovo indirizzo RAM (14336/\$3800) attraverso l'istruzione:

POKE 53272,31

pertanto da questo momento in poi il CharSet customizzato memorizzato in RAM sostituirà completamente i Caratteri Standard del C64 le cui informazioni sono presenti nella Character Generator ROM. Quindi dobbiamo fare attenzione a copiare nella nostra Character Memory RAM tutti Numeri, le Lettere e i Caratteri Grafici, di uno dei 2 CharSet Standard del 64, che intendiamo usare.

Il registro 24 del VIC-II mappato all'indirizzo 53272 (\$D018) imposta sia l'Indirizzo di Base della Character Memory RAM che quello della Screen Memory.

Il VIC-II (Video Interface Chip MOS 6567) è il Chip di Interfaccia Video che con il SID - Sound Interface Device, il Chip Sonoro - ha rappresentato una sorprendente innovazione rispetto ai precedenti integrati in quanto,

peraltro, poteva accedere alla memoria del C64 indipendentemente dalla CPU.

Con i bit 1/2/3 si può settare la locazione iniziale della Character Memory RAM (Valore Tripletta bit*2048) mentre con i bit di posizione 4/5/6/7 - dunque il Nibble (o semibyte) alto o più significativo - si può impostare la locazione di partenza della Screen Memory (Valore Nibble*1024).

*Il valore di default di questi bit è in binario 0001 corrispondente, dunque, a $1*1024=1024/0400$ che è appunto la locazione iniziale di default della RAM Video o Screen Memory.

Il bit 0 è inutilizzato.

Per completezza va detto che se si decide di modificare la posizione di inizio della zona di memoria dove è allocata la Screen Memory bisogna comunicarlo oltre che al VIC-II tramite il registro 53272/\$D018 (come appena visto) anche al Sistema Operativo andando a POKARE nella locazione 648 (\$0288) il Byte Alto dell'Indirizzo della RAM Video. In pratica il valore da immettere nel puntatore 648 (\$0288) si calcola dividendo per 256 l'Indirizzo del primo byte della Screen Memory (dunque andremo a individuare la "Pagina" della Mappa di Memoria dello Schermo):

POKE 648, (Locazione Inizio Screen Memory/256)

Anche nel caso in cui selezioniamo un altro Banco di Memoria (tramite il Registro 56576/\$DD00 del CIA #2), sebbene l'Indirizzo Base della Screen Memory si sposterà AUTOMATICAMENTE, dobbiamo comunicare al SO la nuova ubicazione in memoria POKANDOLA nella locazione 648/\$0288 (in tal caso NON abbiamo bisogno, ovviamente, di comunicarlo anche al VIC-II tramite il Registro 24 mappato all'indirizzo 53272).

Vi ricordo che l'Indirizzo di Inizio della Memoria Video si calcola con la seguente formula:

$IMV = (\text{Numero Banco} * 16384) + HN * 1024$

dove HN rappresenta il Nibble Alto (i bit dalla posizione 4 alla 7) del Registro 53272/\$D018 del VIC-II.

Si può anche affermare che, di DEFAULT, la Screen Memory, che occupa 1 Kb, inizia SEMPRE 1024 byte dopo l'Indirizzo di Inizio del Banco selezionato e gli 8 Puntatori agli Sprite iniziano SEMPRE 1016 byte dopo l'Indirizzo di Base della Memoria Video e dunque essi occupano gli ultimi 8 byte dei 1024 riservati in totale alla Screen Memory.

La Memoria Schermo effettivamente utilizzabile per la stampa dei caratteri è rappresentata dai primi 1000 byte (Default Banco 0: 1024/\$0400 - 2023/\$07E7) in quanto, in Modalità Testa (Text Display), lo Schermo del C64 è costituito da 25 Righe * 40 Colonne = 1000.

Nel Banco N.0, che è quello di default, la Screen Memory inizia, infatti, dalla locazione 1024 e i Puntatori agli Sprite sono locati da 2040/\$07F8 a 2047/\$07FF.

Ora vediamo perché abbiamo inserito (più sopra) il valore decimale 31 nel Registro 53272/\$D018 tramite la POKE:

in rappresentazione binaria il valore decimale 31 (00011111) imposta (tra gli altri) i bit 1/2/3 al valore decimale 7 che moltiplicato per 2048 ci dà, appunto, il valore 14336/\$3800 che corrisponde all'indirizzo della prima locazione RAM in cui è stata ricopiato il contenuto completo o parziale della Character ROM (in altre parole 14336/\$3800 rappresenta la Starting Location del nostro Character Set ossia il nuovo Indirizzo della Character Memory RAM).

Per completezza, avrei anche potuto scrivere l'istruzione POKE nella seguente forma analoga alla precedente facendo ricorso agli Operatori Logici per mascherare i bit con l'AND e successivamente sommarli con l'OR;

POKE 53272,(PEEK(53272)AND240)OR14

oppure si può utilizzare anche quest'altra forma (usata spesso negli esempi presenti in Rete):

POKE 53272,(PEEK(53272)AND240)+14

dove si va a sostituire all'operatore logico OR l'operatore algebrico + , sortendo lo stesso effetto ossia quello di settare, attivare (porre a 1) determinati bit lasciando inalterati gli altri

(l'OR, infatti, è definito anche come "somma logica").

A mio avviso la forma più leggibile è quella che fa interamente ricorso agli Operatori Logici. D'altronde con l'uso di AND e OR è più facile capire l'esatta posizione dei bit su cui si va ad agire!

Se la Starting Location della Character Memory RAM fosse stata la 12288 (\$3000) avremmo dovuto scrivere la seguente istruzione:

```
POKE 53272,(PEEK(53272)AND240)OR12
    oppure
```

```
POKE 53272,29
```

A questo punto si possono ridefinire i Caratteri scelti POKANDO, per ogni Carattere da personalizzare, i valori decimali degli 8 byte del profilo del Carattere ridefinito a partire dal corrispondente Indirizzo RAM della prima riga della matrice del Carattere customizzato così calcolato:

```
Starting Location Character Memory RAM
(14336)+SC*8
```

dove SC è lo SCREEN CODE, il Codice dello Schermo o Codice Video (in alcuni testi indicato anche con DISPLAY CODE o D-CODE), da NON confondere con il Codice ASCII, del Carattere ridefinito.

La Tabella dei Codici dello Schermo (Screen Codes) la potete anche consultare nell'Appendice E a pag.132 del Manuale d'Uso del C64 oltre a trovarla come allegato al presente Manuale.

Tornando alla precedente formula, vorrei ricordare che per Indirizzo Base della Character Memory RAM NON si intende un Indirizzo Assoluto ma è sempre un Indirizzo Relativo nell'ambito del Banco selezionato (Range tra 0 e 14336 in multipli di 2048); se decidiamo di cambiare Banco rispetto al default (Banco N.0) dobbiamo ricordarci di sommare, nella precedente formula, anche il Bank Address calcolato come segue:

```
BANK ADDRESS = Numero Banco*16384.
```

Il profilo di ogni Carattere è contenuto in una Matrice di 8 righe lunghe ciascuna 8 bit per un totale di 8 byte.

Ogni byte della Matrice rappresenta un dot pattern della Riga del Carattere e ogni bit rappresenta un dot (e dunque un pixel sullo Screen Display).

Una buona tecnica di programmazione è quella di creare 2 Cicli FOR/NEXT annidati: un Ciclo Esterno che stabilisce il range degli Screen Codes dei Caratteri da programmare e quello interno che, per ogni Screen Code, cicla da 0 a 7 (dunque per 8 byte) leggendo i Dati dei Caratteri con un opportuno READ/DATA e memorizzandoli con una POKE in RAM, byte per byte, tramite la formula vista più sopra:

```
FOR SC = 10 TO 20:REM RANGE SC 10 THRU 20
  FOR BYTE = 0 TO 7
    READ NUMBER
    POKE 14336+(SC*8)+BYTE,NUMBER
    NEXT BYTE
  NEXT SC
```

Gli 8 valori decimali per ogni Carattere ridefinito da inserire negli statement DATA sono strettamente correlati, ovviamente, alla definizione del Carattere.

Vediamo come avviene la costruzione di un Carattere secondo una tecnica piuttosto semplice: ogni Carattere di un CharSet del C64, come dicevo, è memorizzato in una matrice di 8 Righe*8 Colonne dove ogni Riga corrisponde a un byte e gli 8 bit (Colonne), di cui è costituito ogni byte (Riga), rappresentano i singoli pixel del Carattere.

Come riferimento prendiamo il Character Programmable Worksheet riportato a Pag.112 della **COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE** che allego.

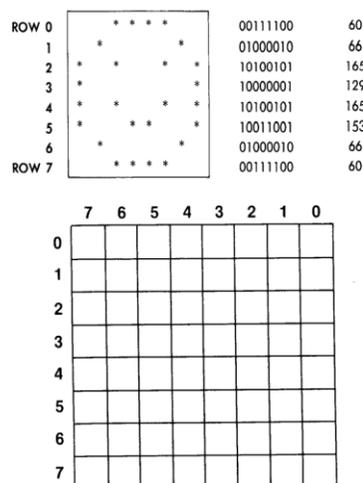


Figure 3-1. Programmable Character Worksheet.

Il Worksheet, che rappresenta la Matrice 8*8 del Carattere che vogliamo creare per sostituirlo a uno dei 256 Characters di uno dei 2 CharSet del C64, è costituito da 8 Righe (Rows) numerate da 0 a 7 dall'alto verso il basso e da 8 Colonne (Columns) numerate sempre da 0 a 7 da destra verso sinistra.

Procedendo dall'alto verso il basso e da sinistra verso destra, per ogni Riga laddove vogliamo che nella Colonna considerata il corrispondente pixel sia acceso/on (in modo da visualizzare sullo Schermo un punto pieno o dot) poniamo a 1 il bit corrispondente mentre lasceremo a 0 il bit se vogliamo visualizzare uno spazio (blank) ossia un pixel spento/off che ha il Colore dello Sfondo dello Schermo (Background Color).

In pratica se ci cimentiamo nella definizione manuale della Forma (Shape) del Carattere personalizzato, facendo uso del suddetto Worksheet proposto dalla PROGRAMMER'S REFERENCE GUIDE, possiamo a piacimento riempire le Colonne delle varie Righe o con una "X" oppure magari annerirle a matita.

Il primo bit da sinistra (il bit di posizione 7 che è definito anche come bit più significativo o MSB = Most Significant Bit) se posto a 1 (settato) vale 2^7 cioè 128, il secondo bit sempre da sinistra (il bit 6) se settato corrisponde a 2^6 (=64) e così via fino ad arrivare al bit 0 (ossia il primo bit più a destra detto anche bit meno significativo o LSB = Less Significant Bit) che vale 1 (2^0) se posto a 1.

I numeri delle Colonne del Character Programmable Worksheet rappresentano, peraltro, le potenze di 2 come detto prima.

La somma algebrica del valore dei singoli bit settati di ogni Riga della Matrice del Carattere ci darà il valore decimale da inserire nella riga DATA così come chiaramente illustrato nella citata Pag.112 della **COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE**.

Dunque andremo a sommare il corrispondente valore numerico delle Colonne riempite con una "X" o annerite, ossia dei bit/pixel posti a 1, di ogni Riga della Matrice procedendo dalla prima in alto (la Row 0) fino all'ultima in basso (la Row 7) ottenendo, pertanto, gli 8 valori decimali che andremo a inserire nell'istruzione DATA

relativa al Carattere che abbiamo scelto di Riprogrammare.

Per ogni Riga della Matrice potremo ottenere un valore decimale compreso tra 0 (tutti i bit/pixel spenti/off) a 255 (tutti i bit/pixel accesi/on) che è appunto il range rappresentabile da 1 byte (256 valori numerici che vanno da 0 a 255). Nella fase di creazione del Carattere è consigliabile che le linee verticali e diagonali della Forma del Carattere (Character Shape) siano larghe almeno 2 punti in modo da essere chiaramente visibili sullo Schermo.

Con questa tecnica possiamo anche definire caratteri più complessi creando griglie più grandi che sono il risultato dell'aggiunta di altri Caratteri aventi, preferibilmente, valori consecutivi di Screen Code: come esempio potremmo pensare di creare una griglia di 2*3 Caratteri ridefinendo le prime 6 lettere dell'alfabeto dalla A alla F (Screen Code da 1 a 6) per creare un'unica immagine (vedi Allegato tratto dal libro **COMMODORE 64 GRAPHICS AND SOUND PROGRAMMING** di STAN KRUTE)

Character Graphics

	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	
198																	48
98																	24
48																	140
63																	252
32																	4
32																	100
32																	103
32																	4
33																	132
48																	252
24																	24
12																	48
7																	224
3																	192
2																	64
3																	192
1																	192
1																	128
0																	128
1																	128
3																	128
7																	0
15																	0
7																	224

Fig. 4-8. An alien creature drawn on a grid that's two characters wide and three characters high.

Come prime 2 linee di programma bisognerebbe effettuare le seguenti operazioni:

- 1) Impostare il Character Set che si intende utilizzare (SET 1 Maiuscolo/Grafico o SET 2 Minuscolo/Maiuscolo)
- 2) Riservare la Memoria per la copia parziale o completa di uno dei 2 Character Set del C64 dalla Character Generator ROM alla Character Memory iRAM attraverso la modifica dei

Puntatori della RAM riservata al Programma BASIC

Vediamo in dettaglio come fare:

per quanto riguarda lo switch a uno dei 2 CharSet del C64, faremo uso dei Caratteri di Controllo, i Character Based Commands, della Tabella ASCII/PETSCII (riportata anche in **Appendice F a pag.135 del Manuale d'Uso del C64** oltre che come allegato al presente Manuale) per andare ad attivare l'uno o l'altro dei 2 Insiemi di Caratteri.

I 2 Character Set possono essere attivati, ovviamente, alternativamente e non contemporaneamente.

Nello specifico andremo a scrivere la seguente istruzione per attivare il SET 1 Maiuscolo/Grafico (che è quello di default all'accensione del C64):

```
10 PRINT CHR$(142)
```

Per attivare, invece, l'altro Insieme di Caratteri (quello detto Minuscolo/Maiuscolo o SET 2) scriveremo:

```
10 PRINT CHR$(14)
```

Per completezza di informazione va detto che i Caratteri di Controllo di cui sopra, al contrario della maggior parte dei Character Based Commands che controllano i colori del testo, il cursore e lo schermo, NON possono essere scritti con uno statement PRINT che fa uso del MODO VIRGOLETTE o QUOTE MODE.

Allo scopo, invece, di riservare adeguata memoria per i Caratteri Programmabili, si devono impostare i puntatori 55/56 (\$0037/\$0038) che modificano il limite della memoria RAM disponibile per il programma BASIC del 64.

In questi casi è buona norma di programmazione ricordarsi di andare sempre a modificare all'inizio del programma la locazione limite entro cui può essere memorizzato il programma BASIC in modo che il programma stesso e le sue variabili non "sporchino" (sovrascrivano) eventualmente la Character Memory RAM e i Dati di Definizione degli Sprite (Sprite Data Pattern), laddove per necessità abbiamo deciso di memorizzare

questi elementi grafici in un'area di RAM compresa tra le locazioni 2049 (\$0801) e la 4096 (\$A000) che rappresentano le locazioni di inizio e fine della Memoria riservata al Programma BASIC.

Per completezza d'informazione va detto che i blocchi relativi ai dati degli Sprite devono partire da una locazione il cui indirizzo deve necessariamente essere un multiplo di 64, e inoltre i byte di definizione di tutti gli elementi grafici utilizzati dal nostro programma devono rientrare nello stesso banco di memoria da 16 Kb, ma questa è un'altra storia...!

In concreto, dopo aver impostato il CharSet come visto sopra, bisogna abbassare il Top della memoria BASIC facendo puntare la coppia di locazioni 55/56 (\$0037/\$0038) al nuovo Indirizzo di Memoria in cui abbiamo effettuato la copia parziale o full del Character Set del C64 - lo Starting Address della Character Memory RAM - oppure, nel caso degli Sprite, all'Indirizzo a partire dal quale abbiamo memorizzato gli Sprite Data Pattern.

I suddetti registri (55/56) sono nel formato Byte Basso (LB)/Byte Alto (HB) e andrebbero impostati al valore 14336 (Indirizzo Base della Character Memory RAM come da me scelto):

```
20 POKE 55,0:POKE 56,56:CLR
In fatti:
0+(56*256)=14336
```

Alla fine del programma, prima dell'istruzione END, dobbiamo innanzitutto ricordarci di POKARE nel registro 53272/\$Do18 il valore decimale 21 in modo da riportare il Char Pointer al valore di default:

```
POKE 53272,21
```

La POKE imposterà a 1 il bit 2 della tripletta di bit di posizione 1/2/3 (come abbiamo già visto) ripristinando pertanto il valore di default della Character Memory Standard del C64 che nel Banco 0 è compresa nel seguente range di indirizzi: 4096/\$1000 - 8191/\$1FFF; se ci trovassimo invece nel Banco 2 troveremmo la Character Memory Standard alle locazioni 36864/\$9000 - 40959/\$9FFF.

Le suddette Character Memory sono delle "immagini" in RAM della Character Generator

ROM (ROM Image) a uso esclusivo del VIC-II nel Banco 0 e nel Banco 2; nei Banchi 1 e 3 la ROM Image NON è disponibile, cioè il sistema NON effettua un'operazione di "Imaging" della Character Generator ROM dal suo indirizzo ROM 53248/\$D000 a specifiche aree di RAM nel Banco 1 e nel Banco 3 come detto.

Pertanto i Banchi di Memoria 1 e 3 NON possono essere usati per la creazione di Caratteri Programmabili.

Quando la ROM Caratteri deve essere letta dalla CPU, si fa lo switch con i Registri di Controllo dell'I/O (Switch Out) tramite la POKE nella locazione 1 (come si è spiegato ampiamente all'inizio) e la Character Generator ROM sarà disponibile alla CPU a partire dall'Indirizzo 53248/\$D000 (Switch In).

L'altra istruzione da inserire nel programma, subito dopo aver ripristinato il valore di default del Registro 53272/\$D018 e poco prima della END, è l'impostazione al valore di default della coppia di puntatori della RAM BASIC:

l'area di Memoria disponibile per il programma BASIC, di default, va dalla locazione 2049/\$0801 - puntata dalla coppia di locazioni 43/44 (\$002B/\$002C) - alla 40960/\$A000 - puntata dalle locazioni 55/56 (\$0037/\$0038) - per un totale, dunque, di circa 38Kb.

Allo scopo di riportare il limite superiore della RAM BASIC al valore di default (40960/\$A000) dobbiamo far puntare la coppia di locazioni 55/56 (\$0037/\$0038) al byte 0 della pagina 160, infatti $(0+(160*256))=40960$:

```
POKE55,0:POKE56,160:CLR
```

È opportuno ricordare che la Memoria del C64, essendo un computer con un'architettura a 8 bit, può essere suddivisa in 256 "Pagine" di 256 byte ciascuna per un totale di 65536 byte (256*256) ossia 64K.

Vorrei inoltre ribadire ancora una volta che nei nostri esempi abbiamo lavorato rimanendo nell'ambito del Banco di Memoria N.0 che contempla i primi 16384 byte della RAM del C64: il VIC-II può vedere solo 16Kb alla volta (Bus Indirizzi a 14 bit) e quindi all'occorrenza gli si possono rendere visibili gli altri 3 banchi,

uno per volta, andando a scrivere in forma NEGATA il numero di uno dei 4 banchi (numerati da 0 a 3) nei bit 0 e 1 del Registro 56576/\$DD00 del CIA #2 - Port A.

In pratica al valore binario su 2 bit del Numero di Banco che si desidera selezionare (il default è il Banco numero 0) va applicato un NOT Booleano.

```
Esempio:
POKE 56576, (PEEK(56576) AND 252)OR1
```

Questa istruzione attiva il Banco 2 compreso tra gli indirizzi 32768 (\$8000) e 49151 (\$BFFF).

Infatti 1 decimale rappresenta 01 in binario che NEGATO (NOT logico) ci darà 10 corrispondente a 2 decimale ossia il numero di Banco (che può variare tra 0 e 3 per un totale di 4 Banchi di 16K ciascuno).

La COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE del C64 riporta che PRIMA di selezionare il Banco di Memoria, bisognerebbe impostare il Data Direction Register A del CIA #2, mappato all'indirizzo 56578 (\$DD02), settando a 1 i bit 0 e 1 in modo tale che i corrispondenti bit del Data Port Register A del CIA #2 (Registro 56576/\$DD00) siano settati per l'Output:

```
POKE 56578, PEEK(56578)OR3
```

Questa operazione NON È NECESSARIA in quanto il valore decimale di default del Registro 56578 (\$DD02) è 63 con i suddetti bit già settati a 1!

I CIA (Complex Interface Adapter) sono 2 integrati (i MOS 6526) che svolgono essenzialmente la funzione di Controller I/O interfacciandosi con la tastiera, con le porte di comunicazione (seriale e parallela), ecc.

Prima di concludere, vorrei fare un'osservazione riguardante l'allocazione della RAM riservata al programma BASIC: ho notato che in alcuni testi, come anche nella più volte citata COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE, viene riportato che, oltre ai puntatori 55/56 (\$0037/\$0038), bisogna modificare anche la coppia 51/52 (\$0033/\$0034) ossia i puntatori del limite della memoria riservata alle variabili stringhe. In realtà è sufficiente intervenire solamente sul contenuto delle locazioni 55/56 (\$0037/\$0038), come ho ampiamente spiegato più sopra, perché le variabili stringhe vengono memorizzate in maniera singolare ossia a partire dalla locazione puntata dalla

coppia 55/56 (\$0037/\$0038) procedendo verso la fine del programma BASIC (cioè verso il basso).

La cosa fondamentale è quella di far SEMPRE seguire la POKE dei Puntatori da uno statement CLR che oltre a resettare Variabili ed Array, di fatto rende effettiva la modifica dei suddetti Puntatori al limite della memoria riservata al Programma BASIC.

ASCII AND CHR\$ CODES

This appendix shows you what characters will appear if you PRINT CHR\$(X), for all possible values of X. It will also show the values obtained by typing PRINT ASC("x"), where x is any character you can type. This is useful in evaluating the character received in a GET statement converting upper/lower case, and printing character based commands (like switch to upper/lower case) that could not be enclosed in quotes.

PRINTS	CHRS	PRINTS	CHRS	PRINTS	CHRS	PRINTS	CHRS
	0	CHR(0)	17	"	34	3	51
	1	CHR(1)	18	#	35	4	52
	2	CHR(2)	19	\$	36	5	53
	3	CHR(3)	20	%	37	6	54
	4	CHR(4)	21	&	38	7	55
	5	CHR(5)	22	.	39	8	56
	6	CHR(6)	23	(40	9	57
	7	CHR(7)	24)	41	:	58
	8	CHR(8)	25	*	42	;	59
	9	CHR(9)	26	+	43	<	60
	10	CHR(10)	27	,	44	=	61
	11	CHR(11)	28	-	45	>	62
	12	CHR(12)	29	.	46	?	63
	13	CHR(13)	30	/	47	@	64
	14	CHR(14)	31	0	48	A	65
	15	CHR(15)	32	1	49	B	66
	16	CHR(16)	33	2	50	C	67

PRINTS	CHRS	PRINTS	CHRS	PRINTS	CHRS	PRINTS	CHRS
D	68	CHR(68)	97	TT	126	CHR(126)	155
E	69	CHR(69)	98	TT	127	CHR(127)	156
F	70	CHR(70)	99	TT	128	CHR(128)	157
G	71	CHR(71)	100	TT	129	CHR(129)	158
H	72	CHR(72)	101	TT	130	CHR(130)	159
I	73	CHR(73)	102	TT	131	CHR(131)	160
J	74	CHR(74)	103	TT	132	CHR(132)	161
K	75	CHR(75)	104	TT	133	CHR(133)	162
L	76	CHR(76)	105	TT	134	CHR(134)	163
M	77	CHR(77)	106	TT	135	CHR(135)	164
N	78	CHR(78)	107	TT	136	CHR(136)	165
O	79	CHR(79)	108	TT	137	CHR(137)	166
P	80	CHR(80)	109	TT	138	CHR(138)	167
Q	81	CHR(81)	110	TT	139	CHR(139)	168
R	82	CHR(82)	111	TT	140	CHR(140)	169
S	83	CHR(83)	112	TT	141	CHR(141)	170
T	84	CHR(84)	113	TT	142	CHR(142)	171
U	85	CHR(85)	114	TT	143	CHR(143)	172
V	86	CHR(86)	115	TT	144	CHR(144)	173
W	87	CHR(87)	116	TT	145	CHR(145)	174
X	88	CHR(88)	117	TT	146	CHR(146)	175
Y	89	CHR(89)	118	TT	147	CHR(147)	176
Z	90	CHR(90)	119	TT	148	CHR(148)	177
[91	CHR(91)	120	TT	149	CHR(149)	178
\	92	CHR(92)	121	TT	150	CHR(150)	179
]	93	CHR(93)	122	TT	151	CHR(151)	180
^	94	CHR(94)	123	TT	152	CHR(152)	181
_	95	CHR(95)	124	TT	153	CHR(153)	182
`	96	CHR(96)	125	TT	154	CHR(154)	183

PRINTS	CHRS	PRINTS	CHRS	PRINTS	CHRS	PRINTS	CHRS
CHR(184)	184	CHR(186)	186	CHR(188)	188	CHR(190)	190
CHR(185)	185	CHR(187)	187	CHR(189)	189	CHR(191)	191

CODES 192-223 SAME AS 96-127
CODES 224-254 SAME AS 160-190
CODE 255 SAME AS 126

RetroEvoluzione elettronica

di Michele Ugolini e Marco Pistorio

Natale 2018 sta arrivando. A che punto dell'evoluzione elettronica siamo giunti?

Mi sembra quasi prendere forma, dentro la mente, l'immagine della cupa lastra monolitica in Odissea nello Spazio, con le scimmie scalpitanti attorno. Perché questi ultimi anni sono popolati da così tanti retrogamer? Moda? Fanatismo? La risposta è complessa, le motivazioni sono molteplici, proviamo a scoprirne alcune. Come disse Neo in Matrix "... La scelta. Il problema è la scelta!"

Tutti noi potremmo scegliere di gettarci nelle mirabolanti grafiche degli ultimi titoli del gaming americano o nipponico, quindi perché stiamo discutendo di quanto sia sacro il Vectrex o accattivante Donkey Kong?

Da cosa nasce tanta passione per il Retrogaming? E' solo nostalgia legata alla nostra gioventù?

E' uno degli ingredienti probabilmente, ma forse quello preponderante è la genialità che ciascuna tipologia di videogiochi sapeva donare all'atto della nascita godendo di territori inesplorati e ricchi di peculiarità ideologica che oggidi verrebbe riassunta da più titoli di svariate saghe.

Ricordate Kevin Flynn ? Entrando in Tron , aveva davanti a sé l'io reale in un ambiente digitale, "roba forte!" esclamò lui stesso. La creatività aveva libero sfogo in un ambiente pressochè sterminato, dove ogni limitazione era confinata unicamente dalla quantità della propria fantasia (cit. di Ready Player One).

Pensate alla immensa genialità profusa in Doom in termini di mappe, gameplay, musica, tipologie di armi e nemici, trama etc.

Non che Call of Duty, Uncharted, Assassin's Creed siano titoli da disdegnare.

Tuttavia Doom non può non essere annoverato tra gli ingredienti del "brodo primordiale" della creazione digitale, che ci ha maturati in ciò che ora siamo, nel 2018!

Da questo "brodo primordiale" purtroppo, ogni tanto, perdiamo alcuni importanti Creativi che ci hanno donato tanto. Recentemente ci ha lasciati **Stan Lee**, il geniale creatore dell' universo Marvel. Grazie di cuore, Stan.

Dobbiamo ricordare anche **Ben Daglish**, che ha composto alcune tra le più raffinate melodie per numerosi 8bit.

Ben, non sarai ricordato ad egual modo di un compositore viennese dell'ottocento ma la tua arte vivrà in eterno nel regno digitale.

Scusate il poco spazio per salutare i grandissimi Stan Lee e Ben Daglish che ne meriterebbero certamente di più ma devo lasciarvi, sperando magari di essere riuscito a "spronarvi" a giocare intensamente alcuni titoli importanti del passato. Così facendo, partendo dalle origini, potrete comprendere meglio lo scenario attuale ed intuire cosa potrà riservarci il futuro!

Ringraziamenti

Chiudiamo, come nostro solito, con i **ringraziamenti** a tutti i **gruppi Facebook** ed ai siti **OldGamesItalia** ed **IlVideoGioco.com** che ci aiutano a condividere la rivista ad ogni uscita, con un ringraziamento particolare a **Vincenzo Scarpa** che sta riservando nel suo ottimo sito **EmuWiki** uno spazio dedicato a RetroMagazine e con un saluto a tutto lo staff di **8 Bit Retroprogramming Italia** con cui RM collabora attivamente per la pubblicazione di contenuti.

Alla prossima!

Disclaimer

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

RetroMagazine viene concesso con licenza: Attribuzione - Non commerciale - Condividi allo stesso modo 3.0 Italia (CC BY-NC-SA 3.0 IT):

<https://creativecommons.org/licenses/by-nc-sa/3.0/it/>

In pratica sei libero di:

Condividere - riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato.

Modificare - remixare, trasformare il materiale e basarti su di esso per le tue opere.

Alle seguenti condizioni:

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

NonCommerciale - Non puoi utilizzare il materiale per scopi commerciali.

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

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

RetroMagazine

Anno 2 - Numero 11

Direttore Responsabile
Francesco Fiorentini

Vice Direttore
Marco Pistorio

Dicembre 2018