

LUGLIO/AGOSTO 2018 - WWW.RETROMAGAZINE.NET

Summer on a Retrogaming Beach

di Starfox Mulder

Ah l'Estate, croce e delizia. Per alcuni momento di svago, partitelle a beach volley in riva al mare con le ragazze, familiare di Peroni ghiacciata mentre guarda le partite dei mondiali e vacanze rilassanti. Per altri invece la calda stagione ha sempre significato un bagno di sudore, strade congestionate dal traffico dei vacanzieri e nessun reale interesse per mostrare il proprio fisico scolpito a suon di picchiaduro Capcom ed abbronzato grazie ai CRT radio-attivi degli anni 80. Noi di RetroMagazine scriviamo per tutti, sia per chi non mette il naso fuori di casa che per i manzi da ombrellone; in fondo la settimana enigmistica stanca e vorrete sapere come affrontare il primo livello di Legend of Grimrock quando rientrerete dalle ferie no?

Nonostante la nostra propensione nazionale popolare però, anche in redazione si sente l'odore di vacanze e tra chi si appresta ad andare in Giappone per futuri reportage e chi si accontenta di un escursione peschereccia ad Innsbruck, l'intera redazione ha gridato FERIE come un telecronista brasiliano urlerebbe GOAL.

Non potevamo però lasciarvi senza di noi per un'intera stagione, ecco quindi che il numero estivo si è trasformato in uno speciale doppio, carico di articoli, recensioni, retrospettive ed interviste, come poche altre volte ci era capitato di farcirlo. Non mancheranno certo gli articoli tecnici a cui vi abbiamo ormai

abituati, così come vedrete nascere nuove rubriche quali RetroGiochiAmo o Exploring Amiga, ma di certo un occhio di riguardo l'abbiamo messo nella recensione dei giochi, in onore della stagione più "leggera" dell'anno.

Per me che son cresciuto al mare la cosa non ha mai fatto tanta differenza ma per molti montanari, l'occasione buona per godere delle tanto desiderate sale giochi era rappresentato proprio dai mesi estivi in cui ci si perdeva tra splendidi cabinati in fumose sale piene di tamarri di qualche lido costiero. Oggi le sale giochi sono praticamente scomparse, lasciando posto a sale slot e rivenditori di sigarette elettroniche, ma l'idea che l'Estate sia la stagione dei videogames vale per me come per tutti i nati prima dei goes, come un dogma radicato nel nostro D.N.A.

Analogamente al cane di Pavlov che salivava al suono del campanello, la mia mente rilascia endorfine ad ogni musicchetta in 8 bit udita distrattamente, quindi per star sul sicuro mi sono fatto una bella compilation da ascoltare sotto l'ombrellone, magari leggendo anch'io il numero che state tenendo ora tra le mani. Che dire? Vi consiglio di fare altrettanto, e se non sapete come sfogare la voglia di videogaming che inevitabilmente vi assalirà, tranquilli: i consigli ve li diamo noi!

RetroMagazine

Anno 2 - Numero 8

- La trinità' femminile dell'Informatica
- D.I.Y. RETROGAMING CLUB
- RetroGiochiAmo: Grimrock - The series
- Navigare nelle BBS con il C64 su VICE
- RetroGiochiAmo: Minesweeper
- Aleph-project di Isacco Luongo
- Retromath: a spasso tra i dati mancanti
- Esplorando l'Amiga - parte 1
- Demo in Basic per C64 - L'alieno!
- JoyTest64
- Console 8bit: GAMEBOY
- ABC della merenda... TI99/4A - parte 3
- Le avventure testuali: da Infocom a Inform

GIOCHI

- Tetris (Gameboy)
- Green Beret (Arcade)
- Salvate lo Stregatto (Multiplatforma)
- Super Burnout (Atari Jaguar)
- Druid (Commodore 64)
- Captain Blood (Olivetti Prodest PC 128)
- Wimbledon Championship Tennis (MD)
- Sid Meier's Pirates! (Amiga)

Intervista a Gideon Zweijtzter
Sito Web: CPCrueZ

IN EVIDENZA IN QUESTO NUMERO



Aleph Project di Isacco Luongo

Il microprocessore Zilog Z80 può indirizzare solo 65.536 bytes di memoria complessiva. Ma Isacco Luongo è convinto che possa fare di meglio!

Articolo a pagina 12



Intervista a Gideon Zweijtzter

Il designer della cartuccia 1541 Ultimate e della scheda Ultimate64 ha concesso al nostro David La Monaca un'intervista esclusiva! Abbiamo ripercorso con Gideon le tappe salienti della sua avventura di progettista di moderne periferiche per C64.

Articolo a pagina 44

La trinità' femminile dell'Informatica

di Alberto Apostolo

In questo articolo si illustrano brevemente le vite di tre donne che sono state pioniere dell'Informatica: Ada Byron (nota anche come Ada Lovelace), Grace Hopper e Jean Sammet.

Ada Byron è nota per essere ritenuta la prima persona in assoluto a scrivere un programma per una macchina calcolatrice.

Grace Hopper è stata il primo militare di sesso femminile a raggiungere il grado di Ammiraglio della Marina degli Stati Uniti d'America e ha scritto il primo compilatore per computer (che traduceva un linguaggio simbolico in istruzioni per un computer).

Jean Sammet ha lavorato al comitato che ha inventato il linguaggio COBOL ed è stata considerata una autorità nel campo dei linguaggi di programmazione.

Queste donne sono nate e cresciute in epoche nelle quali una donna doveva lottare per infrangere gli stereotipi imposti dalle convenzioni sociali e coronare le proprie aspirazioni intellettuali e professionali (in più avendo la responsabilità di essere mogli e madri, come per esempio Ada Byron). Certamente non sono le uniche figure femminili che hanno contribuito al progresso dell'Informatica e ispirato le nuove generazioni a intraprendere lo studio delle materie scientifiche: per tale motivo si invitano i lettori ad un approfondimento che parte da alcuni dei testi elencati nella Bibliografia ([Ind14],[ORE13],[SP15]).

Ada Byron, la prima programmatrice della Storia

Ada Byron nacque il 10 Dicembre 1815 a Londra, dal poeta Lord George Gordon Byron e da Anne Isabella Milbanke (una studiosa di matematica).

Non ebbe quasi mai rapporti con il padre e fu la madre a educarla alle Scienze e al Calcolo.

Fu introdotta a Corte nel 1833 e presentata nello stesso anno a Charles Babbage durante una cena. Ada e sua madre ebbero occasione di visitare lo studio di Babbage a Londra dove un prototipo della Macchina Differenziale era

in mostra. La macchina affascino' Ada che ne riconobbe la bellezza della invenzione.



Figura 1: Ada Byron (immagine da Google).

La leggenda narra che Ada Byron e Charles Babbage si innamorarono, anche se non si trovano riscontri nelle cronache dell'epoca. Nel 1834, Ada Byron ebbe modo di stringere amicizia con l'astronoma (e matematica) inglese Mary Somerville, la quale guidò Ada negli studi di matematica, inviandole libri e problemi da risolvere. Esse discussero regolarmente di Matematica e Scienze, includendo discussioni sulle macchine calcolatrici di Babbage.

Ada Byron era affascinata dall'idea della Macchina Analitica con le sue applicazioni, ed ebbe una corrispondenza regolare con Babbage dove poteva comunicargli le sue personali idee riguardo tale macchina. Infatti aveva intravisto, per le macchine calcolatrici, la possibilità di andare al di là del semplice calcolo numerico, quando lo stesso Babbage e i suoi contemporanei si focalizzavano solo su

questo aspetto. Credeva che la Macchina Analitica potesse manipolare dei simboli secondo regole prefissate. In tal modo un numero poteva rappresentare una entità oltre che una quantità (e dunque essere utilizzata non solo in ambito matematico).

Nel 1835 sposò William King (che divenne il Conte di Lovelace nel 1838). Ebbe con lui 3 figli: Byron, Annabella, Ralph Gordon. Nonostante i doveri familiari, nel 1841 iniziò gli studi di matematica avanzata con il matematico inglese Augustus De Morgan. Afflitta da malattie per tutta la vita, morì il 27 Novembre 1852 all'età di 37 anni per un cancro all'utero.

La sua produzione più importante è una traduzione commentata del saggio (in francese) dell'italiano Luigi Menabrea "Notions sur la machine analytique" di Charles Babbage. Nei commenti, Ada Byron spiega come si poteva programmarla e fornì un esempio di un algoritmo per generare i numeri di Bernoulli, considerato come il primo programma per macchine calcolatrici della Storia (Figura 2). Pertanto Ada Byron, si può ritenere la prima programmatrice in assoluto della Storia.

Let us now suppose that we have *two* expressions whose values have been computed by the engine independently of each other (each having its own group of columns for data and results). Let them be ax^n , $b \cdot p \cdot y$. They would then stand as follows on the columns:—

V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
+	+	+	+	+	+	+	+	+
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
a	n	x	ax^n	b	p	y	bpy	$\frac{ax^n}{bpy}$

Figura 2: Frammento del primo programma di Ada Byron, pubblicato nelle "Taylor's Scientific Memoirs" del 1840 [LM84].

Nel 1979, il Dipartimento della Difesa degli Stati Uniti ha onorato il ricordo di Ada Lovelace, battezzando ADA un innovativo linguaggio di programmazione per grandi sistemi di calcolo. Curiosamente fino a quel

momento il suo nome non era nemmeno citato nelle enciclopedie.

La British Computer Society ha istituito la Medaglia Lovelace per premiare chi si distingue nel progresso dell'Informatica. Il vincitore e' invitato l'anno successivo a dare una pubblica dissertazione.

Nell'annuale Giornata di Ada Lovelace si ricordano le donne che hanno dato il loro contributo alla Matematica, alle Scienze, all'Ingegneria.

Grace Hopper, l'Ammiraglio del Cyberspazio

Grace Murray (in seguito al matrimonio divenuta Grace Hopper) era nata a New York il 9 Dicembre 1906. Maggiore di tre fratelli, fin da bambina fece apparire i segni di un interesse per l'Ingegneria. Gia' all'eta di 7 anni cercava di capire come funzionassero certi meccanismi come gli allarmi a tempo.



Figura 3: Grace Hopper al Vassar College (immagine da Google).

Crescendo, dimostro' di non temere di infrangere gli stereotipi femminili. A scuola praticava pallacanestro, hockey su prato e pallanuoto. A 17 anni entro' al Vassar College per studiare Matematica e Fisica. Probabilmente i suoi interessi derivavano dal fatto che sua madre si interessava di Matematica mentre suo padre era un ingegnere. Dopo la laurea al Vassar, conseguì la laurea specialistica in Matematica e il Dottorato alla Universita' di Yale. Fu la prima

donna a conseguire il Dottorato in Matematica a Yale.

L'insegnamento continuo' a essere la sua passione fino allo scoppio della II Guerra Mondiale. Compreso che arruolarsi nelle forze armate americane poteva essere una opportunita' di affermazione professionale. Malgrado il fisico esile, fu finalmente accettata dalla Marina degli Stati Uniti nel 1943 ma non assegnata in prima linea. Venne assegnata al Bureau of Ordinance Computation Project dell'Universita' di Harvard, dedicandosi alla decrittazione dei codici segreti che permisero agli Alleati di vincere la guerra. Lavorava a stretto contatto solo con gli uomini. I computer allora erano enormi macchine e funzionavano con le schede perforate e le bobine magnetiche. Era sola in un mondo dove le donne stavano fuori dalla porta a battere a macchina. Fumava. La sua bellezza si notava dopo un po', nascosta dietro la divisa bianca della Marina. Non amava apparire. Ma non dimenticava mai di mettersi lo smalto per le unghie. Il calcolatore sul quale lavorava, il Mark I, pesava piu' di 5000 Kg e poteva eseguire solo le 4 operazioni. Queste limitazioni la ispirarono a creare qualcosa di meglio.

Nel 1949 era una poche donne esperte in un settore che ancora non fruttava denaro. Fu assunta alla Eckert-Mauchly Computer Corporation e nel 1952 termino' la realizzazione del primo compilatore, ossia un traduttore che leggeva simboli comprensibili da un essere umano e li trasformava in istruzioni comprensibili dal computer.

Il compilatore rese l'attivita' della programmazione piu' facile e permise di far eseguire ai computer molte piu' cose dei semplici calcoli. All'epoca molti scienziati e tecnici stentaron a credere che fosse stato possibile creare qualcosa del genere finche' non lo videro con i loro occhi. L'invenzione del compilatore le procuro' molto rispetto e fu promossa alla direzione della programmazione automatica della societa' per la quale lavorava.

Inoltre diede contributi alla creazione del linguaggio FLOW-MATIC e del linguaggio COBOL (del quale il FLOW-MATIC e' un precursore). Nel frattempo proseguiva la sua carriera militare nella Riserva della Marina

fino al congedo nel 1986 con il grado di Ammiraglio di Divisione. Mori' il Primo Gennaio 1992 ed e' sepolta al cimitero militare di Arlington. Non ebbe figli e divorzio' mantenendo il cognome del marito, il professor Vincent Foster Hopper.

Grace Hopper fu uno dei principali programmatori del computer Mark I. In questa macchina trovo' il primo bug per computer, ossia una falena morta che era finita nel Mark I e le cui ali bloccavano la lettura di un nastro perforato.

La parola bug era gia' stata usata in precedenza nel 1889 per descrivere un difetto ma Grace Hopper e accreditata di avere coniato la parola debugging per descrivere il lavoro di eliminazione degli errori da un programma [Dix10].

Nel suo ufficio a Washington, aveva appeso un orologio che andava al contrario: le lancette procedevano in senso anti-orario cosi' come era invertita la posizione dei numeri. -Come fa a leggere l'ora?-, le chiedevano tutti. -Non si vede perche' debba guardare la faccenda da un punto di vista che hanno deciso gli altri! -, era la risposta.

L'Anita Borg Institute tiene ogni anno una conferenza intitolata Grace Hopper Celebration of Women in Computing per ricordare il contributo delle donne nell'Informatica.

Jean Sammet, la creatrice del COBOL

Jean Elaine Sammet divenne una delle prime donne esperte nell'area dell'Informatica relativa ai linguaggi di programmazione. Senza il suo duro lavoro, la programmazione dei computer non sarebbe quello che e' oggi.

Nacque a New York il 23 Marzo 1928. L'interesse per la Matematica si manifesto' in giovane eta'. Da ragazzina partecipo' a tutte le lezioni di matematica che poteva.

Nel 1948 conseguì la laurea di primo livello in Matematica al Mount Holyoke College. Appena un anno piu' tardi, conseguì la laurea specialistica in Matematica presso l'Universita' dell'Illinois. Non conseguì mai un Dottorato (anche se ne ricevette uno honoris causa trent'anni piu' tardi). Nel 1955 inizio' a lavorare alla Sperry Gyroscope, una azienda che realizzava congegni elettronici e altre

apparecchiature. Questo fu il suo lavoro nel campo dei computer. Come esperta di matematica, supervisionava un gruppo di programmatori scientifici.



Figura 4: Jean Elaine Sammet (immagine da Google).

Tre anni più tardi si fece assumere alla Sylvania Electric Products. Qui partecipò a molti progetti, compreso il software per un primo computer chiamato MOBIDIC (MOBILE Digital Computer) che era usato dall'Esercito Americano. All'epoca i computer erano molto grandi e questo necessitava di un veicolo con rimorchio per poter essere trasportato.

Nello stesso periodo, Jean Sammet stava lavorando con il CODASYL Short Range Committee (Conference on DATA SYstem Language). Tale gruppo era impegnato per creare un linguaggio standard per computer che potesse essere usato in ambito commerciale, per impedire che ogni azienda fosse costretta a inventarne uno da sola.

Il linguaggio COBOL (COMmon Business Oriented Language) fu il risultato del lavoro di questo comitato. Oggi è uno dei linguaggi di programmazione più vecchi esistenti.

Nel 1961, Jean Sammet divenne un manager della IBM al Boston Programming Center. Mentre si trovava alla IBM partecipò allo sviluppo del FORMAC (FORMula MANipulation Compiler), un linguaggio per l'elaborazione simbolica delle formule matematiche.

Nel 1969 scrisse un importante testo intitolato "Programming Languages: History and Fundamentals", che divenne molto di

uso e che la rese ancora di più conosciuta come una esperta del suo settore. Inoltre trascorse qualche tempo come insegnante universitaria presso l'Adelphi University.

Appartenente alla ACM (Association for Computing Machinery), nel 1974 fu la prima donna a ricevere l'incarico di Presidente dell'associazione. Nel 1976 fu accettata come membro della National Academy of Engineering. Nel 1978, il Mount Holyoke College le assegnò un Dottorato honoris causa. Nel 1979 contribuì allo sviluppo di un altro linguaggio per computer, l'ADA (chiamato così in onore di Ada Lovelace). Nel 1985 ricevette l'ACM Distinguished Service Award e nel 1989 fu premiata con l'Augusta Ada Lovelace Award dalla Association for Women in Computing. Nel 2001 ebbe i più alti onori per essere nominata Computer History Museum Fellow.

Jean Sammet morì il 20 Maggio 2017, dopo una breve malattia.

Alla Prima Conferenza sulla Storia dei Linguaggi di Programmazione avvenuta a Los Angeles nel 1978, vi fu l'ardente intervento di Jean Sammet per difendere il linguaggio COBOL dall'accusa di essere un linguaggio antiquato. ... e il COBOL ha i più completi meccanismi di Input/Output di qualsiasi altro linguaggio di programmazione esistente oggi. Avete mai notato che la maggior parte dei linguaggi di oggi non includono l'I/O nella loro definizione? Linguaggi come l'ALGOL oppure l'ADA oppure il C? Li delegano a qualche genere di library package separato!-Poi guardo' diritto al pubblico. -E sapete perché non includono l'I/O? Perché è difficile!- I partecipanti cominciarono a ridere, ma le risate finirono velocemente man mano che si rendevano conto che essenzialmente aveva ragione. Ancora oggi i meccanismi per l'I/O formattato forniti dalla maggior parte dei linguaggi di programmazione sono poco più di una riformulazione di quelli già forniti dal FORTRAN decenni fa. Ma la comunità COBOL, disperatamente bisognosa di buoni meccanismi I/O per i moduli e i documenti complessi e strutturati caratteristici della programmazione commerciale, si era semplicemente rimboccata le maniche e aveva fatto il difficile, faticoso lavoro necessario per rendere quei meccanismi disponibili [Fav03].

Bibliografia

- [Dix10] J.B. Dixit, Fundamentals of Computers, Laxmi Publications Ltd, 2010.
- [Fav03] J. Favaro, Perché è difficile, Computer Programming, n.128, Ottobre 2003.
- [Ind14] S. Indovino, Women in Engineering, Simon and Schuster, 2014.
- [LM84] G. Lolli, C. Mangione, Matematica e Calcolatore, Le Scienze Quaderni, n.14: pp. 3-9, Marzo 1984.
- [ORE13] G. O'Regan, Giants of computing: A compendium of Select, Pivotal Pioneers, Springer Science & Business Media, 2013.
- [SP15] M. Serani, M. Pennisi, Donne che amano la tecnologia. Da Ada Lovelace a Lisa Simpson, Corriere della Sera, 2015.
- [Wik17] Wikipedia, ultima consultazione: 10 agosto 2017.

```
IDENTIFICATION DIVISION.  
    PROGRAM-ID. HELLO-WORLD.  
*  
ENVIRONMENT DIVISION.  
*  
DATA DIVISION.  
*  
PROCEDURE DIVISION.  
    PARA-1.  
        DISPLAY "Hello, world."  
*  
        EXIT PROGRAM.  
    END PROGRAM HELLO-WORLD.
```

Figura 5: Esempio di 'Hello World' in COBOL.



Figura 6: Macchina analitica di Charles Babbage - Museo Nazionale della Scienza e della Tecnologia Leonardo da Vinci di Milano.

D.I.Y. RETROGAMING CLUB

...ovvero come fondare un proprio club e vivere felici.

di Starfox Mulder

Salve a tutti, sono Starfox Mulder e scrivo su RetroMagazine dal numero 2, ma solo perché il primo numero Francesco se l'è fatto tutto da solo, e gestisco un club di retrogaming. In questo articolo vi parlerò di come ho fatto a tirarlo su e tenerlo in piedi ancora oggi. Do It Yourself (l'acronimo D.I.Y. del titolo) quindi, ed è esattamente la stessa cosa che feci con i Bit-elloni, mio team di nascita, due anni fa. Prima di cominciare a parlarvene però ci tengo a precisare alcuni punti fondamentali:

1- Non voglio insegnare nulla a nessuno. Non mi piace chi si mette in cattedra a fare l'uomo vissuto, per questo quanto vi racconterò in seguito spera di essere un "se ce l'ho fatta io potete farcela tutti" e non un libretto d'istruzione sul come riuscire nell'impresa.

2- Prendete questo articolo come un racconto di una missione andata a buon fine, salvo il caso in cui siate dotati di: buona volontà, un paio di amici motivati quanto voi ed una passione smodata per le serate in compagnia. Se vi mancano questi tre elementi, o anche uno solo dei tre, difficilmente riuscirete nella cosa.



Dopo queste due premesse vien da sé domandarsi: "ok, ma di che missione stai parlando?"

Creare un punto di ritrovo periodico per appassionati di retrogaming, in cui giocare

assieme, sfidarsi in appositi tornei e conoscere nuove persone con cui condividere il proprio hobby.



Molti di noi sono legati al retrogaming o al retrocomputing per nostalgia, ricordandosi le esperienze giovanili, ma altri si sono avvicinati a questo mondo solo di recente, con il chiaro intento di scoprire quanto di bello si fossero lasciati indietro, partendo proprio dal lato ludico della cosa. Chi si limita alla ricerca individuale, magari condividendo la propria esperienza solo sui forum o nei social, probabilmente non troverà attrattiva nello sfidare persone mai viste, in match di multiplayering offline (trovo quasi fastidioso dover sottolineare offline ma dati i tempi è necessario), ma i Bitelloni nacquero a scopo aggregativo e questo obiettivo non poteva esaurirsi in un blog su cui ci dilettevamo a scrivere recensioni sopra le righe. Ci serviva un luogo, un edificio reale, una sede in cui invitare chiunque a portare le proprie console e condividerle con gli altri. Da dove partire però?

Prima ho elencato tre elementi necessari per gettarsi nell'impresa ed ho tenuto fuori quello

che tutti si aspetterebbero essere invece il primo dei requisiti: i soldi.

Beh gente, vi stupirete nello scoprire che non servono affatto, non come li intendete voi.

Certo, qualche spesa capiterà di farla ma se saprete ragionare in ottica associativa, alla fine dell'anno sarete sempre in pari.

Come dicevo, non c'è una regola su come creare un gruppo di retrogaming ma posso raccontarvi "come lo abbiamo fatto" e quindi tanto vale procedere per step:

STEP 1 – IL TEAM

Quegli "amici motivati quanto voi" saranno il motore degli eventi ed a meno che non siate dotati di un ego pari a quello di certi tristemente noti leader storici sappiate che gli amici vi servono perché le difficoltà ci saranno ed arriverà ben presto il momento che senza una pacca sulle spalle ed un "stavolta ci penso io, tu riposati" non si andrà avanti. Noi siamo in tre, me compreso, ma penso che già in due si possa fare, purché non si parta all'avventura in solitaria.



Il Team ci deve essere perché ogni serata sarà un piccolo evento e richiederà preparativi e presenza in loco, cose che una volta rodiate vi comporteranno uno sforzo piuttosto leggero ma prima di allora necessiteranno pazienza e dedizione. Capita che qualcuno si ammali, che

un altro per esigenze di lavoro o famiglia non possa venire, e se non si è in numero maggiore di 1 quell'imprevisto comporterebbe il far saltare l'intera serata: non proprio una bella immagine per un club che deve iniziare a farsi conoscere.

Una volta creato il gruppo che si occuperà di gestire gli eventi ci vuole ovviamente...



STEP 2 – UNA SEDE

Il luogo fisico in cui tenere gli eventi. Le possibilità sono infinite ma la più economica è quella gratuita. Ogni comune ha degli edifici che lascia in concessione ad associazioni no profit che svolgono attività sul territorio e guarda caso voi farete proprio quello, quindi perché non usufruirne?

Lo so cosa vi sta passando per la testa: "quindi dovremmo fondare un'associazione?" Assolutamente No, vade Retro (l'uso di questo termine non è puramente casuale).

La normativa in merito alle associazioni è complessa ed i controlli si sono molti inaspriti negli ultimi anni per far fronte a tanti furbetti che sfruttavano la cosa per creare aziende camuffate da associazioni così da pagare meno tasse. La verità è che in Italia ci sono all'attivo un numero incalcolabile di associazioni pre-esistenti e la maggior parte di esse, anche nella vostra città, stanno avendo dei problemi. Magari perché il comune si è stancato di concedere spazi a gente che registra 10 iscritti a stagione. Magari perché i membri del consiglio direttivo sono già anni che fanno altro e l'associazione esiste solo perché nessuno si è preso la briga

di chiuderla ufficialmente. I magari sono tanti ma il succo è che si può tentare di offrire una collaborazione ad una di queste associazioni in difficoltà ed ottenere un vantaggio reciproco. Ad esempio, un argomento che fa abbastanza presa è: <<Ciao, siamo i taldeitali. Stiamo cercando un'associazione con cui collaborare per organizzare delle serate ludiche in cui coinvolgere i soci e soprattutto attirarne tanti altri nuovi. Ci servirebbe un posto in cui organizzare i nostri eventi e magari una zona in cui appoggiare un paio di armadi in cui terremo il materiale occorrente. In cambio ci possiamo accordare per dividere i proventi delle serate e dei tesseramenti legati ai nostri eventi. Vi interessa essere quell'associazione?>>.

In soldoni è come andare da un concessionario a proporgli di vendere auto per lui senza venir per questo stipendiati ma raccogliendo una piccola percentuale sugli utili. Se il concessionario dovesse viaggiare a vele spiegate si limiterà a declinare l'offerta mentre se le cose non gli stanno andando così bene vi vedrà come dei possibili salvatori e vi garantisco che se saprete muovervi con la buona volontà di cui si parlava sopra lo sarete.



STEP 3 – IL MATERIALE

Il Club deve possedere dei beni suoi. Televisori a tubo catodico, console, home-pc, ciabatte elettriche e quant'altro. Partendo da un budget di 0 euro va da se che queste cose inizialmente non ci saranno e dovrete metterle di mano vostra. Ad apertura club noi abbiamo organizzato serate utilizzando quasi esclusivamente le mie console e le TV che avevo rimediato da alberghi che le gettavano nei cassonetti perché obsolete. Questa è una condizione di partenza necessaria ma di relativamente breve durata poiché da che siamo partiti ad inizio 2017 senza nulla, oggi, ad un anno e mezzo di distanza, abbiamo 6

televisori, un raspberry pi 3, 6 console, un cabinato di Point Blank e due home-pc, tutti proprietà del club. Qualcosa è stato donato da noi stessi, o dai soci, ma la maggior parte delle cose sono state acquistate coi proventi delle serate. Difficile fare i conti precisi ma l'ultimo investimento fatto ammontava a 300€ di spesa e vi parlo di soldi che avevamo in cassa. Come questo sia possibile ve lo spiego nello...



STEP 4 – AUTO-FINANZIAMENTO

Entrare in un club dovrebbe essere permesso a tutti, ma tutti dovrebbero fare la propria parte. Se spesso risulterà impossibile chiedere un contributo attivo per organizzare gli eventi, altrettanto scontato deve essere il tesseramento. Ognuno si organizza come vuole, spesso in accordo con l'associazione a cui ci si appoggia, ma solitamente 10€ all'anno sono un buon compromesso. Da lì in poi noi abbiamo scelto di organizzare serate gratuite con tesseramento obbligatorio sin da subito ma da pagarsi dalla seconda serata a cui si parteciperà, il tutto al fine di incoraggiare quanta più gente possibile a "provare" un nostro evento, senza per questo obbligarli a un contributo immediato. Dalla seconda serata invece è chiaro che quel che facciamo sia piaciuto e per tanto si entra di diritto a far parte del club. Oltre al tesseramento ci sono i tornei a premio, che per tanto richiedono una quota partecipativa. 2-3€ a torneo, a seconda del premio messo in mostra. Se si prevedono 16 iscritti, con 3€ a partecipante si incassa quasi un cinquantone ed ecco che prendendo un premio da una trentina di euro (anche una console loose per far gola) avanza comunque una quota da tenere per il club. In ultimo il bar. Se ne avete la possibilità sarà sicuramente cosa gradita a tutti poter usufruire di bevande



fresche e qualche merenda per ricaricare le forze dopo duri match passati a menarsi in Street Fighter 2. Niente prezzi da discoteca, mi raccomando, basta quel minimo surplus per ricavarne qualcosa dopo aver ripagato le spese del supermarket.

STEP 5 – COINVOLGERE IL PUBBLICO

Il Club lo fanno gli associati e la gente vuole divertirsi. Concetto semplice ma tutt'altro che scontato. Ho partecipato a tante interessanti serate gestite da persone ben più competenti di me ma spesso erano rivolte ad un pubblico di appassionati già ben introdotti nell'ambiente. Questo è male!

Ci saranno sempre eventi specifici per "hobbyisti" ma quel che serve per attirare nuovi utenti che poi in seguito diverranno naturalmente appassionati è saperli coinvolgere ed il modo migliore è farli giocare. Che durante la serata si possa accedere ad un Amiga 600 o ad un 3DO quel che conta è farci girare dei videogames di indubbia qualità e spingere gli avventori a parteciparvi. Che sia con un torneo a premi o con una competizione del tipo "il migliore vince una bevuta offerta dal club" quel che conta è che giochino, si divertano, capiscano il valore di quel che stanno apprezzando e finiscano (un domani) per interessarsene. Siamo partiti giocando e siamo arrivati a serate culturali con retrospettive su console e case di produzione. Abbiamo invitato ospiti per eventi speciali al fine di illustrare console sfortunate

commercialmente e spesso proponiamo contest alla "ottieni il miglior record della serata" su titoli arcade dei primi anni 80. Si parte da quello che tutti possono capire, ossia il divertimento, per poi arrivare a quello che interessa più agli appassionati, mai il contrario.

CONCLUSIONE

L'inizio ci ha richiesto fatica e molto impegno. Contattare gli amici uno ad uno, dir loro di portare altre persone, fare volantinaggio nelle scuole, condividere a più non posso gli eventi su facebook e molto altro ancora. Una faticaccia, non lo nego, ma ad ogni successiva

serata la cosa diventava più facile. Se prima portavo di mio 4 console ora ne porto (quando capita) una se non addirittura nessuna. Se prima si impiegava tanto tempo a fare public relationship ora sono gli utenti stessi a scriverci per sapere "a quando la prossima serata". Nel giro di appena un anno eravamo già autonomi come club ed apprezzati da chi ci ospitava così come da chi veniva alle serate che organizzavamo per capire cosa fosse questo retrogaming. Ora stiamo dicendo di no a richieste di collaborazione perché ognuno di noi ha una vita, una famiglia e degli impegni da portare avanti oltre al club ma anche perché è estate e ci siamo presi qualche mese di ferie. Si riaprirà ad Ottobre e speriamo sempre di incontrare nuovi appassionati pronti a venire in visita così come trasformare inesperti giovani in futuri retrogamers d'assalto. Durante la breve vita del club abbiamo già visto crearsi amicizie, amori, passioni ed anche qualche sana diatriba. Un successo insomma ma non sono qui per lodarci quanto semmai per provare ad ispirare altri sognatori. Avete mai sognato un club tutto vostro in cui condividere la vostra passione? Guardate stralunati un punto fisso davanti a voi e gridate insieme a me: SI PUÒ FAREEEEE!

Nota: I Bit-elloni si trovano a Gradara (PU), in via del cimitero 1. Le serate sono aperte a tutti e si tengono da Ottobre a Maggio di ogni anno, due volte al mese. Per restare aggiornati sugli eventi iscrivetevi alla [pagina facebook de "I Bit-elloni"](#).

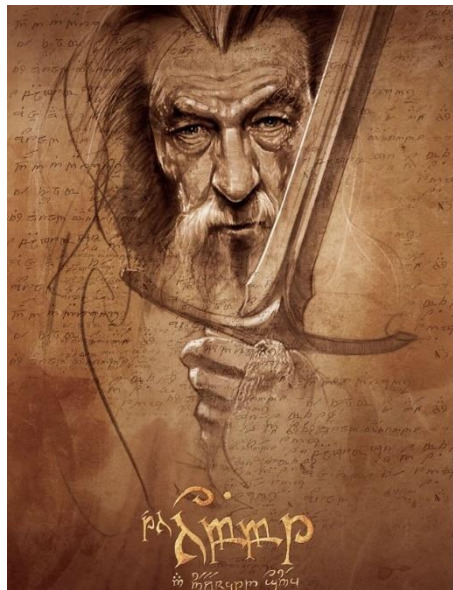


RetroGiochiAmo: Grimrock - The Series

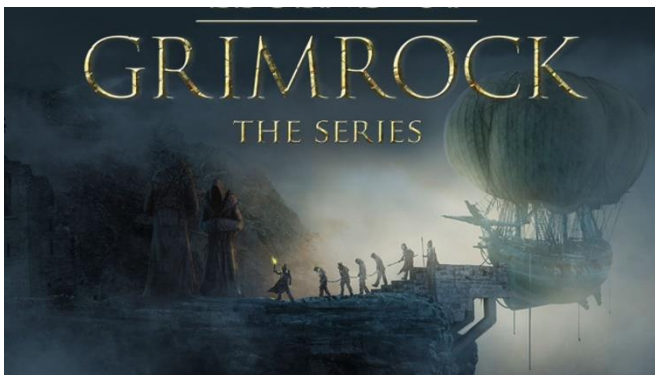
di Ermanno Betori

Dungeon, Dungeon delle mie brame, chi è l'eroe più tosto del reame?

Quanti giovani degli anni 70-80 si sono immedesimati nei giochi RPG? Tanti, ma nella specifica realtà Italiana dell'epoca, questo genere di gioco non venne subito apprezzato, in quanto spesso i giochi prefiggevano un background culturale che era spesso inesistente. Basti pensare che questo tipo di giochi da tavolo denominati "Giochi di Ruolo alias Role-Play-Game", ove il giocatore si doveva immedesimare in un personaggio (l'eroe) che viveva in un mondo simulato, nacquero nel 1975 e molta dell'ambientazione per creare il mondo virtuale fu ripresa all'epoca dai libri di Tolkien come l'Hobbit, Il Signore degli Anelli...



Oppure dalle epopee delle saghe del nord Europa o dalla mitologia Celtica. Genere storico - culturale che in Italia era quasi inesistente. Furono molto popolari in Europa fino quasi alla metà degli anni 90 dopodiché ebbero un declino e si ritenevano estinti. Oggi inauguriamo come da titolo una rubrica dedicata ai videogiochi RPG. Nello specifico data l'enormità di argomenti, cominciamo parlando nello specifico dei RPG Crawl. Questa definizione definisce un gioco di ruolo ambientato in un ambiente fantastico, dove è presente un labirinto nel quale è sempre intrappolato l'eroe che deve combattere



mostri, trovare tesori ecc.. Titoli famosi in grafica bidimensionale furono ad esempio Wizardry, The Bard's Tale, Cosmic Soldier, Might and Magic. Nella nostra trattazione invece presenteremo un sotto genere famosissimo che ebbe gloria con i computer 16bit tipo Amiga, Atari ST, Dos 286 e superiori. Parliamo dei RPG Crawl 3-D, che il capostipite fu il gioco Dungeon Master.

Titolo che era un richiamo potente per tutti gli appassionati di questo tipo di giochi. Letteralmente Dungeon Master significa Signore dei Sotterranei, e nel gioco da tavolo viene identificato nell'arbitro o narratore che decide i destini dei vari eroi e detta le regole del mondo. Perciò il Dungeon Master è IL gioco. Se il narratore è abile il gioco è stupendo, altrimenti pessimo. Si pensi che in Giappone una serie di libri, cartoni animati e videogiochi chiamati Record of Lodoss War è un universo fantasy ideato dallo scrittore giapponese Ryo Mizuno e nasce come racconto ispirato ai diari di gioco di una campagna a Dungeons & Dragons di cui Mizuno era il dungeon master.

Come videogiochi Crawl ne furono creati svariati, noi partiremo nella loro scoperta cominciando dall'ultimo creato, cioè dal leggendario GRIMROCK.....

Legend of Grimrock è un videogioco di ruolo sviluppato e pubblicato dalla software house finlandese indie Almost Human nel 2012 e il suo seguito nel 2014.

Cosa dire del gioco rispetto ai suoi illustri predecessori?

Fantastico!
Stupendo!
Meraviglioso!



I programmatori della Almost Human, complice l'inverno finlandese che li ha fatti lavorare in una ambientazione adeguata, vedi freddo polare, notte eterna e forse qualche licantropo di passaggio, hanno esattamente recepito lo spirito dei vecchi giochi rieditando la grafica e creando una serie di trappole / indovinelli che sono state distribuite in modo abbastanza adeguato, così si parla in politic correct mode.. in modalità vecchio giocatore attempato, la definizione cambia in Vigliacchi! Infami! Carogne! Fate credere che a 50 anni suonati uno è ancora un bimbo capace (primi 10 livelli) e invece negli ultimi massacrate il party in modo umiliante, specialmente quando uno pensa che stava per vincere SOB!!.



Il gioco è la summa di tutte le ambientazioni create negli anni 80, ricrea lo stato d'animo di ansia, inquietudine, paura che un giocatore aveva quando era dentro il labirinto... Vecchi ricordi ritornano, antichi mormorii, rumori, odore di monitor crt surriscaldati.... I nati nel nuovo millennio hanno il giusto titolo che li può avvicinare a questo tipo di giochi...



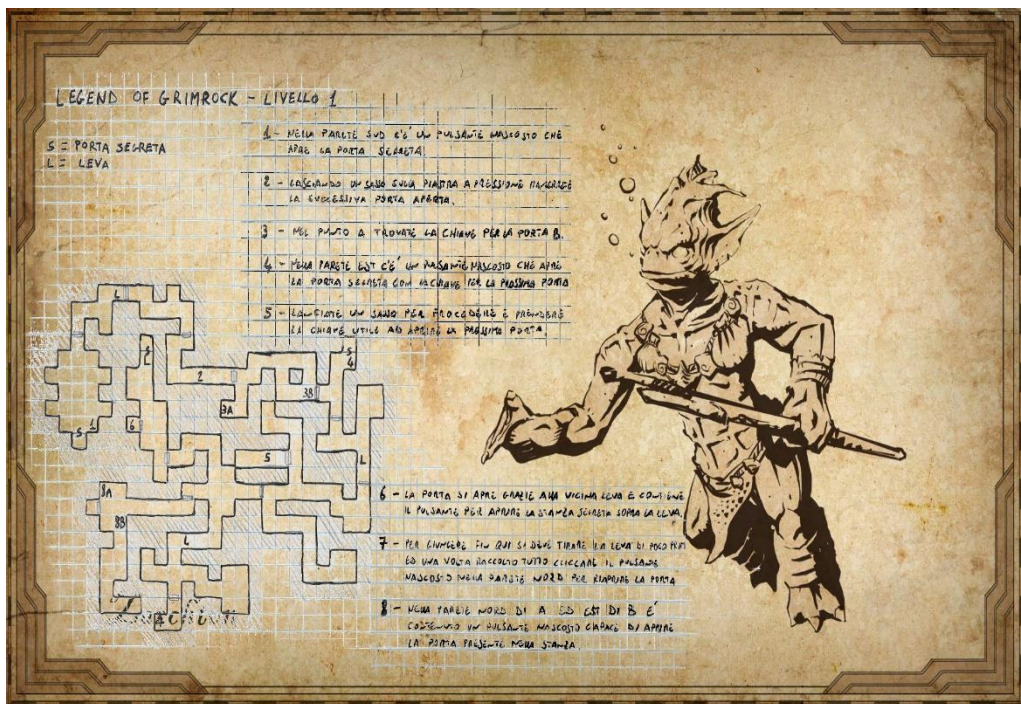
Giusto come aiutino, vincere non sarà facile in quanto senza la giusta arma il nemico è immortale, dato che molti giovani avventurieri non hanno mai iniziato questi giochi, diamo ai nostri giovani baldi eroi, la mappa insieme a delle info del primo livello faticosamente conquistato.

Per giocare ed immedesimarsi nel vero old rpg gamer consiglio di ricrearsi una delle seguenti schede e sopra di esse scrivere la vostra mappa, infatti nel vero labirinto non si ha la automappa!



Per i giocatori neofiti, dopo aver scelto la partita senza la auto-mappa, iniziate ad esplorare il labirinto seguendo una parete (esempio sempre il lato destro), e cominciate a disegnarla.

Quando trovate una intersezione seguite sempre la direzione presa a meno di indizi presenti sulle pareti o altro. A poco a poco

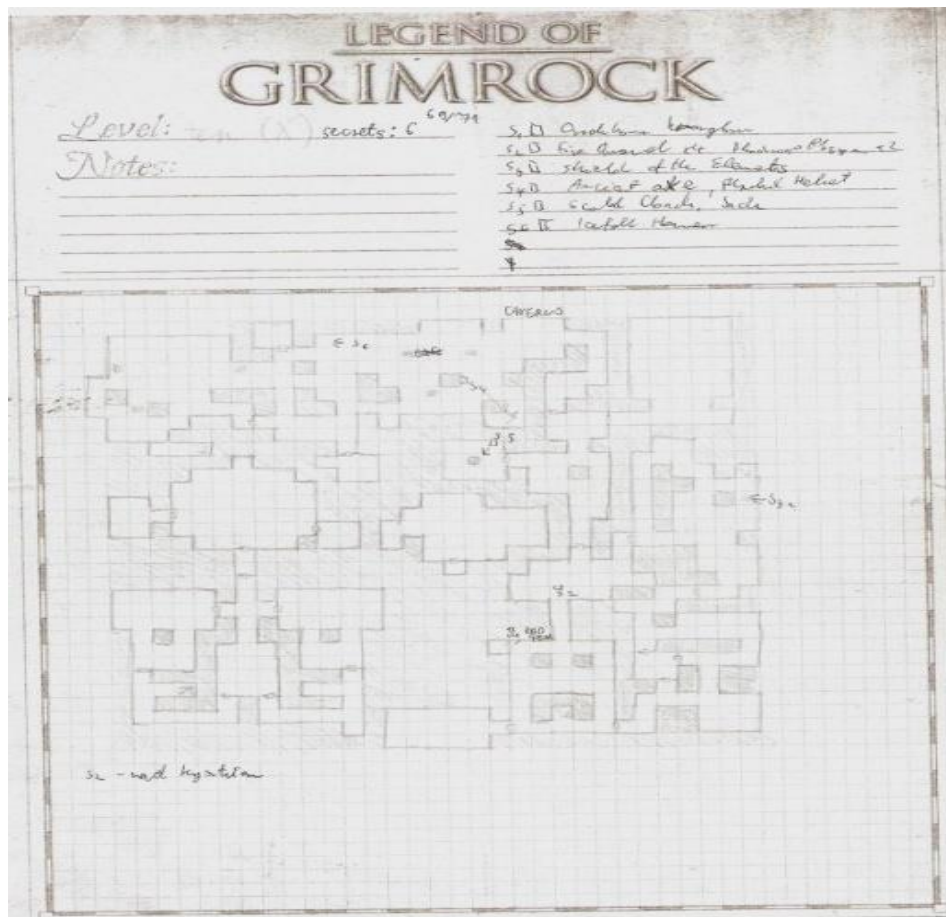


sarete in grado di disegnarvi la vostra mappa (vedi esempio sotto).

Per adesso non mi resta che augurarvi buon divertimento e se vi perderete nei labirinti di Grimrock non dovete fare altro che avvisarci, manderemo una nutrita schiera di navigati

avventurieri a cercarvi ed allo stesso tempo a cavarvi dagli impicci.

Un saluto dal vostro vecchio avventuriero, mi troverete come sempre alla taverna dell'aquila verde a bere birra in compagnia di amici o dare consigli e suggerimenti a voi giovani eroi.



Navigare nelle BBS con il C64 su VICE

di Marco Pistorio

E' possibile loggarsi all'interno delle BBS ormai raggiungibili comodamente tramite Internet e visitarle sfruttando l'emulazione Commodore64 di VICE? E' questo il quesito che mi è stato posto dall'amico **Giuseppe Monica**, che contestualmente saluto. Inizialmente pensavo non fosse possibile, non comunque sfruttando soltanto VICE. In effetti è necessario qualcos'altro, ma è una cosa fattibile, e proseguendo nella lettura di questo articolo scopriremo come fare. Intanto, una breve presentazione relativa alle BBS.

Un **BBS** (acronimo di **Bulletin Board System**) è un computer che utilizza un particolare software per permettere ad utenti esterni di connettersi ad esso attraverso la linea telefonica, dando la possibilità di utilizzare funzioni di messaggistica, file sharing centralizzato, ma anche semplici giochi di ruolo testuali, i **M.U.D. (Multi Users Dungeons)**. Questo sistema nasce negli anni '70 ed ha costituito il fulcro delle prime comunicazioni telematiche amatoriali.

I movimenti che diedero vita ai concetti di **shareware** e **software libero** sono nati nei BBS, attraverso lo scambio di software non protetto da limitazioni di copia, come per esempio **PKZIP** e le demo di **Wolfenstein 3D** e **Doom** (giochi divenuti celebri proprio grazie alla loro diffusione libera). Per gestire o utilizzare un BBS erano necessarie conoscenze tecniche più elevate della media e attrezzature particolari, il che rendeva solitamente i BBS regno di appassionati ed esperti.

Per un breve periodo di tempo, intorno agli anni '90, insieme ad un amico ne gestii uno personalmente, **JONICA BBS**. So quindi per esperienza diretta, non soltanto come utente, quanto impegno ci stava dietro la gestione di una BBS, quante cose era necessario configurare, verificare, aggiornare affinché tutto funzionasse per il meglio.

Oggi lo scenario è cambiato parecchio, ed in particolare da quando Internet è entrato prepotentemente nelle nostre case e nelle nostre vite.

I BBS raggiungibili oggi funzionanti sono pochissimi rispetto ad una volta, in compenso però è possibile accedere loro proprio tramite

Internet, sfruttando il protocollo telnet più precisamente.

Da questo punto dell'articolo in poi vedremo come navigare all'interno di un BBS con il Commodore 64 emulato tramite VICE su un normale PC

Cosa occorre ed impostazioni da verificare

A) tcpser

B) Programma terminale su C64, consiglio striketerm: <http://csdb.dk/release/?id=130807>

C) VICE versione 3.1

tcpser si occuperà di prendere tutto ciò che arriverà al PC da internet, mediante protocollo tcp-ip e lo invierà a VICE, in maniera bidirezionale.

Per fare funzionare correttamente tcpser serve anche **cygwin1.dll**.

Per automatizzare le cose ho creato un semplice file batch, go.bat il cui contenuto è il seguente:

```
tcpser.exe -v 25232 -s 2400 -p 23 -l 4
```

Per Vostra comodità ho creato un file .zip che contiene già tcpser, la suddetta dll ed il file batch sopra descritto.

Vi consiglio di scompattare da qualche parte nel vostro pc il file .zip e copiare poi i files tcpser.exe, cygwin1.dll e go.bat all'interno della cartella di WINVICE.

Da prompt di DOS, la prima cosa da dare è lanciare il file batch go.bat digitando semplicemente go e battendo invio, dopo essersi spostati sulla directory che lo contiene (consiglio WINVICE) e NON CHIUDERE la finestra all'interno della quale opererà.

Successivamente, lanciare l'emulatore VICE.

Nel menù Impostazioni, scegliere Impostazioni RS232, Impostazioni ACIA... e verificare quanto segue:

Voce "Abilita l'emulazione dell'interfaccia ACIA RS232" IMPOSTATA,
ACIA Device: RS232 Device 1;
ACIA Location: \$DE00;
ACIA Interrupt: NMI;
ACIA Mode: Normal.

Nella voce di menù successiva, Impostazioni USERPORT RS232...

"Abilita l'emulazione della userport RS232" IMPOSTATA

Userport device: RS232 device 1

Userport baud rate: 2400

Infine, nella terza ed ultima voce di menù relativa ai settaggi RS232, impostare quanto segue:

RS232 device 1 127.0.0.1:25232

RS232 device 2 127.0.0.1:25232

RS232 device 3 127.0.0.1:25232

RS232 device 4 127.0.0.1:25232

Salvare quindi le impostazioni.

Finalmente si naviga!

Aprire il file striketerm.d64 e lanciare il programma all'interno di VICE.

Verificare che il modem impostato sia: User port e che la velocità in Baud sia: 2400

Se così non fosse, cambiare tali impostazioni come sopra specificato.

Dare F5 (Dialer)

Premere la lettera 'a' per impostare la prima voce disponibile per la chiamata automatica.

Battere '3' e specificare un nome per la voce, ad esempio scorprio e dare invio.

Specificare IP Adr. Ad esempio scorp.us.to e dare invio.

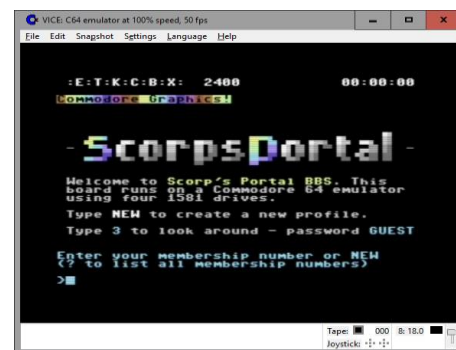
Specificare Port. Ad esempio 23 e dare invio.

Alla specifica del tipo di terminale, spostare il cursore fino a "Return to Menù" e dare invio.

Alla specifica ID battere semplicemente invio.

Alla specifica Passwd battere semplicemente invio.

A questo punto, avendo riempito opportunamente la voce 'a', basta premere invio per effettuare la chiamata e seguire le indicazioni via via fornite durante la sessione di terminale (esempio, battere Backspace, battere invio etc).



N.B. Procedura testata su Windows 10 con VICE 3.1. Scaricate il materiale a corredo dell'articolo da: <http://www.retro magazine.net/download/bbs.zip>

RetroGiochiAmo: Minesweeper

di Francesco Fiorentini

Nel numero 3 di RetroMagazine abbiamo visto come *installare Windows 3.1 su una macchina virtuale utilizzando VirtualBox*, ma ci siamo fermati appena dopo l'installazione e non abbiamo fatto niente altro. Lanciamo quindi Virtualbox ed avviamo la nostra macchina virtuale contenente Windows 3.1 per accedere al gioco **Minesweeper** (o **Campo minato** nella versione italiana).

Prima di passare a spiegare come giocare, permettetemi di spiegare il perché della scelta di questo particolare software. Tutti più o meno hanno giocato almeno una volta a Minesweeper, ma per quanto possa sembrare un gioco semplice ho riscontrato che amici e colleghi non avevano capito come giocare effettivamente (usando i pulsanti del mouse in modo corretto) e cliccavano abbastanza casualmente sulle caselle finendo inesorabilmente per inciampare in una mina e morire...

L'obiettivo del gioco è facilmente intuibile, ci troviamo all'interno di un campo minato e dobbiamo scoprire l'esatta posizione di tutte le mine per terminare vittoriosi la partita.

Per rendere l'esperienza di gioco sfidante e divertente al punto giusto, vi suggerisco di impostare la dimensione dell'area di gioco a 24x30 con 150 mine. Una volta avviata la partita, il gioco ci assicura che il primo click del mouse non incontrerà nessun mina, ma i successivi saranno a rischio. Inoltre, per assicurare una partita divertente suggerisco di cliccare fino a quando non troviamo un'area vuota. Quando troviamo una casella vuota cioè non contenente né una mina né l'indicazione di una mina nelle vicinanze, tutte le altre caselle vuote adiacenti verranno scoperte permettendoci quindi di poter cominciare a giocare.

Ok, adesso abbiamo un'intera area ripulita e vediamo una serie di numeri adiacenti alle caselle ancora coperte; cosa significano? Ogni numero sta ad indicare la quantità di mine presenti nelle 8 caselle che circondano il numero stesso.

*		*
*	3	

Nell'esempio sopra (tab. 1), il numero ci indica che ci sono 3 mine a ridosso della casella; sta a noi scoprire dove sono e segnalarle correttamente. Ovviamente un numero singolo non è sufficiente e quindi dobbiamo interpretare gli altri numeri adiacenti e le caselle già bonificate per riuscire nella nostra impresa (tab. 2). Onestamente qui c'è poco da aggiungere, serve un ragionamento logico per capire dove sono tutte le mine e contrassegnarle. Passiamo invece a descrivere le funzionalità dei tasti del mouse perché non tutti le conoscono correttamente.

Il **tasto destro** ha due funzionalità; alla prima pressione verrà posta una **bandierina** sopra la casella interessata mentre ad una successiva pressione apparirà sulla casella un **punto interrogativo**. Vediamo a cosa servono.

La **bandierina** indica esplicitamente al programma che in quella casella siamo sicuri sia presente una mina! *Attenzione a piazzare bene le bandierine, perché saranno fondamentali nel prosieguo della nostra spiegazione e del gioco!*

Il **punto interrogativo** invece ha puramente una funzione visiva per aiutarci nel ragionamento per la scoperta delle mine. Personalmente lo trovo inutile e non lo utilizzo, ma può darsi che altri giocatori lo trovino utile.

Il **tasto sinistro** invece ha soltanto la funzionalità di **scoprire la casella** cliccata. Se la casella non contiene nessuna mina il gioco prosegue, altrimenti

saremo inesorabilmente morti.

La funzionalità ad alcuni sconosciuta invece è la **pressione contemporanea di entrambi i tasti del mouse**. Vediamo di capire insieme a cosa serve.

La pressione contemporanea di entrambi i tasti del mouse ha la funzionalità di indicare visivamente quelle che sono le caselle adiacenti al click e, **nel caso ci siano bandierine sufficienti a coprire il numero di mine indicato, rilasciando la pressione a scoprirle**. Attenzione quindi a piazzare le bandierine correttamente perché con questa funzione, basta un errore per terminare la partita anzitempo! ☺

Bene, adesso avete in mano le chiavi per terminare con successo ogni partita di Minesweeper, ma come scoprirete potrebbe non essere sempre sufficiente. Mi è capitato più di una volta di rimanere con solo due caselle contenenti una sola mina ma con indicazioni insufficienti per capire quale delle due fosse minata e quale vuota...

Che aspettate quindi, riprendete in mano RetroMagazine 3, avviate/installate Windows 3.1 e... se avete domande, sono qui! ☺



Aleph-project di Isacco Luongo

la CPU Z80 diventa a 24 bit di indirizzamento!

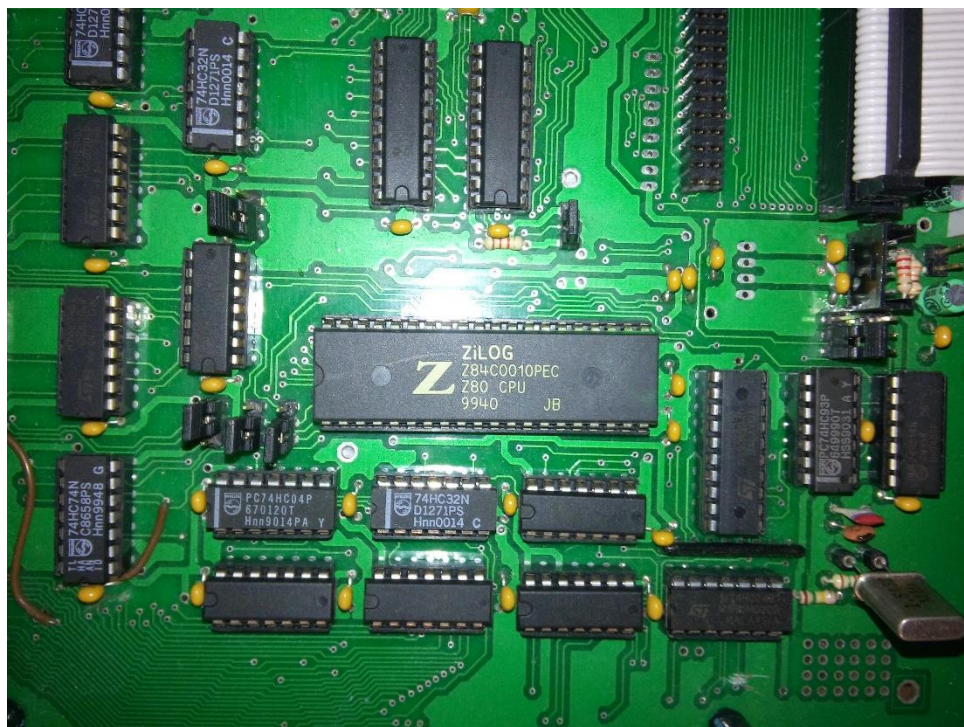
di Marco Pistorio

L'autore del progetto **Isacco Luongo**, contattato dalla nostra redazione ci rivela alcuni dettagli esclusivi.

Come tutti sappiamo il microprocessore Zilog Z80 è classificato come cpu a 16 bit di indirizzamento e 8 bit dati, quindi può indirizzare solo 65.536 bytes di memoria complessiva, sia essa ram e/o rom.

Si tratta di una CPU che esiste da oltre 40 anni ed è stata studiata in maniera approfondita dai più grandi studiosi del pianeta, la lista di home-computer che implementano lo Z80 come processore è lunghissima, ed è passata pure sotto la lente di osservazione dell'ing. Kazuhiko Nishi che ne tirò fuori lo standard MSX, questa CPU è stata per lungo tempo il cavallo di battaglia del nobile anglosassone Clive Sinclair, che inventò lo Spectrum, ovvero l'anti-commodore per eccellenza, prima che nascesse la disputa Atari-Amiga.

Ovviamente il declino delle vendite di questo gioiello di tecnologia dal cuore italiano, ha una data ben precisa, **e coincide con la nascita dell'intel 8088 e del primo computer IBM a 20 bit di indirizzamento.** Quindi la data



coincide con l'origine del duopolio Intel-Microsoft.

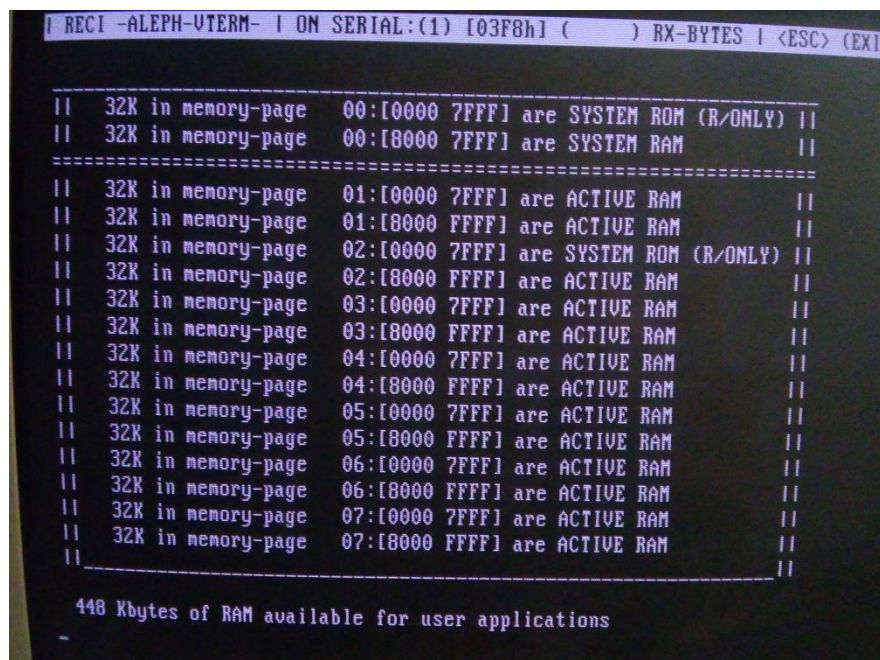
L'universo dei sistemi satelliti, basati sulle varianti del sistema CP/M come Osborne, North-Star, Cromemco, che pure ci fornì la

"Dazzler" prima scheda-video a colori della storia del computer finì per soccombere sotto i colpi ben assestati dalla "nuova creazione di IBM". Ma cosa c'era in quel computer che agli altri mancava? Come tutti sappiamo erano i tempi in cui Bill Gates pronunciò quella ingloriosa frase: "640k di memoria ram possono bastare a chiunque".

Ecco cosa mancava agli altri, la memoria. Ed IBM quasi la regalava, secondo il mito dell'epoca.

Ed ora torniamo ai nostri giorni, e l'autore del progetto Isacco Luongo con aria serafica, ci rivela che in realtà le tante aziende che fallirono in America ed altrove, lo fecero perché non impararono mai a sfruttare appieno le capacità offerte dalla CPU Z80, fu un errore che ha origini lontane, da ricercarsi nel patriottismo del militare della marina statunitense Gary Kildall.

Dice Isacco Luongo: **perché Gary Kildall non basò il sistema operativo CP/M sullo Z80 ma sul suo antenato 8080 di intel.** Così quelle caratteristiche e innovazioni volute dal fisico Veneziano Federico Faggin furono del tutto



ignorare nei sistemi basati su CP/M, quindi l'epilogo fu irreversibile.

Intanto, oggi dopo 37 anni da quel fatidico 12 agosto 1981 scopriamo grazie alle ricerche di uno studioso autodidatta che la CPU Z80 può indirizzare con un piccolo aiutino, fino a 32 megabyte di memoria, ed anche oltre con alcuni accorgimenti di progetto.

Sicuramente l'autore per aiutino intende, grossi micro-controllori in tecnologia FPGA, oppure annovera nel progetto chip di parti terze che svolgono il lavoro al posto della CPU? **Niente affatto, risponde il progettista Isacco Luongo, nel mio progetto ci sono solo porte logiche e qualche comparatore di uguaglianza. – Il lavoro sporco lo fa tutto il processore Z80, ed ha un registro speciale per indirizzare le pagine di memoria estesa.** – un indirizzo esteso ha la stessa forma di un indirizzo intel 8088, in modalità reale, solo che Aleph non segmenta, e gli indirizzi sono univoci.

Soffermiamoci su questo punto; la CPU intel 8088 può indirizzare la stessa locazione in 4096 modi diversi, e solo gli otto bit più a sinistra del byte di segmento sono realmente significativi; invece per Aleph ad ogni numero corrisponde un indirizzo diverso, quindi è un indirizzamento lineare, non segmentato, ma contiguo.

Ad esempio per indirizzare l'indirizzo posto a 256k dall'indirizzo nullo su un intel 8088 scriveremmo in hex 4000:0000h: in ambiente Aleph lo stesso indirizzo lo raggiungiamo con lo Z80 con una coppia di registri che contiene il seguente indirizzo hex 04:0000h oppure linearmente come accade con il MC68000 in hex 4.0000h.

Chiaramente l'indirizzo più alto ottenibile in questo modo sarebbe hex 0FF:FFFFh, ovvero quell'indirizzo di memoria che dista 16.777.215 bytes dall'indirizzo NULL.

Per eseguire il dump della memoria si inserisce il seguente comando:

```
dump PP.AAAA <invio>
```

analogamente per inserire in memoria un testo ascii si scrive:

```
insert PP.AAAA <invio>
```

dove pp è il numero di pagina ad 8 bit + AAAA che rappresenta l'offset a 16 bit, per un totale di 24 bit lineari di indirizzamento. Con pagine da 64k ciascuna.

Ovviamente è possibile avere "codici" in una pagina e "dati" un'altra pagina situati ovunque nello spazio dei 16 megabytes, non solo lo Z80 può caricare programmi e dati che comprendono più pagine di 64k ciascuno, può saltare da una pagina ad una subroutine che si trova in un luogo remoto dei 16 megabytes e tutto questo si poteva fare già nell'anno 1981. Infatti l'autore del progetto dichiara, ho usato solo componenti che già esistevano in quegli anni, per dimostrare che la CPU Z80 poteva dare molto di più ma gli si chiedeva solo di funzionare a metà.

Infatti si ricorda che, lo Z80 aveva un doppio set di registri, era una specie di PENTIUM 4, HIPER-THREADING, un dual-core 8080 con una CPU vera ed un altro core virtuale posto in una memoria/registro selezionabile all'occorrenza per cambiare task o contesto di sistema.

L'autore di Aleph conclude dicendo, sono un Amighista purosangue, ma per colpa del mio professore di sistemi all'Istituto Tecnico Industriale di Avellino Giuseppe Pastena mi sono ritrovato a studiare la CPU Z80, per me era l'anti-commodore, ma poi ho dovuto rivedere la mia posizione "almeno sul processore".

Il progetto oramai non è più solo un progetto, su carta, è cresciuto e non mi basta più una scrivania per poter contenere le sue schede. – E fa un bel po' di cose, tra breve avrà una scheda video tutta sua così staccherà definitivamente i suoi cordoni ombelicali ovvero (i cavi nullmodem) che collegano Aleph al mondo esterno.

Dal punto di vista software sto scrivendo il primo assembler a 24 bit per CPU Z80, così potrà eseguire programmi complessi con eseguibili lunghi a piacere fino a 16 o 32 megabytes. (Dipende da come vorrò giocarmi l'ultimo bit di indirizzamento).

Per ora il sistema comunica con un programma di terminale che ho dovuto scrivere per dos/freeDOS, che si chiama R.E.C.I. – v.1.05 – il cui listato è grande quanto il dizionario di greco di mia figlia Sara!

Sapresti quantificare in qualche modo il tempo impiegato per portare avanti fino a questo momento "aleph" ed i relativi costi?

Il tempo impiegato coincide con il tempo che mi rimane libero, dopo il tempo da dedicare al lavoro, alla famiglia ed al fisiologico tempo di break "no-code time". Ora siamo in un tempo di break. – Ma tra pochi giorni, staccherò tutti i contatti internet compreso e via, full-immersion nel progetto. – Ho classificato il periodo di tempo dal 2006, al 2008 come progetti pre-Aleph. – Dal 2008 al 2016 nasce Aleph-o. --Nel frattempo ho scritto un libro di storia civica sul mio paese l'ho pubblicato e venduto (nel periodo 2012/2013), ho partecipato alla vita sociale e amministrativa del mio paese di residenza. – Ho iniziato a pensare seriamente ad Aleph solo nell'estate del 2016. Ma il progetto ha coinvolto solo e soltanto una sola persona, ho già litigato altre volte con soci fannulloni ed incompetenti e soprattutto disonesti. Una volta feci entrare un amico nel locale dove tengo le schede, e da allora non trovo più il mio unico chip della cpu intel 8080 con piastra in oro. (Quindi ho smesso di fare visite guidate). Siete i primi ad entrare "virtualmente" nel progetto e la fiducia è ai minimi storici, mentre l'autostima cresce col crescere delle cose che aleph riesce a fare.

Dopo una doverosa introduzione, veniamo ad una serie di domande dirette a cui Isacco ha gentilmente accettato di rispondere in maniera esauriente.

Se dovessi esprimere, in percentuale, lo stato di avanzamento, a quale valore ci troveremmo secondo te?

Il peggio è passato. Ho diviso il progetto "hardware" in tre parti fondamentali.

a) la fase sperimentale, non sai quasi nulla su quello che può succedere quindi fai grandi schede con più esperimenti, e poi connetti e o bypassi le parti in base al funzionamento durante i test (in sintesi le schede grandi costano meno di quelle piccole, così ho raggruppato più cose da testare su un'unica scheda)... questa fase è alle spalle (Aleph-o).

b) Aleph-1: la fase retro-technology, la vecchia cpu Z80 con i compagni di sempre, le porte logiche TTL/HCMOS fa più cose perché finalmente si esprime con la sua piena potenza. – Grazie alla fase sperimentale, ora sappiamo in che modo rendere più efficace la comunicazione tra parti del sistema. – Ho

creato nuovi segnali formattando quelli emessi dal processore, la fase sperimentale è stata la più lunga ma anche la più produttiva. (Niente stress, ho rifiutato eventi e presentazioni inutili, non ho fretta né un direttore col fiato sul collo, mai pensato a risparmiare). – In questa fase l'unico obiettivo è fare il salto di qualità. – Diamo al processore tutto quello che vuole e vediamo dove ci porta. – Chiunque avrebbe potuto farlo, ma nessuno ha mai ascoltato il processore. – [Esempio banale: appena si accende il processore genera l'indirizzo 00:0000, li dovrebbe esserci la rom-BIOS. Ma il CP/M vuole che tale indirizzo sia "ram" per scaricarci (da floppy) i puntatori fondamentali del sistema, il CP/M non rispetta nemmeno i vincoli dell'intel 8080]. – Ecco perché nei primi Radio Shack TRS-80 quell'indirizzo era in Rom, e per lo stesso motivo quelle macchine ebbero problemi ad installare il CPM).

C) Aleph-1 (mini) – (all to do) Per il momento questa fase è tutta da fare. Ed avrà le sue difficoltà, però si basa su solide fondamenta, quindi non vedo la stessa frustrazione della fase sperimentale. Lì ho visto le cose più strane, ho visto collegamenti fatti per sbaglio funzionare, e circuiti da manuale fallire i test. – Ho dovuto più volte prendere nota dalla realtà e chiedermi perché sta funzionando? – Nella fase mini, tutto il presente deve essere semplicemente implementato in un chip ALTERA CPLD/FPGA. – E tutto il progetto diventerà grande quanto un francobollo, e viaggerà anche a velocità superiori a 500 Mhz. – oggi siamo tra i 4 o max 12 Mhz. – Il semplice cambio di tecnologia ci darà la velocità e l'integrazione VLSI. – Senza nessuno sforzo. – Diciamo che sono al 60 % dall'obiettivo che mi ero prefissato nel l'estate del 2016. – Ma in termini di tempo l'esperienza precedente si concatena e gli strumenti precedenti ora si utilizzano, "l'esponenziale comincia a salire" – quindi non impiegherò altri due anni per presentare il prototipo, avremo qualcosa di funzionante – "e comprensibile anche per i profani dell'elettronica" per natale di quest'anno. –

Successivamente allo stadio "prototipo", quali saranno i passi successivi che intendi percorrere?

Anno 2019, inizia la "presentation phase". Tutti dovranno conoscere la potenza e la

semplicità di un CORE Z80 che si esprime liberamente ovunque vi sia un transistor ed una tensione compatibile, lì ci può stare aleph. – Anche perché, tutto il codice Z80 dall'anno 1976 ad oggi potrebbe girare su Aleph senza modifiche. – Anche il CP/M potrà girare su Aleph ma sarà un sotto-sistema, da eseguire



in una paginetta di memoria qualunque. – Di questa fase ne parleremo in seguito. –

Comprendiamo che sfruttare le potenzialità dello Z80 e renderlo capace di un indirizzamento di memoria di tutto rispetto è per te fonte di soddisfazione, risultato legato certamente ad uno studio profondo ed a notevoli sforzi, di tempo libero e non soltanto. Quali sono le tue aspettative in merito ad "aleph"?

Non credo di aver fatto qualcosa di importante, questo deve essere il mercato e la società a dirlo, solo se, ritengono che bisogna dire basta a codici lenti e voluminosi che non fanno altro che intasare le risorse di rete e di sistema. Ma è possibile che un sistema come Windows ma anche linux, per generare un semplice programma che scriva "hello world" a video debba linkare centinaia di library ed occupare di base 500k memoria? – Usando il buon vecchio rasoio di Occam noto dall'anno 1300 circa, ci insegna: "A parità di condizioni, la via da preferire è quella più semplice".

Prevedi uno sbocco economico che possa ripagare in qualche modo i tuoi sforzi?

Sarebbe auspicabile, ma sono pragmatico "solo i tedeschi con l'euro pesante" forse potrebbero recepire il segnale, e magari potrebbero inglobare il progetto in qualcosa che loro già stanno facendo in altro modo. Tuttavia, non credo che in Italia progetti come questo possano ricevere il giusto credito né mi illudo di presentare il prodotto in Italiano. –

Certo, non imparerò il tedesco, però sono convinto che la pubblicazione del manuale sarà "only English". – L'Italia è il paese che ha fatto fallire riviste come Nuova-Elettronica, Fare Elettronica "sostenuta dalla Micorchip" diventa online, quindi mi meraviglia il vostro interesse per la materia, che vedo positivo ma in netta controtendenza rispetto alla massa che resta totalmente indifferente. –

Hai pensato a realizzare una raccolta fondi per completare il tuo progetto?

No, e non credo proprio che lo farò mai. Perderei tutta la mia creatività, avrei dei padroni a cui rispondere, e mi comincerebbero a chiedere come hanno speso i loro soldi. No mi dispiace, non è per me, amo troppo la mia libertà, dovrei incontrare solo un direttore o dei soci più creativi e fantasiosi e più bizzarri di me, per poterci andare d'accordo. La raccolta fondi la facciano per combattere il diabete mellito-1, o vadano agli istituti di ricerca seri, io sono un "hobbyist", autodidatta.

Ritieni questa idea percorribile e soprattutto utile?

Se Aleph dovesse entrare nelle macchine elettro-medicali, in un misuratore di glicemia, o in un misuratore di pressione sanguigna allora credo che sarebbe non solo utile, ma etico e percorribile il cammino della fondazione, con la raccolta fondi e quello che vi orbita intorno. – Ma per ora "il mio pragmatismo" interviene, tuttavia mi metto in ascolto, se c'è qualcosa che non conosco, potrei valutare e decidere, possibile ma improbabile allo stato attuale.

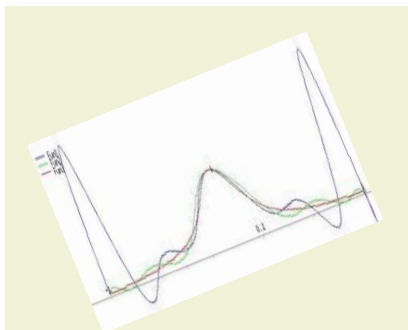
Per ora mi fermo qui. Ciao, buona serata, grazie ancora x aver voluto condividere con noi questo ambizioso ed estremamente interessante progetto e... a presto! Attendo tue notizie.

Avrei preferito sentire bussare il campanello, avrei preferito farti accomodare nel salotto, ed avrei preferito fare queste due chiacchiere con te dal vivo, ma riesco ad abituarci ai tempi nuovi. Meno salotti, più monitor e tastiere per comunicare...

Per saperne di più sul progetto Aleph di Isacco Luongo, seguite la pagina Facebook: <https://www.facebook.com/projectalfazero/>

Retromath: a spasso tra i dati mancanti

di Giuseppe Fedele



Cosa significa approssimare una funzione o un insieme di dati con una funzione? E' possibile ricavare dati mancanti avendone a disposizione altri?

In queste pagine introduciamo uno dei problemi più antichi e, forse, più affascinanti della matematica: l'interpolazione...

...e vediamo, inoltre, come realizzare un algoritmo di interpolazione utilizzando il BASIC del Commodore 64.

$$\binom{m+n}{r} = \sum_{k=0}^r \binom{m}{k} \binom{n}{r-k}$$

Il problema di costruire una funzione definita a partire da una serie di valori numerici è inevitabile in tutti quei casi in cui è necessario estrarre delle informazioni non incluse o non contenute in modo esplicito nei dati stessi.

L'approccio relativamente più semplice e più utilizzato per risolvere questo problema è l'interpolazione, in cui una funzione approssimante viene costruita in modo tale da concordare perfettamente con la funzione incognita nei punti di misura.

Una breve cronologia

La parola "interpolazione" deriva dal verbo latino "interpolare", una contrazione di "inter", che significa "tra", e "polare", che significa "lucidare". Vale a dire, "completare l'andamento di una funzione o di un grafico all'interno di un dato intervallo a partire dai valori noti in certi punti isolati dell'intervallo".

Esempi di campi in cui questo problema si pone naturalmente sono l'astronomia e, in relazione a questo, il calcolo del calendario.

Per quanto riguarda l'astronomia, alcune delle prime prove, giunte fino a noi, dell'uso dell'interpolazione provengono dall'antica Babilonia e dalla Grecia. Intorno al 300 a.C., gli astronomi babilonesi utilizzavano non solo forme lineari, ma anche forme più complesse di interpolazione per prevedere le posizioni del sole, della luna e dei pianeti di cui erano a conoscenza, come si può trovare sulle antiche tavolette cuneiformi astronomiche (Figura 1) provenienti da Uruk e Babilonia nel periodo Seleucido (gli ultimi tre secoli a.C.).



Figura 1. Antica tavoletta astronomica cuneiforme (300 a.C.)

Anche in Grecia, intorno al 150 a.C., Ipparco di Rodi usò l'interpolazione lineare per costruire una "funzione corda", simile a una funzione sinusoidale, per calcolare le posizioni dei corpi celesti. Esempi successivi si trovano nell'Almagesto ("The Mathematical Compilation", ca. 140 d.C.) di Claudio Tolomeo, l'astronomo-matematico greco di origine egiziana che ha proposto la visione geocentrica dell'universo che ha prevalso fino al XVI secolo. Più a est, le prove cinesi dell'interpolazione risalgono al 600 d.C. circa. Liu Zhuo utilizzò l'interpolazione di secondo ordine per costruire un "Calendario Imperiale Standard". Nel 625 d.C., l'astronomo e matematico indiano Brahmagupta introdusse un metodo per l'interpolazione di secondo ordine della funzione sinusoidale e, successivamente, propose un metodo per l'interpolazione di dati non equispaziati.

Un'altra regola per la realizzazione delle interpolazioni di secondo ordine (Figura 2) si trova in un commento attribuito a Govindasvami (ca. 800-850 a.C.).

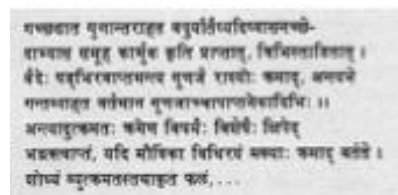


Figura 2. Antico passaggio sanscrito (ca. 800-850 a.C.)

La teoria dell'interpolazione nei paesi occidentali ha iniziato a svilupparsi solo dopo una grande rivoluzione nel pensiero scientifico. In particolare i nuovi sviluppi nel campo della fisica e dell'astronomia, iniziati da Copernico, proseguiti da Keplero e Galileo e culminati nelle teorie di Newton, diedero un forte impulso al progresso della matematica.

Non c'è una sola persona che abbia fatto così tanto per questo campo, come per tanti altri, come Newton. Newton inizia il suo lavoro fondamentale sull'argomento e alla fine getta le basi della teoria classica dell'interpolazione. I suoi contributi possono essere trovati in:

- Una lettera a Smith, datata 8 maggio 1675, e anche una lettera a Oldenburg, datata 24 ottobre 1676, in quest'ultima delle quali scrisse: (...) *Ho un altro metodo non ancora pubblicato per affrontare facilmente il problema. Si basa su una soluzione comoda, pronta e generale di questo problema. Per descrivere una curva geometrica che passa per un insieme di punti dati. (...) Anche se a prima vista può sembrare intrattabile, è tuttavia vero il contrario. Forse è davvero uno dei problemi più belli che io possa mai sperare di risolvere.*
- Un manoscritto intitolato *Methodus Differentialis*, pubblicato nel 1711, ma probabilmente scritto a metà degli anni Settanta (Figura 3).
- Un manoscritto intitolato *Regula Differentiarum*, scritto nel 1676 ma scoperto per la prima volta nel XX secolo (Figura 4).
- Lemma V dal Libro III della sua celebre *Philosophiae Naturalis Principia Mathematica*, pubblicata nel 1687. Questo Lemma contiene la sua formula generale di interpolazione per dati non equispaziati (Figura 5).

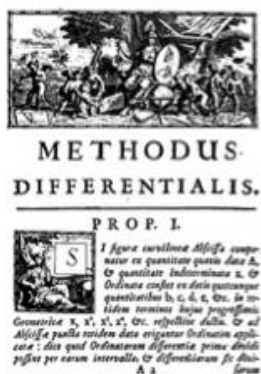


Figura 3. Methodus Differentialis, 1711



Figura 4. Regula Differentiarum, 1676



Figura 5. Philosophiae Naturalis Principia Mathematica, 1687

Una rappresentazione alternativa molto elegante della formula generale di Newton, che non richiede il calcolo di differenze finite o divise, fu pubblicata nel 1779 da Waring. Oggi si è soliti attribuirlo a Lagrange che, in apparente ignoranza del lavoro di Waring, la pubblicò 16 anni dopo.

All'inizio del XX secolo, il problema dell'interpolazione per differenze finite o divise era stato studiato da astronomi, matematici, statistici che avevano elaborato la maggior parte delle varianti delle ben note formule originali di Newton. Già nel 1821, Cauchy studiò l'interpolazione come rapporto di due polinomi e dimostrò che la soluzione a questo problema è unica, la formula Waring-Lagrange è un caso speciale quando il polinomio al denominatore è pari a uno.

Una generalizzazione di diversa natura fu pubblicata nel 1878 da Hermite, che studiò e risolse il problema di trovare un polinomio di cui anche le prime derivate assumono valori prestabiliti in determinati punti, dove l'ordine della derivata può differire da un punto all'altro. In un articolo pubblicato nel 1906, Birkhoff studiò un problema ancora più generale: dato un qualsiasi insieme di punti, trovare una funzione polinomiale che soddisfi criteri prestabiliti riguardanti il suo valore e/o il valore di uno qualsiasi delle sue derivate per ogni singolo punto.

Il problema

Assegnate $n + 1$ coppie (x_i, y_i) dove y_i rappresenta il valore misurato in corrispondenza di x_i , trovare un polinomio $p(x)$ tale che

$$p(x_i) = y_i, \quad i = 0, \dots, n.$$

Esempio: Si rileva la temperatura in alcune ore della giornata (Tabella 1) e si vuole sapere l'andamento della temperatura in istanti della giornata in cui non è stata rilevata, ma comunque compresi tra l'istante iniziale e finale in cui sono avvenute le nostre rilevazioni. La Figura 6 mostra i dati e il polinomio che li interpola. Tale polinomio passa esattamente per i punti misurati.

Tabella 1. Misure di temperatura

Ora	6	8	11	14	16	18	19
T°	4	7	10	12	11.5	9	7

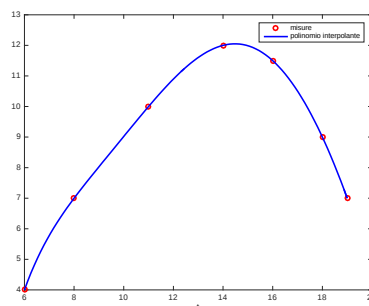


Figura 6. Esempio di interpolazione

Il problema si risolve individuando un polinomio di grado n che consenta quindi di valutare il fenomeno anche in istanti diversi da quelli a disposizione. Le quantità x_i in cui si hanno a disposizione le misure (istanti temporali nell'esempio precedente) vengono solitamente chiamati **nodi**. Mettiamoci nell'ipotesi in cui tutti i nodi siano distinti tra di loro (questo vuol dire che tutte le misurazioni avvengono in istanti diversi).

Come facciamo a trovare il polinomio interpolante? Iniziamo con l'osservare che, avendo a disposizione $n + 1$ nodi/misurazioni si hanno $n + 1$ gradi di libertà relativi agli $n + 1$ coefficienti di un polinomio (che quindi sarà di grado n : un polinomio di grado n è univocamente determinato dalle sue $n + 1$ radici).

Se indichiamo con $p(x)$ un generico polinomio di grado n , ovvero

$$p(x) = \sum_{i=0}^n a_i x^i$$

imponendo le condizioni di interpolazione $p(x_i) = y_i$, $i = 0, \dots, n$, si ottiene un sistema di $n + 1$ equazioni in $n + 1$ incognite (le incognite sono appunto i coefficienti a_i del nostro polinomio):

$$a_0 + a_1 x_0 + a_2 x_0^2 + \dots + a_n x_0^n = y_0$$

$$a_0 + a_1 x_1 + a_2 x_1^2 + \dots + a_n x_1^n = y_1$$

$$\vdots$$

$$a_0 + a_1 x_n + a_2 x_n^2 + \dots + a_n x_n^n = y_n$$

o equivalentemente

$$Va = y$$

dove $a = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix}$, $y = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$ e la matrice $V \in \mathbb{R}^{(n+1) \times (n+1)}$, legata ai nodi, ha la seguente struttura:

$$V = \begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix}$$

Questa matrice è nota come **matrice di Vandermonde** in onore del matematico francese Alexandre-Théophile Vandermonde che, per primo, ne studiò le sue proprietà. Se tutti i nodi di interpolazione sono distinti allora la matrice può essere invertita e quindi il problema dell'interpolazione ammette un'unica soluzione. All'aumentare dei nodi e quindi dei dati da interpolare questa matrice diventa però difficile da invertire e gli algoritmi numerici di inversione falliscono nel trovare la corretta soluzione al problema dell'interpolazione (si parla di malcondizionamento della matrice).

Per risolvere il problema si preferisce quindi usare una diversa rappresentazione del polinomio interpolante

$$p(x) = \sum_{i=0}^n y_i L_i(x)$$

dove gli $L_i(x)$ sono polinomi di grado n noti con il nome di **polinomi di Lagrange**. Imponendo le condizioni di interpolazione, ovvero il passaggio del polinomio per i dati misurati, si ha:

$$p(x_j) = \sum_{i=0}^n y_i L_i(x_j) = y_j, \quad j = 0, 1, \dots, n$$

quindi i polinomi $L_i(x)$ devono soddisfare le seguenti proprietà

$$L_i(x_j) = \begin{cases} 0 & \text{se } j \neq i \\ 1 & \text{se } j = i \end{cases}$$

E' facile dimostrare allora che i polinomi avranno la seguente struttura:

$$L_i(x) = \frac{\prod_{j=0, j \neq i}^n (x - x_j)}{\prod_{j=0, j \neq i}^n (x_i - x_j)}$$

Se abbiamo ad esempio 4 nodi, allora i polinomi di Lagrange saranno:

$$L_0(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)}$$

$$L_1(x) = \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)}$$

$$L_2(x) = \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)}$$

$$L_3(x) = \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)}$$

Anche questa formulazione del polinomio interpolante non si presta ad una implementazione efficiente su calcolatore. Inoltre ogni volta che si aggiunge una nuova misurazione occorre ricalcolare tutti i polinomi di Lagrange e riavviare quindi l'algoritmo di interpolazione.

L'idea di Newton fu quella di ottenere una versione ricorsiva del polinomio interpolante di grado $n + 1$ a partire dal polinomio interpolante di grado n . Dimostrò infatti che vale il seguente risultato:

Sia $p_n(x)$ il polinomio di grado n che interpola una funzione $f(x)$ nei nodi $x_0, x_1, \dots, x_n$. Sia $p_{n+1}(x)$ il polinomio di grado $n + 1$ che interpola la $f(x)$ nei nodi $x_0, x_1, \dots, x_n, x_{n+1}$. Allora $p_{n+1}(x)$ può essere scritto come

$$p_{n+1}(x) = p_n(x) + a_{n+1} \omega_n(x)$$

dove

$$\omega_n(x) = \prod_{i=0}^n (x - x_i)$$

$$a_{n+1} = \frac{f(x_{n+1}) - p_n(x_{n+1})}{\omega_n(x_{n+1})}$$

$$a_0 = f(x_0)$$

Un famoso algoritmo di interpolazione basato sull'idea di Newton fu pubblicato nel 1970 ad opera di Björck e Pereyra (Figura 7), ed è proprio questo algoritmo che vogliamo qui riproporre in codice retrò.

MATHEMATICS OF COMPUTATION, VOLUME 24, NUMBER 112, OCTOBER 1970

Solution of Vandermonde Systems of Equations

By Åke Björck* and Victor Pereyra

Abstract. We obtain in this paper a considerable improvement over a method developed earlier by Bulster and Pereyra for the solution of systems of linear equations with Vandermonde matrices of coefficients. This is achieved by observing that a part of the earlier algorithm is equivalent to Newton's interpolation method. This allows also to produce a progressive algorithm which is significantly more efficient than previous available methods. Algol-60 programs and numerical results are included. Confluent Vandermonde systems are also briefly discussed.

Figura 7. Articolo di Björck e Pereyra del 1970

Il codice proposto legge il numero di nodi N , i nodi $X(i)$ e i valori della funzione da interpolare valutata sui nodi $Y(i)$ (rispettivamente in riga 5000, 5010 e 5020) e calcola i coefficienti del polinomio interpolante di grado $N - 1$ (il vettore C).

Si ma adesso? Il problema come detto è utilizzare il polinomio interpolante

$$p(x) = \sum_{i=1}^N C(i) x^{N-i}$$

per ricavare il valore della funzione in un altro punto in cui non si hanno misure. Si deve cioè essere in grado di valutare il polinomio per un generico valore di x .

Per la sua valutazione in un punto x occorrerebbero $N - 1$ addizioni e $2N - 3$ moltiplicazioni. Se consideriamo invece la formulazione equivalente

$$p(x) = \left(\left(\left(C(1)x + C(2) \right)x + C(3) \right)x + \dots \right)x + C(N-1) + C(N)$$

allora per la stessa valutazione bastano $N - 1$ addizioni e $N - 1$ moltiplicazioni. L'algoritmo di valutazione (noto come **algoritmo di Horner**) è riportato di seguito.

Algoritmo di Björck e Pereyra su C64

```
100 READ N
110 FOR I=1 TO N
120 : READ X(I)
130 NEXT I
140 FOR I=1 TO N
150 : READ Y(I) : A(I)=Y(I)
160 NEXT I
1000 FOR K=0 TO (N-1)
1010 : IF N<K+2 THEN GOTO 1050
1020 : FOR I=(N-1) TO (K+1) STEP -1
1030 : A(I+1)=(A(I+1)-A(I))/(X(I+1)-X(I-K))
1040 : NEXT I
1050 NEXT K
1060 FOR K=N-1 TO 1 STEP -1
1070 : FOR I=K TO N-1
1080 : A(I)=A(I)-A(I+1)*X(K)
1090 : NEXT I
1100 NEXT K
1200 FOR I=1 TO N
1210 : C(I)=A(N+1-I)
1220 NEXT I
5000 DATA 4
5010 DATA 1,2,3,4
5020 DATA -18,-33,-8,87
```

Algoritmo di Horner su C64

```
100 READ N
110 FOR I=1 TO N
120 : READ C(I)
130 NEXT I
140 READ X
1000 Y=C(1)
1010 FOR J=2 TO N
1020 : Y=Y*X+C(J)
1030 NEXT J
5000 DATA 4
5010 DATA 5,-10,-20,7
5020 DATA 3.5
```

L'algoritmo legge i coefficienti del polinomio ($C(1), C(2), \dots, C(N)$) e il nodo X in cui valutare il polinomio e fornisce la valutazione nella variabile Y .

Bibliografia:

- [1] A. Björck and V. Pereyra. Solution of Vandermonde systems of linear equations. Math. of Computation, 24:893-903, 1970.
- [2] G. Fedele. Vandermonde systems: Theory and Applications, in Computational Mathematics: Theory, Methods & Appl., Nova Science Publishers, Inc.
- [3] N.J.Higham. Accuracy and Stability of Numerical Algorithms. SIAM, 1996.

Esplorando l'Amiga - parte 1

di Leonardo Giordani

Recentemente ho deciso di dedicare del tempo alla retroprogrammazione, ed in particolare di esplorare le architetture di alcuni famosi microcomputer e console degli anni 80 e 90. Sistemi gloriosi come lo ZX Spectrum, il Commodore 64, o il Nintendo Entertainment System, giganti a 16 bit come l'Amiga, i suoi rivali basati sull'MS-DOS e le architetture x86, che infine vinceranno la sfida, e gli immortali Nintendo Super NES (Super Famicom) e SEGA Megadrive (Genesis)

Sono profondamente convinto che studiare le architetture un ottimo metodo per diventare programmatori migliori, anche ora che abbiamo processori a 64 bit, linguaggi di quinta generazione e Internet. I vecchi sistemi hanno molte limitazioni, e per ottenere buoni risultati il programmatore deve conoscere l'hardware e sfruttare ogni singolo bit che questo può fornire.

Questo talvolta non è più vero nei tempi moderni. Da una parte questo è un bene, poiché ci permette di concentrarci sulle decisioni relative al prodotto e di stare ad un livello di astrazione maggiore. Dall'altra era ed è tuttora utile ad un programmatore, lo sarà sempre, dover avere a che fare con un sistema limitato e dover cercare di spremere tutto il possibile da esso.

Questa serie di articoli si occupa del Commodore Amiga. Sono già state spese migliaia di parole sull'Amiga, e non vorrei aggiungere nulla se non "pietra miliare" alle descrizioni di questo sistema. Questo articolo e i prossimi non sono e non vogliono essere una descrizione precisa e completa dell'architettura dell'Amiga. Piuttosto li penso come una serie di "note di laboratorio" per me stesso, che scrivo man mano che mi addentro nella piattaforma. Le condivido nella speranza che possano essere utili ad altri programmatori che vogliono approfondire questi temi.

Questi articoli sono scritti originariamente in inglese sul mio blog thedigitalcatonline.com. Mi occupo personalmente della traduzione ed essendo italiano di origine spero di commettere meno errori in questa versione

che in quella inglese. Ringrazio sentitamente Francesco Fiorentini e lo staff di RetroMagazine innanzitutto per l'ottimo prodotto, di cui personalmente sentivo la mancanza, e in secondo luogo per avermi dato lo spazio per pubblicare questi articoli.

Programmare in Assembly sull'Amiga

Se vogliamo scrivere programmi in Assembly per l'Amiga possiamo lavorare direttamente sul sistema (reale o emulato) o usare un cross-compiler.

Personalmente preferisco lavorare sul mio sistema Linux; per quanto io ami le architetture vecchie devo ammettere che un buon sistema Unix e un editor moderno sono decisamente utili.

La crosscompilazione è un concetto molto semplice. Il compilatore elabora il codice sorgente e, invece che creare codice macchina per l'architettura su cui viene eseguito, lo crea per un'architettura differente. In questo caso l'architettura ospite (host) è Linux/amd64 e l'architettura di destinazione (target) è l'Amiga.

Consiglio di creare una directory dove conservare tutto quanto riguarda la crosscompilazione relativa ad un sistema (in questo caso l'Amiga): compilatore, documentazione, script.

Installare vasm

Su Linux possiamo utilizzare sia il compilatore GCC che vasm. Quest'ultimo usa la stessa sintassi degli assembler tradizionali dell'Amiga, specialmente per quanto riguarda le direttive come `macro` e `include`, e quindi rende molto più facile utilizzare codice pubblicato in libri o riviste degli anni in cui l'Amiga era in commercio.

Per installare vasm potete eseguire il seguente codice:



```
wget -q
"http://sun.hasenbraten.de/vasm
/release/vasm.tar.gz"

tar xvf vasm.tar.gz

rm vasm.tar.gz

cd vasm

make CPU=m68k SYNTAX=mot

cd ..
```

Installare un disassembler

La suite bvcc di Volker Barthelmann contiene un disassembler per le architetture m68k (famiglia Motorola 68000). Potete trovarlo all'indirizzo:

<http://sun.hasenbraten.de/~frank/projects/>

Un disassembler è uno strumento molto utile che può svelare molte cose di come un programma funzioni. Per installarlo potete eseguire il seguente codice:

```
wget -q
"http://sun.hasenbraten.de/~fra
nk/projects/download/vdam68k.ta
r.gz"

tar xvzf vdam68k.tar.gz

cd vda/M68k/

make
```

che creerà l'eseguibile `vda/M68k/ vda68k` che potrete usare per disassemblare programmi Amiga o ROM dump.

Installare l'NDK

Il Native Development Kit (l'SDK dell'Amiga) contiene gli include file che possono essere

molto utili (anche se non generalmente necessari) quando si sviluppa. Sfortunatamente l'NDK è ancora sotto copyright in mano a qualcuno di quelli che stanno cercando di far risorgere l'Amiga. Quest'ultima idea è sicuramente interessante, ma non vedo come bloccare la distribuzione di documentazione relativa allo sviluppo su piattaforme morte e sepolte da 20 anni possa aiutare tali progetti.

Ad ogni modo, una vecchia versione dell'Amiga Developer CD, reperibile all'indirizzo https://archive.org/details/amiga-developer-cd-v1_1, contiene la versione 3.1 dell'NDK, che è più che sufficiente per quello che mi preggio di fare in questa serie di articoli.

Consiglio di rinominare la directory Includes&Libs e di chiamarla Includes_Libs per semplificare le righe di comando del compilatore, visto che & è un carattere speciale in bash.

Helper script e test

Ho creato un piccolo helper script `asm.sh` per semplificare lo sviluppo:

```
#!/bin/bash

BASE="/where/you/put/everything"

"${BASE}/vasm/vasmm68k_mot -
kicklhunks -Fhunkexe -

I${BASE}/NDK_3.1/Include_Libs/i
nclude_i -o ${1/.asm/} -nosym
$1
```

Non dimenticate di eseguire `chmod 775 asm.sh` per renderlo eseguibile. Ora potete eseguire il seguente comando

```
echo -e "loop:\n btst
#6,\$bfe001\n bne loop\n rts\n"
> test.asm && ./asm.sh test.asm
```

che compila un piccolo programma Assembly che rimane in un ciclo fintanto che non premete il pulsante destro del mouse. Se tutto è stato installato correttamente dovreste ottenere il seguente output

```
vasm 1.8c (c) in 2002-2018
Volker Barthelmann
vasm M68k/CPU32/ColdFire cpu
backend 2.3b (c) 2002-2017
Frank Wille
vasm motorola syntax module
3.11c (c) 2002-2018 Frank Wille
vasm hunk format output module
2.9b (c) 2002-2017 Frank Wille
CODE(acrx2): 12 bytes
```

Ed eseguendo il comando `file` dovreste ottenere il seguente risultato

```
$ file test
test: AmigaOS loadseg()ble
executable/binary
```

LEA, PC e l'offset relativo

In Assembly si possono gestire celle di memoria utilizzando sia il loro valore che il loro indirizzo, così come si può fare con variabili e puntatori in C o con concetti simili in altri linguaggi di alto livello. Possiamo anche creare delle etichette (label) che l'assembler convertirà poi in indirizzi, ma tali etichette sono (quasi) sempre relative all'inizio del codice stesso, poiché il codice potrebbe venir caricato ovunque in memoria.

Questo significa che, in generale, tutti gli indirizzi che andremo ad utilizzare quando saltiamo in altre parti del codice devono essere relativi all'istruzione corrente.

Il processore Motorola 68k chiama questo metodo di indirizzamento *Program Counter Indirect with Displacement Mode*. La descrizione che ne dà il Programmer's Reference Manual è

"In this mode, the operand is in memory. The address of the operand is the sum of the address in the program counter (PC) and the sign-extended 16-bit displacement integer in the extension word. The value in the PC is the address of the extension word. (2.2.11, page 13)"

[In questa modalità l'operando è in memoria. L'indirizzo dell'operando è la somma dell'indirizzo nel program counter (PC) e il dislocamento intero a 16 bit con segno nella extension word. Il valore nel PC è l'indirizzo della extension word (2.2.11, pagina 13) (TdA)]

La sintassi Assembly tradizionale per questa modalità di indirizzamento è `(d16,PC)` o `d16(PC)`, dove `d16` è un offset (dislocamento) da 16 bit.

L'istruzione `lea`, per esempio, supporta questa modalità, pertanto possiamo trovare codice come

```
0364: 41fa ffa6 lea
0x30c(pc),a0
```

Qui l'istruzione si trova all'indirizzo `0x0364` e carica nel registro `a0` l'indirizzo effettivo dell'istruzione che si trova a `0x030c`.

Attenzione: il nostro codice inizia sempre all'indirizzo `0x0`, pertanto si potrebbe essere tentati di mettere il valore `0x030c` in `a0` e procedere. Però noi vogliamo caricare nel registro l'indirizzo che l'istruzione avrà durante l'esecuzione del programma, e questa sarà differente da quella corrente. L'offset, però, sarà lo stesso, in quanto il codice non cambia forma quando viene eseguito, e questo è il motivo per cui questa modalità di indirizzamento è utile.

Codifica dell'offset relativo

Potrebbe sorgere una domanda a questo punto. Perché il codice mostra l'indirizzo `0x30` se il dislocamento è relativo?

Proviamo ad analizzare il modo in cui l'istruzione è rappresentata in formato binario, per vedere come il processore la riceve. Il valore `0x41faffa6` in forma binaria è

```
0100 0001 1111 1010 1111 1111 1010 0110
```

E se lo dividiamo secondo lo schema degli opcode del Motorola 68k (Programmer's Reference Manual, sezione 8) otteniamo

```
0100 000 111 111010 1111111110100110
^      ^      ^
lea a0 (d16,PC)
```

Pertanto sappiamo che questa è un'istruzione `lea` verso `a0` che usa il Program Counter with Displacement. L'indirizzo passato come argomento è `111111110100110`, che deve essere interpretato come un "intero a 16 bit con segno" come specificato dal Reference Manual.

Il valore è il complemento a due del decimale `-90` e, siccome il PC punta all'indirizzo stesso (`0x0366`), l'indirizzo risultante sarà `0x0366 - 90`, ovvero `0x030c`.

Il fatto che il PC punti all'indirizzo e non all'istruzione potrebbe sfuggire.

Il manuale dice:

The value in the PC is the address of the extension word. 2.2.11 (2-13)

[Il valore nel PC è l'indirizzo della extension word (2.2.11, pagina 13) (TdA)]

Quindi, mentre il codice Assembly usa il valore corretto per semplicità, l'opcode effettivo contiene il dislocamento reale dalla posizione corrente.

Come aprire una libreria

Quando scriviamo codice in un linguaggio ad alto livello come il C, normalmente facciamo riferimento a funzioni provenienti da una libreria esterna, per poi fornire il codice oggetto della libreria stessa nella riga di comando del compilatore. Il codice della libreria è incluso nel codice del programma o caricato a runtime (shared library), ma in entrambi i casi la chiamata di funzione, a livello del codice macchina, è solamente un salto ad un differente indirizzo di memoria.

Se scriviamo un programma direttamente in linguaggio Assembly le cose non sono differenti e possiamo sempre contare sul linker per includere in modo corretto le librerie esterne. In una piattaforma come l'Amiga, invece, è tradizione accedere alle librerie di sistema in modo diretto, saltando manualmente all'indirizzo corretto, e questo è il motivo per cui spesso libri e tutorial utilizzano "numeri magici".

Quando il sistema operativo dell'Amiga carica in memoria una libreria analizza la sua

struttura e crea la cosiddetta *jump table*. Questa non è altro che un array che elenca tutti gli indirizzi delle funzioni che la libreria esporta.

Questo è un modo molto pratico e semplice per lasciare il sistema operativo libero di caricare la libreria in qualsiasi posizione di memoria.

La principale libreria di sistema, Exec, non è differente, ma viene caricata come parte del processo di bootstrap, e l'indirizzo base di questa libreria viene salvato in memoria all'indirizzo `0x4`. Per usare le funzioni di Exec, quindi, dobbiamo semplicemente eseguire un'istruzione `jsr <address>` (jump to subroutine), dove `<address>` è la posizione in memoria della funzione che vogliamo chiamare. Siccome non sappiamo l'indirizzo assoluto, dato che la libreria è caricata dinamicamente, utilizzeremo la jump table per trovare il dislocamento della funzione rispetto all'indirizzo base della libreria.

Molti programmatori Amiga conoscevano (e conoscono) tali dislocamenti a memoria, il che è perfettamente valido in quanto il sistema operativo dell'Amiga promette di non cambiare tali indirizzi di versione in versione. Quindi, per esempio, la funzione `OpenLibrary` può essere reperita all'indirizzo `-552` rispetto all'indirizzo base della libreria, mentre `CloseLibrary` è all'indirizzo `-434`.

Per chiamare la funzione `OpenLibrary`, pertanto, dobbiamo eseguire il seguente codice:

```
move.l 4.w,a6 ; a6 = base
address of Exec
jsr -552(a6) ; OpenLibrary()
```

La prima istruzione carica il valore contenuto all'indirizzo `0x4` nel registro `a6`. In questo modo il registro contiene l'indirizzo base di Exec. Poi il codice salta alla subroutine il cui indirizzo è `552` byte prima di quell'indirizzo base. Quindi se `a6` contenesse l'indirizzo `0x20000` il codice salterebbe a `0x1fdd8` (`0x20000 - 552`).

La funzione `OpenLibrary`, però, si attende dei parametri, come si può vedere nella pagina di documentazione `exec.library/OpenLibrary`

(http://amigadev.elowar.com/read/ADCD_2.1/Includes_and_Autodocs_3_guide/node022_2.html). Il puntatore al nome della libreria deve essere contenuto nel registro `a1`, mentre la versione minima della libreria va posta in `d0`. Il codice diventa quindi

```
; a6 = base address of Exec
move.l 4.w,a6
; 0 = accept all versions
clr.l d0
; a1 = address of the libname string
move.l #libname,a1
; OpenLibrary()
jsr -552(a6)
libname:
dc.b "somename.library",0
```

Prossimi articoli

Nei prossimi articoli si parlerà più approfonditamente della jump table e delle funzioni di base di Exec. Daremo inoltre un'occhiata ai tipi di dato standard forniti dall'NDK ed a come vengono definite le strutture dati complesse.

Risorse

Motorola M68000 Family Programmer's Reference Manual:

<https://www.nxp.com/docs/en/reference-manual/M68000PRM.pdf>

AmigaOS Developer Docs:

<http://amigadev.elowar.com>

*** 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/2018/05/28/exploring-the-amiga-1/>

Enjoy it!

Demo in Basic per C64 - L'alieno!

di Alessandro Paloni

La Demo proposta in questo numero, particolarmente elementare nella veste grafica, utilizza esclusivamente linguaggio Basic, senza alcun riferimento a codice Assembly, Interrupts, ecc, tuttavia è in grado di offrire una panoramica generale sull'utilizzo degli Sprite, su come sia possibile compiere animazioni, seppur molto semplici, richiamare subroutine per effetti sonori, disegnare oggetti sullo schermo, persino raccogliere un oggetto senza utilizzare codice per la gestione delle collisioni, ed altro ancora.

Vi è infine un piccolo omaggio ad uno dei giochi considerati tra i più brutti della storia dei videogames, "ET", sviluppato da Howard Scott Warshaw e rilasciato nell'anno 1982 per Atari 2600.

Il poco tempo a disposizione concesso al programmatore, appena 5 settimane, fu probabilmente la causa di qualche problemino presente nel gioco, tuttavia, sia per alcuni contenuti presenti (easter egg, disposizione randomica degli oggetti da trovare) che per gli Sprite che caratterizzavano i personaggi, quest'avventura 8 bit dovrebbe essere, a mio parere, rivalutata.

Ma veniamo a noi, alla nostra Demo in Basic e all'immane presentazione che dovrebbe accompagnare ogni rilascio su C64!

In un angolo remoto dell'universo, un'astronave aliena mentre sta sorvolando a bassa quota una zona disabitata, è costretta, a causa di un cortocircuito, ad un atterraggio di emergenza!! Tutto farebbe supporre che per l'equipaggio non ci sia possibilità di sopravvivenza...

Nota del redattore – Ci scusiamo per il carattere minuscolo utilizzato nel listato dovuto ad esigenze di impaginazione, ma essendo possibile copiarlo ed incollarlo e/o utilizzare lo zoom dell'Adobe Reader, riteniamo non essere un problema insormontabile. :-P

```

5 REM DEMO BASIC C64 di Alessandro Paloni
8 RI=10: REM IMPOSTO VARIABILE CHE USERO' PER EFFETTI SONORI
10 PRINTCHR$(147): REM PULISCO LO SCHERMO
20 POKE 53280,16: POKE 53281,16: REM SETTO BORDO E SCHERMO A NERO
29 REM LE ISTRUZIONI DI SEGUITO CREANO LO SCENARIO
30 POKE 1024,46: POKE 55296,1:POKE 1024+35,81: POKE 55296+35,1
35 POKE 1024+110,46:POKE 55296+110,1: POKE 1024+371,81:POKE 55296+371,1
40 POKE 1024+425,46:POKE 55296+425,1: POKE 1024+550,46:POKE 55296+550,1
45 POKE 1024+605,46:POKE 55296+605,1: POKE 1024+790,46:POKE 55296+790,1
46 FOR I=0TO39 : POKE1984+I,68: POKE 56256+I,9: NEXT : REM DISEGNA LINEA DI TERRA
51 IF FLAG=1 GOTO 110: REM SE FLAG=1 SALTO LA STAMPA A VIDEO DELLA SCRITTA DI SEGUITO
60 PRINT " ": PRINT " ":PRINT " ": PRINT " ":PRINTCHR$(30) " da qualche parte,
nell'universo ..."
71 GETAS : REM ASPETTO CHE VENGA PREMUTO SPAZIO
75 IF AS= " " THEN GOTO 71: REM LOOP
80 PRINTCHR$(147) : REM PULISCE LO SCHERMO
100 FLAG=1:PRINTCHR$(147):GOTO 20:REM UN FLAG PER NON RISTAMPARE LA SCRITTA A VIDEO
110 V=53248: REM INIZIO DEMO - IMPOSTO I REGISTRI VIC
111 POKEV+21,4 : REM POKE V+21,4 SETTA LO SPRITE 2 DEL VIC
112 POKE 2042,13 : REM PUNTATORE AL BLOCCO DATI RELATIVO ALLO SPRITE
116 POKE 53289,5 : REM SETTO IL COLORE VERDE DELL'UFO
120 FORN=0TO62 : READ Q:POKE 832+N,Q:NEXT: REM LEGGO I DATI PER DISEGNARE LO SPRITE
130 FOR X=0TO 150 STEP 2 : REM LOOP PER MUOVERE L'UFO VERSO SINISTRA
131 GOSUB 1010: REM CHIAMO SUBROUTINE EFFETTI SONORI
140 POKE V+4,X:REM VALORE POSIZIONE ASSE X
150 POKE V+5,60:REM VALORE POSIZIONE ASSE Y
151 REM POKE V+6,X: REM POKE V+7,200:POKE V+8,X:POKEV+9,100
160 NEXT X : REM INCREMENTO X
170 POKEV+16,0: GOTO 2410 : REM SALTO AL MOVIMENTO VERTICALE
171 REM EFFETTO SONORO ASTRONAVE
1010 POKE 54296,04: REM VOLUME ON - SETTATO A 4 SU UNA SCALA 0-15
1020 POKE 54278,240: REM SETTO SUSTAIN
1030 POKE 54276,33: REM NOTA ON SU CONTROLLO REGISTRO VOICE 1 DEL SID
1040 FORN=1 TO10:REM GENERO UN SUONO VARIABILE CON I VALORI DI N
1050 POKE 54273,15+ABS(50-N)
1060 NEXTN
1090 POKE 54276,0: REM NOTA OFF
1100 POKE 54296,0: REM VOLUME OFF
1110 RETURN: REM RITORNO AL PUNTO DI CHIAMATA
1140 REM SPRITE ASTRONAVE
2000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2002 DATA 0,255,0,3,0,192,4,0,32,12,0,48
2003 DATA 12,0,48,63,255,252,64,0,2,140,60,49
2004 DATA 64,0,2,48,0,12,15,255,240,0,0,0
2005 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2006 DATA 0,0,0
2007 REM SPRITE ET
2008 DATA 255,255,240,255,255,255,195,255,255,195,255,255
2009 DATA 255,255,255,255,255,255,252,0,31,252,0,31
2010 DATA 0,3,255,0,7,255,0,15,255,255,255,255
2011 DATA 255,255,255,193,255,255,1,255,255,1,255,255
2012 DATA 1,255,255,1,198,31,3,192,31,7,192,31
2013 DATA 31,192,127
2410 POKE 53277,PEEK(53277)OR4:REM RADDOPPIA DIMENSIONI DELLO SPRITE 2
2415 GOSUB 2620 : REM CHIAMO LA SUBROUTINE DEL SUONO ALLARME ROSSO
2420 FOR X=60 TO200: POKE V+4,150:POKE V+5,X: NEXT : REM INIZIA LA DISCESA
2421 FOR X=200TO229:POKE V+4,150:POKE V+5,X: REM RALLENTA LA DISCESA DELL'ASTRONAVE
2422 FOR I=1TO100: NEXT : NEXT
2431 REM ANIMAZIONE ET
2441 POKEV+21,0 : REM DISATTIVO LO SPRITE
2443 POKE 53277,PEEK(53277)AND 0:REM RIPRISTINA DIMENSIONI DELLO SPRITE 2 E SETTO VERDE
2444 FOR I=1TO100 :POKE 53281,5: POKE 53281,1: NEXT: POKE 53281,16:REM ESPLOSIONE
2460 FORN=0TO62 : READ Q:POKE 832+N,Q:NEXT: REM LEGGE I DATI DI ET (SEMPRE SU SPRITE 2)
2461 POKE 56256+15,16:POKE 56256+16,16:POKE 56256+17,16:POKE 56256+18,16:POKE
56256+19,16
2465 POKEV+21,4: POKE 1948,83: POKE 56220,2: REM ATTIVO SPRITE E DISEGNO CUORE
2470 FOR X=150TO 0 STEP -3 : REM MUOVO ET VERSO SINISTRA
2480 POKE V+4,X: POKE V+5,220:POKE V+6,X:
2482 IF X=45 THEN RI=1: GOSUB 2620 :POKE 56220,16:REM POKE LOC. 56220 CANCELLA IL CUORE
2490 FOR I=1TO100: NEXT: GOSUB 2700: REM CHIAMO SUBROUTINE EFFETTO PASSI
2500 NEXT
2610 GOTO 2610 REM LOOP DI FINE
2620 REM EFFETTO SONORO ALLARME ROSSO
2621 POKE54278,240:POKE54277,0
2622 POKE54275,8:POKE54276,65
2623 POKE54296,4
2624 FORL=1TORI : IF RI<> 1 THEN POKE 53289,L : REM CAMBIA COLORE ALL'UFO
2625 FORM=1TO50 STEP2
2626 POKE54273,M
2627 FORN=1TO10
2628 NEXTN:NEXTM
2630 POKE54273,0
2633 NEXTL
2634 POKE54296,0: POKE 53289,5: REM VOLUME A ZERO E SETTO A VERDE LO SPRITE
2644 RETURN : REM RITORNO AL PUNTO DI CHIAMATA
2700 REM EFFETTO SONORO PASSI ET
2710 POKE 54277,6:POKE54296,08
2720 POKE 54272,15: POKE54273,2
2770 POKE 54276,129
2775 POKE 54296,0
2780 RETURN

```

JoyTest64

un piccolo tool in assembly C64 per testare i vostri joystick

di David La Monaca (Cercamon)

A chi, come il sottoscritto, coltiva una passione *moderata* per il retrocomputing, di tanto in tanto può capitare di mettersi alla ricerca di qualche pezzo hardware del passato oppure ritrovarsi fra le mani un vecchio home computer con accessori da testare. Aver a che fare con macchine di oltre 30 anni fa è sempre un azzardo: ogni piattaforma ed ogni modello ha i suoi punti deboli. E spesso, quando si acquistano vecchi computer da privati tramite eBay o su altri siti di usato, in realtà si acquistano piccoli o grandi lotti che includono diversi accessori fra cui alimentatori, monitor, mouse, dischetti, manuali e... joystick.

Nel corso di circa 5 anni di attività nel retrocomputing mi sono arrivati molti joystick, di tutte le forme, modelli e marche, dai più ricercati e preziosi per le loro indiscusse qualità funzionali fino ai più economici. Ovviamente anche le condizioni in cui sono finiti nelle mie mani sono state le più disparate: da quelli "in perfetto stato di conservazione" a quelli completamente distrutti. Nel corso di questi ultimi anni ho continuato a raccogliere tutti i joystick di tipo "Atari" (9-pin D socket) in una grande scatola, ripromettendomi prima o poi di testarli a dovere per effettuare una cernita in base alla qualità, allo stato di funzionamento effettivo e anche per capire quali valeva la pena di riparare, quali rivendere e quali gettare definitivamente (previa consultazione con alcuni amici più esperti di me) nella spazzatura. Quando è arrivato il momento di effettuare i controlli, sulle prime pensavo di usare un C64 ed un paio di giochi, collegando i diversi joystick. E così ho cominciato, ma poi, grazie ad uno spunto offertomi da un post su FB del mio caro amico **Emiliano Buttarelli** (*MetalRedStar*, il suo post riguardava una sua ottima utility in BASIC per testare joystick), mi sono detto che potevo costruirmi un piccolo tool in assembly 64. Naturalmente scrivere un programmino in BASIC per testare le porte joystick del C64 sarebbe più che sufficiente e soprattutto rapido, ma ho colto l'occasione per rispolverare un po' delle mie conoscenze di assembly e creare un programma di test con un minimo di interfaccia grafica per rendere almeno un po' divertente la sequela noiosa di collegare e controllare più di 30 joystick!

Ho accennato alla possibilità di usare il BASIC per testare le porte joystick del C64. In effetti, su questa macchina basta controllare il valore di due precise locazioni di memoria (56320 e 56321) per conoscere lo stato della leva e del pulsante fire dei joystick collegati alle due porte a 9 pin del 64. Un esempio veloce?

```
10 J1=PEEK(56321):J2=PEEK(56320)
20 FOR I=4 TO 0 STEP -1
30 ?(J2 AND 2^I)>0;(J1 AND 2^I)>0;
40 NEXT
50 ?
60 GOTO 10
```

Che ne dite di uno ancora più spartano, magari in formato *one-liner*?

```
1 ?PEEK(56320);PEEK(56321);:GOTO1
```

Molto semplice, in effetti. E tutto sommato funzionale allo scopo. L'unica pecca di queste versioni è forse la scarsa leggibilità dei risultati (numerici) stampati a video e, in parte, la velocità d'esecuzione dei controlli. Un programma scritto in assembly deve

(\$10) per il FIRE. Questi valori corrispondono ai primi 5 bit delle locazioni relative alle porte joystick, quando i diversi bit assumono il valore 1 mentre i restanti 4 sono a zero. Quindi per ciascuna delle due locazioni vale la rappresentazione degli 8 bit (da 0 a 7):

```
□□□□□□□□
7 6 5 4 3 2 1 0
```

I valori che ci interessa controllare nelle locazioni \$DC00 e \$DC01 sono quindi:

```
0000 0001 = $01 = 1 -> UP
0000 0010 = $02 = 2 -> DOWN
0000 0100 = $04 = 4 -> LEFT
0000 1000 = $08 = 8 -> RIGHT
0001 0000 = $10 = 16 -> FIRE
```

Per effettuare il controllo in assembly si potrebbe eseguire la sequenza: caricare il valore della locazione nell'accumulatore (il registro di lavoro della CPU), confrontare il valore con tutti quelli che corrispondono alle posizioni del joystick più il pulsante di FIRE e segnalare in qualche modo all'utente quale

test	LDA \$DC00	; valore di \$DC00 (oppure \$DC01) in A
	CMP #\$01	; confronta il valore di A con il valore 1 (UP)
	BEQ joyup	; se uguale il joy è UP, prosegui da "joyup"
	CMP #\$02	; controlla se A = 2
	BEQ joydown	; se uguale prosegui da label "joydown"
	...	
	CMP #\$10	; controlla se A = 16 (FIRE)
	BEQ joyfire	; se uguale prosegui da label "joyfire"
	JMP test	; ricomincia il ciclo dei controlli
joyup	STA \$D020	; joystick UP, cambia colore del bordo a 1 = bianco
	...	
joydown	STA \$D020	; joystick DOWN, cambia colore del bordo a 2 = rosso
	...	

Listato 1 – Test semplice del contenuto di una locazione della porta joystick

necessariamente basarsi sullo stesso principio: controllare ripetutamente il valore delle locazioni corrispondenti alle porte joystick e visualizzare sullo schermo eventuali cambiamenti di stato per segnalare i 5 eventi possibili (UP, DOWN, LEFT, RIGHT e pulsante FIRE) o le loro combinazioni.

Il valore assunto dalle locazioni 56320 (porta 2, \$DC00 in esadecimale) e 56321 (porta 1, \$DC01) ci dirà quale o quali eventi sono in corso, cioè che cosa sta facendo l'utente con il joystick. I valori corrispondenti alle diverse posizioni della leva e del pulsante di fuoco sono 1 (\$01) per UP, 2 (\$02) per DOWN, 4 (\$04) per LEFT, 8 (\$08) per RIGHT e infine 16

condizione si è verificata. Qualcosa di simile a quanto mostrato nel listato 1.

Tutto questo non è però molto efficiente: servono molti confronti (CMP) e salti condizionati (BEQ) che non rappresentano di certo la migliore delle soluzioni. Inoltre c'è il problema che nelle locazioni in esame gli ultimi 3 bit (quelli numerati da 5 a 7) potrebbero anche assumere valori diversi da 000 (tutti e tre i bit a zero). Questo perché nei C64 le medesime locazioni vengono utilizzate anche da altri processi, ad es. la pressione dei tasti e quindi i nostri confronti CMP non funzionerebbero. Ci sono altre maniere di controllare i bit da 0 a 4 nelle locazioni \$DC00

e \$DC01, ad esempio effettuando operazioni booleane di tipo AND usando delle opportune maschere di bit. Altra possibilità (simile all'AND logico) è quella di utilizzare l'istruzione **BIT** disponibile nelle CPU 6502/6510, ad es.:

```
LDA #%00010000 ; test FIRE (16)
BIT $DC00       ; operazione di AND
                ; col val. in $DC00
BEQ jfire       ; se corrisponde
                ; vai a label jfire
```

C'è però un modo più semplice e veloce per controllare i primi 5 bit delle locazioni ed è quello di utilizzare una delle operazioni sui bit che l'assembly mette a disposizione. In particolare, ci è molto d'aiuto l'istruzione LSR (Logical Shift Right), "spostamento logico verso destra", ossia la traslazione verso destra di tutti gli 8 bit contenuti nell'accumulatore. In questa operazione gli 8 bit del byte che si trova in A vengono ruotati come mostrato in figura ed il primo bit (il bit 0) viene trasferito nel bit di carry del registro di stato mentre il bit 7 (quello più a sinistra) viene riempito con uno 0 (cfr. Figura 1).

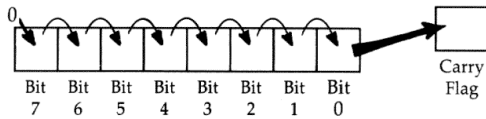


Figura 1 – L'istruzione LSR

Dal momento che ad ogni esecuzione del comando LSR, il valore del bit 0 finisce nel bit di Carry, basta verificare il valore di quest'ultimo per capire quale evento si sia verificato sulle porte joystick ad ogni dato momento. In pratica, ripetendo per 5 volte l'istruzione LSR e verificando ogni volta se il bit di carry è a 1 (uno) oppure a 0 (zero) con l'istruzione **BCC (Branch Carry Clear)**, possiamo controllare il contenuto dei 5 bit meno significativi (cioè quelli numerati da 0 a 4) dell'accumulatore e quindi capire lo stato del joystick in base al confronto che ha dato esito positivo (bit di carry = 1).

Ma cos'è il bit di Carry? La CPU 6510 del C64 controlla un particolare byte denominato **SR (Status Register)**, di cui 7 bit su 8, detti anche *flag*, vengono utilizzati per tenere traccia di tutte le operazioni in corso. Il bit 0 del registro di stato è detto bit di Carry (o di "riporto") e serve per segnalare un riporto nelle operazioni aritmetiche (come ADC e SBC), ma viene anche utilizzato nelle operazioni di spostamento e rotazione dei bit (LSR, ASL, ROL, ROR) e nelle operazioni di confronto (CMP, CPX, CPY). Il flag di Carry assume ovviamente il valore 1 (uno) oppure 0 (zero) a seconda dell'esito di un'operazione appena compiuta dal processore. Lo stesso vale per

gli altri bit del registro di stato che segnalano gli altri eventi possibili (Negative, oVerflow, Break command, Decimal, Zero flag – vedi anche Figura 2).

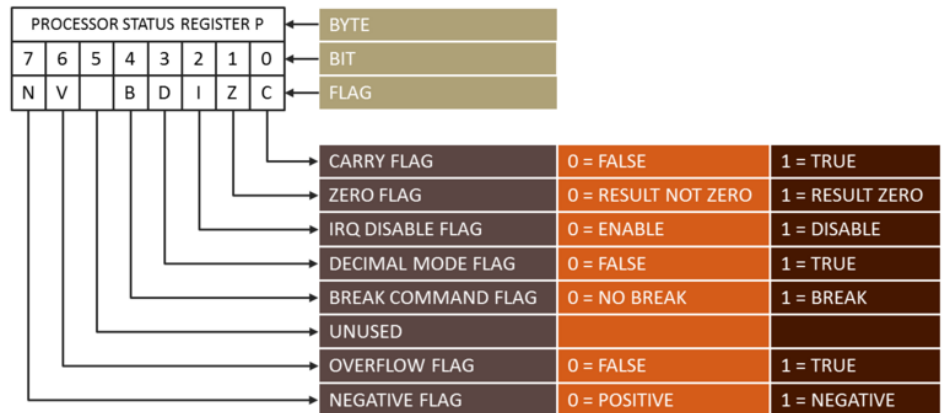


Figura 2 – Il registro di stato del 6510

Utilizzando LSR per traslare i bit dell'accumulatore e BCC per controllare che il bit di carry sia a zero (Clear) oppure a 1 (Set), potremo rapidamente verificare tutti e 5 gli eventi delle porte joystick. In sintesi, il nostro ciclo di controllo delle locazioni relative alle porte joystick del C64 diventa qualcosa di simile al seguente:

```
test      LDA $DC00
          LSR
          BCC noUP
joyUP     STA $D020

noUP      LSR
          BCC noDN
joyDN     STA $D020

noDN      LSR
          BCC noLT
joyLT     STA $D020

noLT      LSR
          BCC noRT
joyRT     STA $D020

noRT      LSR
          BCC noFR
joyFR     STA $D020

noFR      JMP test
```

Sicuramente più snello ed efficiente. Bene, ora che abbiamo approntato il nucleo del ciclo principale di controllo del nostro programma, possiamo dotarlo di un'interfaccia grafica semplice ed efficace per comunicare visivamente all'utente sullo schermo gli eventi delle porte joystick.

Utilizzeremo una semplice schermata di testo con un carattere grafico tratto dalla tabella PETSCII, in particolare un "pallino" (codice PETSCII 81 = #\$51). Posizioneremo i "pallini" di colore grigio a forma di croce e

predisporremo così due diagrammi per rappresentare i due joystick. In base agli eventi rilevati (UP, DOWN, LEFT, RIGHT e FIRE), il programma colorerà semplicemente

il pallino di rosso chiaro (cfr. figura 3).

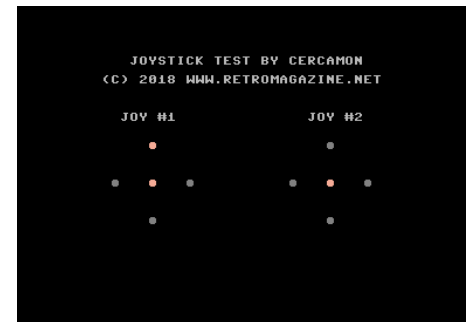


Figura 3 – JoyTest64 in azione

Il listato completo di JoyTest, pubblicato nelle pagine seguenti (Listato 2), è ampiamente commentato. Tenendo conto di quanto detto finora a proposito del sistema di test delle porte joystick, con poche conoscenze di base dell'assembly e della mappa di memoria del C64, non è impossibile seguire il flusso del programma, dalla struttura molto lineare.

Un **archivio zip** col sorgente del programma (**joytest.asm**) e l'eseguibile (**joytest.prg**) pronto per l'uso è disponibile per il download all'URL:

<http://www.retro magazine.net/download/joytest64.zip>

Per informazioni specifiche sul programma o chiarimenti sul suo listato sorgente, non esitate a contattarci scrivendo un messaggio di posta elettronica a:

retromagazine.redazione@gmail.com

Listato 2 – sorgente JOYTEST64 – parte 1/3

```
; luogo, data:      Lugano, 10.09.2017
; autore:          Cercamon
; sorgente:        joytest.asm
; compilatore/IDE: CBM Prg Studio (Windows)
; descrizione:     Controllo porte joystick del C64

; imposta label per alcune locazioni di memoria/schermo
JOYPORT2 = $dc00 ; indirizzo memoria joystick porta 2
JOYPORT1 = $dc01 ; indirizzo memoria joystick porta 1

J1UP_S   = $059a ; locazione mem. schermo per JOY 1 UP
J1UP_C   = $d99a ; locazione mem. colore per JOY 1 UP
J1DOWN_S = $06da ; loc. schermo JOY 1 DOWN
J1DOWN_C = $dada ; loc. colore JOY 1 DOWN
J1LEFT_S = $0636 ; loc. schermo JOY 1 LEFT
J1LEFT_C = $da36 ; loc. colore JOY 1 LEFT
J1RIGHT_S= $063e ; loc. schermo JOY 1 RIGHT
J1RIGHT_C= $da3e ; loc. colore JOY 1 RIGHT
J1FIRE_S = $063a ; loc. schermo JOY 1 FIRE
J1FIRE_C = $da3a ; loc. colore JOY 1 FIRE

J2UP_S   = $05ad ; locazione mem. schermo per JOY 2 UP
J2UP_C   = $d9ad ; locazione mem. colore per JOY 2 UP
J2DOWN_S = $06ed ; loc. schermo JOY 2 DOWN
J2DOWN_C = $daed ; loc. colore JOY 2 DOWN
J2LEFT_S = $0649 ; loc. schermo JOY 2 LEFT
J2LEFT_C = $da49 ; loc. colore JOY 2 LEFT
J2RIGHT_S= $0651 ; loc. schermo JOY 2 RIGHT
J2RIGHT_C= $da51 ; loc. colore JOY 2 RIGHT
J2FIRE_S = $064d ; loc. schermo JOY 2 FIRE
J2FIRE_C = $da4d ; loc. colore JOY 2 FIRE

; a partire da locazione 2049 (inizio area BASIC)
; inserisce linea BASIC per l'autorun "10 SYS 2064"
*=$0801
BYTE $0b,$08,$0a,$00,$9e,$32,$30,$36,$34,$00,$00,$00

; inizio codice
*=$0810 ; locazione di inizio codice 2064
ldy #$00 ; colore nero
sty $d020 ; per il bordo
sty $d021 ; e per lo sfondo

clrloop lda #$20 ; ciclo pulizia schermo, $20=32
sta $0400,y ; $0400=1024 inizio mem. Schermo
sta $0500,y
sta $0600,y
sta $0700,y
lda #$0c ; colore grigio intermedio
sta $d800,y ; $d800 inizio mem. Colore
sta $d900,y
sta $da00,y
sta $db00,y
iny ; registro y usato come contatore
bne clrloop

sei ; disabilita gli interrupt

; disegna schermo e stampa etichette e simboli grafici
prints1 ldx #$00 ; stampa primo titolo (string1)
s1 lda string1,x
cmp #$00
beq prints2
sta $0430,x ; pos. schermo titolo 1 ($0430)
inx
jmp s1

prints2 ldx #$00 ; stampa secondo titolo (string2)
s2 lda string2,x
cmp #$00
beq printl1
sta $047d,x ; pos. schermo titolo 2 ($047d)
inx
jmp s2
```

Listato 2 – sorgente JOYTEST64 – parte 2/3

```
printl1 ldx #$00 ; stampa la prima label (label1)
l1 lda label1,x
cmp #$00
beq printl2
sta $051F,x ; pos. schermo label1 ($051F)
inx
jmp l1

printl2 ldx #$00 ; stampa seconda label (label2)
l2 lda label2,x
cmp #$00
beq printgraph
sta $0533,x ; pos. schermo label 2 ($0533)
inx
jmp l2

printgraph lda #$51 ; stampa carattere 'pallino'
; (codice PETSCII 81=#$51)
sta J1UP_S ; J1 UP
sta J1LEFT_S ; J1 LEFT
sta J1FIRE_S ; J1 FIRE
sta J1RIGHT_S ; J1 RIGHT
sta J1DOWN_S ; J1 DOWN
sta J2UP_S ; J2 UP
sta J2LEFT_S ; J2 LEFT
sta J2FIRE_S ; J2 FIRE
sta J2RIGHT_S ; J2 RIGHT
sta J2DOWN_S ; J2 DOWN
ldx #$0b ; colore grigio scuro in X
ldy #$0a ; colore rosso chiaro in Y

; controllo porta joystick #2
testloop lda JOYPORT2 ; preleva valore porta
testUP lsr ; effettua shift a destra
; di A, bit 0 nel carry
bcc noUP ; test UP ($01), se bit di
; carry=0 usa colore normale
stx J2UP_C ; bit carry=1, leva su UP
jmp testDN ; vai test successivo (DOWN)
noUP sty J2UP_C ; no UP, pallino colore grigio

testDN lsr ; nuovo shift a destra,
; bit 1 di A nel carry
bcc noDN ; test DOWN ($02)
stx J2DOWN_C ; bit di carry=0, leva DOWN
jmp testLT ; test successivo (LEFT)
noDN sty J2DOWN_C ; no DOWN, pallino normale

testLT lsr ; altro shift, bit 2 nel carry
bcc noLT ; test LEFT ($04)
stx J2LEFT_C ; ecc.
jmp testRT
noLT sty J2LEFT_C

testRT lsr ; altro shift, bit 3 nel carry
bcc noRT ; test RIGHT ($08)
stx J2RIGHT_C
jmp testFB
noRT sty J2RIGHT_C

testFB lsr ; shift, bit 4 nel carry
bcc noFB ; test FIRE ($10=16)
stx J2FIRE_C
jmp testport1 ; salta a test porta 1
noFB sty J2FIRE_C

; controllo porta joystick #1
; stessa sequenza di controlli eseguiti per la porta #2
testport1 lda JOYPORT1
J1testUP lsr
bcc J1NOUP
stx J1UP_C
jmp J1testDN
J1NOUP sty J1UP_C
```

Listato 2 – sorgente JOYTEST64 – parte 3/3

```

J1testDN  lsr
           bcc J1NODN
           stx J1DOWN_C
           jmp J1testLT
J1NODN    sty J1DOWN_C

J1testLT  lsr
           bcc J1NOLT
           stx J1LEFT_C
           jmp J1testRT
J1NOLT    sty J1LEFT_C

J1testRT  lsr
           bcc J1NORT
           stx J1RIGHT_C
           jmp J1testFB
J1NORT    sty J1RIGHT_C

J1testFB  lsr
           bcc J1NOFB
           stx J1FIRE_C
           jmp goback
J1NOFB    sty J1FIRE_C

goback    jmp testloop    ; ricomincia dall'inizio

; titoli e label da stampare sullo schermo
; tutte le stringhe sono terminate da $00

           ; "JOYSTICK TEST BY CERCAMON"
string1    BYTE $0A, $0F, $19, $13, $14, $09, $03, $0B,
$20, $14, $05, $13, $14, $20, $02, $19, $20, $03, $05,
$12, $03, $01, $0D, $0F, $0E, $00

           ; "(C) 2018 WWW.RETROMAGAZINE.NET"
string2    BYTE $28, $03, $29, $20, $32, $30, $31, $38,
$20, $17, $17, $17, $2E, $12, $05, $14, $12, $0F, $0D,
$01, $07, $01, $1A, $09, $0E, $05, $2E, $0E, $05, $14,
$00

           ; "JOY #1"
label1     BYTE $0A, $0F, $19, $20, $23, $31, $00

           ; "JOY #2"
label2     BYTE $0A, $0F, $19, $20, $23, $32, $00

; fine codice
;
; Un ringraziamento speciale a Hermit, Ice00, MP Soft
; e Salvo Bee per la loro gentile revisione

```



Console 8bit: GAMEBOY

di Starfox Mulder

Quando meno te l'aspetti ecco che arriva... l'estate! Spiagge, turiste tedesche, calippo, partite al calcio balilla. Ok, lo so che mi state guardando con cipiglio omicida, chiusi nei vostri uffici di Milano o Roma, ma io abito al mare quindi son fortunello e ste cose me le godo tra un turno e l'altro di lavoro. Va da se che andare in spiaggia non escluda la presenza dei nostri amati apparecchi elettronici, peccato che nel mio portapacchi della bicicletta mal si sistema una tv a tubo catodico con annesso Atari 7800. Rimandate quindi all'autunno le presentazioni delle console di terza generazione, facciamo un salto netto e passiamo alla prima console di quarta. Classe 1989, il GameBoy fu la prima console portatile della stor...oppure no?!

Quando si parla di portatilità si dovrebbero intavolare tante premesse ma due sono fondamentali: escludiamo gli apparecchi elettronici che non permettono di sostituire la cartuccia di gioco al suo interno (quindi niente Game & Watch, classe 1980) ed escludiamo prodotti sperimentali con lo schermo integrato nella cartuccia (quindi niente MicroVision, classe 1979). Postulate queste regole allora sì, il Nintendo GameBoy fu la prima console portatile, capace di anticipare di pochi mesi il Lynx Atari e di un annetto buono il Game Gear Sega.

La forza del Gameboy però non si limitò alla rapidità nell'uscire per prima ma seppe imporsi grazie ai suoi brand in una maniera che nessun'altra compagnia, da lì fino ad oggi, potesse davvero competere con la casa di Kyoto.

Lo so, non si dovrebbe mai anticipare il finale di una storia, ma qualsiasi sia il vostro anno di nascita credo sappiate come me che Nintendo non ha mai avuto reali rivali in ambito handheld.

Pensare che il primo GameBoy, a fronte di un prezzo molto più contenuto delle avversarie ed una durata delle batterie davvero altissima (15 ore effettive contro le circa 5 delle altre, che per altro richiedevano l'aggiunta di ulteriori due batterie stilo), deficitava in tutto il resto. Dove gli avversari scelsero la retroilluminazione ed i giochi a colori, Nintendo puntò allo schermo verde con toni di grigio e nessuna illuminazione interna. Quanto il tutto fece la gioia degli attuali oculisti non lo sapremo mai ma di fatto al me stesso bambino non interessava granchè,

dato che l'attrattiva era da cercarsi in ben altro: i giochi.

Nintendo era la casa di Zelda, Super Mario, Metroid, Kirby ed un'infinità di saghe già all'epoca leggendarie che su GameBoy diedero il loro meglio. Data la capacità immediata di saper vendere, i giochi della piccola portatile li si poteva trovare nel negozietto in spiaggia così come nell'emporio della tal comunità montana: avevano invaso il mondo.

Mai come allora fu chiaro quanto la disponibilità creava l'interesse. La console uscì in bundle con una delle migliori versioni di Tetris mai programmate, il cavo link per collegare due apparecchi per le partite in multiplayer e le ovvie istruzioni di gioco. Ben presto però i titoli che andarono a far parte del roster della console crebbero esponenzialmente fino ad arrivare ad oltre mille giochi (per il solo GameBoy originale) tutti compatibili a prescindere dalla regione di vendita. Già, perché la portatilità permetteva di bypassare problemi di hertz o corrente e per tanto ogni console era region-free sin da subito e senza necessità di modifiche.

Ma che cuore batteva all'interno del piccolo scrigno delle meraviglie? Come il nome della rubrica vi ricorda, stiamo parlando di un 8bit Sharp derivato dallo Zilog Z80 con 8kb di S-Ram interna e la possibilità di leggere cartucce fino ad 8Mb. Già in quel periodo le console a 16 bit non erano più una chimera da un pezzo, eppure concentrarsi sul "noto" si rivelò una scelta molto valida, soprattutto data ormai la competenza con cui lo si poteva sfruttare al meglio. "Dalle rape non si ricava sangue" dicevano gli antichi, ma dal GameBoy si ricavarono capolavori impensabili. Dopo una partenza timida a colpi di Alleyway (simil Arkanoid) e Tennis, si arrivarono a vedere interi mondi resi in portatilità, come quelli di Link's Awakening, fino ad arrivare alla potenza grafica del Donkey Kong Land che per alcuni risultò ben più incredibile del Country su Super Nintendo.

Il GameBoy aprì davvero la pista alle console portatili che da allora si susseguirono senza sosta fino ai giorni nostri ed anche l'ultima nata in casa Nintendo (la Switch) ebbe in lui l'illustre antenato.

Ci si vede quindi in spiaggia amici, con la vostra copia di RetroMagazine sotto braccio e il cavo link per sfidarci ad un doppio; ovviamente su GameBoy.

Alla prossima console!

CARATTERISTICHE TECNICHE

Produttore	Nintendo
Tipo	Console Portatile
Generazione	Quarta
In vendita	Novembre 1989
Dismissione	Marzo 2003
Supporto	Cartuccia
Unità' vendute	118,69 milioni

FONTE:
[HTTPS://EN.WIKIPEDIA.ORG/WIKI/GAME_BOY](https://en.wikipedia.org/wiki/Game_Boy)

GAME BOY



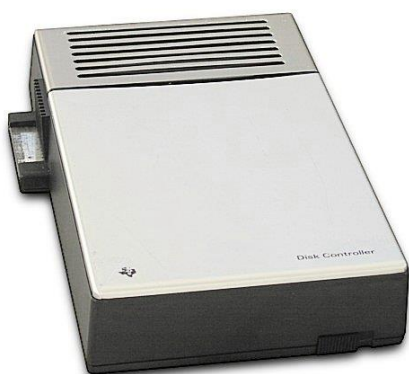
ABC della merenda... TI99/4A - parte 3

... le Periferiche storiche create da terze parti.

di Ermanno Betori

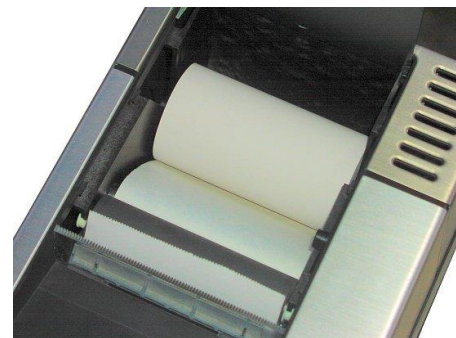
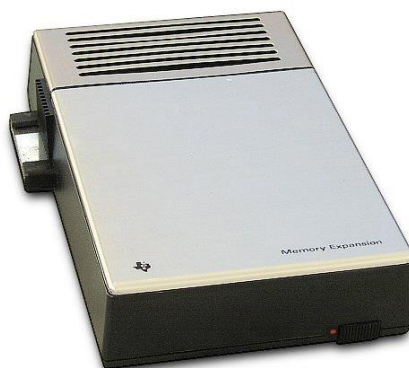
Dopo aver presentato nello scorso numero le periferiche principali create da mamma Texas, ci accingiamo ad esplorare il mondo delle periferiche nella loro complessità! Iniziamo con una rapida carrellata per descrivere quelle create per il TI99/4. Le periferiche sidecar.

Disk Drive Controller PHP1800



Qui abbiamo la scheda con l'interfaccia RS-232, notare la doppia porta seriale posteriore.

32K Memory Expansion PHP 2200



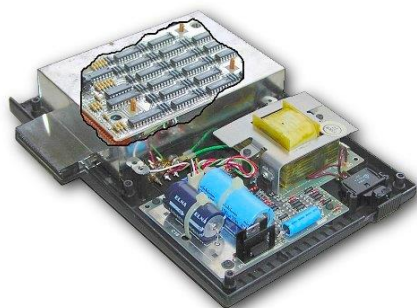
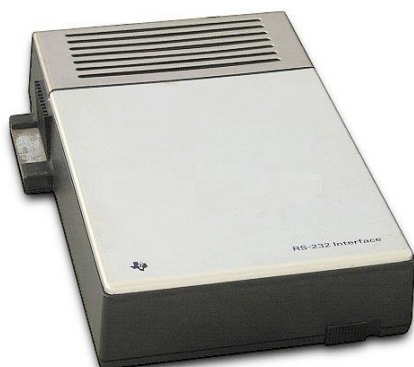
La stampante termica a stato solido di Texas Instruments era una stampante termica a 32 caratteri la cui stampa ovviamente corrispondeva all'output dello schermo. Oggi sono diventate rare ed è difficile averne una in buono stato. La sua velocità di stampa è di circa 30 caratteri al secondo ed è molto silenziosa.

TI Telephone Coupler (Modem) PHP1600



L'immagine sopra mostra la vista frontale e laterale del disk controller che può gestire fino a 3 floppy disk drives, equivalente a quello inserito come scheda nel Peripheral Box.

RS-232 Interface PHP1700



Nella figura che mostra l'esploso della espansione di memoria si nota il circuito di alimentazione che è presente in ogni modulo sidecar.

TI Solid State Thermal Printer PHP1900



Questo modem è uno dei primi tipi di modem commerciali, infatti era denominato accoppiatore acustico e funzionava alla velocità massima di 300 baud. Era compatibile con il modem tipo Bell 103. Per funzionare richiedeva di avere un'interfaccia RS-232 alla quale agganciarsi.





Sotto rivediamo la configurazione completa oggi molto rara. Da notare che non ho descritto il Floppy Drive in quanto equivalente a quanto già esposto nel numero scorso.

stampanti parallele e una porta per il speech synthesizer.



Il sistema di espansione micro 9900 ha un controller del disco DSDD (360k), un'interfaccia RS232 e PIO e 32K di RAM in una forma molto compatta. Questo sistema è collegato al lato della console ed è molto silenzioso (a differenza del PEBOX che è molto rumoroso) ed ha le dimensioni di due sintetizzatori vocali. Quindi è l'ideale per avere una configurazione Tl99 compatta e completa.



Il Tl99/4A al contrario del suo predecessore usa il PEBOX, sempre descritto nel numero precedente, come contenitore nel quale inserire tutte le schede di espansione sopra descritte e altre che verranno.



Qui mostriamo la configurazione ancora in uso da Ron Reuter grande appassionato del Tl99 che cura il sito:

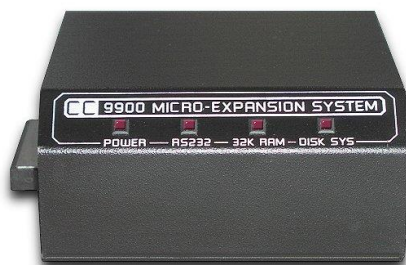
<http://www.mainbyte.com/tl99/index.html>



Di queste schede ne furono create di diversi modelli da varie case costruttrici, partiamo in ordine temporale.. la prima ditta che intervenne dopo il ritiro della Texas Instruments dal mercato degli Home Computer, fu la Corcomp (U.S.A.). Questa nell'arco di pochi anni commercializzò delle schede uguali come uso a quelle della Texas Instruments ma migliorate nelle prestazioni. Ad esempio il controller floppy poteva leggere scrivere floppy da 360k (DS/DD) contro quello della TI che era limitato a 180k (DS/SD). Abbiamo inoltre la scheda seriale, di memoria da 32k e una particolare scheda chiamata Triple Teach che combinava un orologio interno real time, un buffer per le



Molto interessante fu invece dalla Corcomp la creazione del Corcomp 9900 Micro-Expansion System che ebbe un grande successo, in pratica era già all'epoca una soluzione economica, pratica, funzionale di sostituire il PEBOX della TI con tutte le schede già installate. Infatti la TI vendeva solo il PEBOX vuoto e poi uno lo configurava a suo piacere comprando le schede.



Altre schede furono prodotte dalla ditta Myarc (U.S.A.) che ha una menzione speciale per due motivi.. il primo è che commercializzò schede di memoria non solo da 32k ma addirittura da 512kbyte! Il suo controller floppy disk fu costruito in due varianti dove in una poteva formattare i floppy a 80 tracce e fino a 720kbyte (DS/QD), mentre nell'altra

versione abbiamo il primo ibrido controller floppy HD MFM che permetterà l'uso dei floppy HD (1.44Mbyte) e degli hd MFM con tagli a partire da 20MB fino a 255MB.

Scheda Controller Floppy Drive Myarc



Menzioniamo che Paolo Bagnaresi (uno dei massimi esperti Italiani del TI99 che scrisse articoli relativi al TI99 sulla rivista dell'epoca Micro&Personal Computer), insieme a Roberto Maffioletti riuscirono a clonare tale scheda migliorandone il firmware e nella fine degli anni 80 veniva venduta ai soci dei vari user group allora attivi.

Scheda Controller FD / HD Myarc



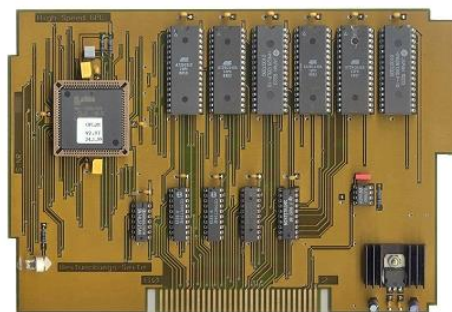
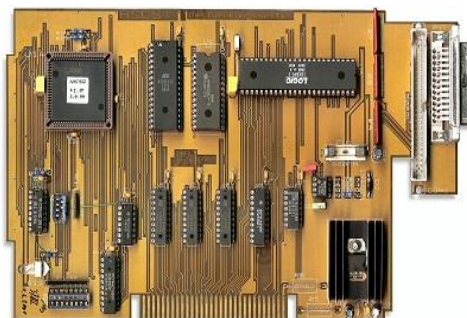
Scheda RS232 Myarc



e descrivendo le schede di produzione Europea, parliamo in particolare di quelle create dal System-99 User-Group

Qui mostriamo le schede create dal SNUG. La particolarità è che sono le migliori come hardware a livello costruttivo e che inserite nella loro totalità creano un super sistema TI99/4A. Di fatto la console è inserita nel PEBox sotto forma di scheda come il Geneve

che vedremo più avanti, ed in più abbiamo una scheda SCSI per interfacciare HD,CD ecc..., una scheda speech synth con molti più vocaboli e fonemi dell'originale, una scheda ram disk che simula floppy di varie taglie 720,360 ecc.. per un totale di 3mbyte, una scheda video con chip grafico 9938 o 9958 superiore all'originale 9918, una scheda emulatrice di Grom/Rom (ne simula fino a 18) che è stata per quasi 30 anni la migliore nel suo tipo di uso soppiantata ora dalla cartuccia Ultimate Grom ed infine una scheda controller floppy disk con clock real time incluso.



Ed anche per questo articolo e' tutto, non mi resta che augurarvi buone vacanze e darvi appuntamento al prossimo numero.

Le avventure testuali: da Infocom a Inform

di Giorgio Balestrieri

Beh, ci siamo, è estate ed il tempo di reclamare il giusto riposo è giunto. In questo doppio numero estivo ci siamo proposti di lasciare da parte la programmazione e complicati listati di codice per dedicarci a qualcosa di più leggero, adatto magari ad una lettura sotto l'ombrellone.

Leggendo gli ultimi numeri di RetroMagazine, alcuni lettori avranno notato (o perlomeno fortemente sospettato) la passione di chi vi scrive per le avventure testuali, a cui è dedicata anche in questo numero la recensione di un gioco. In questo articolo vi parleremo un po' di come nascono questi giochi, non tanto dal punto di vista storico, quanto da quello dello sviluppatore e dei tool utilizzati per crearli. Niente di complicato; come già anticipato, manterremo un livello totalmente divulgativo a tutto beneficio del vostro relax estivo. Beh, quasi almeno, qualche dettaglio tecnico ve lo beccherete comunque, ma niente di impegnativo, promesso.

Per produrre avventure testuali esistevano ed esistono diverse tecniche e tool; qui ci concentreremo su quelli utilizzati dalla Infocom, i cui giochi rappresentano l'eccellenza del genere, ripercorrendo la genesi di questi strumenti che vengono ancora oggi utilizzati da schiere di appassionati che continuano a scrivere storie interattive.

Infocom ed il fantastico mondo delle avventure testuali

Fondata nel 1979 da Dave Lebling e Marc Blank, due studenti del MIT di Cambridge, insieme ad altri otto amici (di cui molti anch'essi provenienti dal MIT), la Infocom è stata la più importante casa di produzione di giochi di questo tipo, tanto da aver definito lo standard di qualità per il settore. Con questo brand sono nate avventure come "Zork", "Deadline" o "A mind forever voyaging", veri capolavori nella storia dei videogiochi in grado di mettere chiaramente in risalto la natura profondamente artistica dei giochi interattivi (altro che giochini elettronici per bambini) e che dimostrano come temi

importanti come la politica ("A mind forever voyaging" fu scritto come critica al governo Reagan) possano essere affrontati e discussi in questo tipo di opera.

La nascita di Zork (e della Infocom)

Le origini di una delle più belle ed apprezzate avventure di tutti i tempi furono frutto quasi del caso. Lebling e Blank lavoravano al Laboratory for Computer Science (LCS) del MIT dove fu sviluppato un linguaggio di programmazione derivato dal LISP chiamato MDL, MIT Designing Language, familiarmente noto col nomignolo di Muddle, che apportava interessanti migliorie al LISP e per cui furono scritte diverse librerie di supporto, una delle quali permetteva la serializzazione degli oggetti (v. riqu. 1), dando così la capacità ai programmatori di salvare lo stato del programma e recuperarlo alla successiva esecuzione. Per mostrare le funzionalità di questa caratteristica, i nostri eroi decisero di scrivere un programma che ne facesse uso. La passione di entrambi per "Colossal Cave Adventure", il gioco che diede origine alle avventure testuali, e l'attività di Blank presso un altro gruppo, non ufficiale, del MIT dove sviluppava giochi di avventura a beneficio degli studenti del campus fecero il resto: avrebbero sviluppato un'avventura testuale. Blank si occupò di definire la trama e gran parte del mondo di gioco, mentre Lebling si concentrò sullo sviluppo del parser (v. RetroMagazine n.2, "The Count", nota 1). Tim Anderson e Bruce Daniels si unirono al progetto e quello che avrebbe dovuto essere semplicemente un demo diede vita a ciò che sarebbe poi stato il primo gioco della Infocom: il leggendario Zork.

Da Zork a Zil e Zip

Come "Colossal Cave Adventure" anche Zork non aveva né grafica né suoni o musiche; tutto ciò che accadeva durante il gioco veniva descritto dal testo che compariva dopo che il giocatore aveva comunicato le sue intenzioni al software tramite la tastiera, ma il suo parser era notevolmente più intelligente. Oltre a semplici comandi come "nord" o "prendi spada", Zork era in grado di capire espressioni

complesse come "Attacca l'orrendo troll con l'ascia bipenne", distinguendo quindi tra più bersagli (se ce ne fossero stati) e più oggetti disponibili nell'inventario con un unico comando. Per raggiungere questo grado di intelligenza, Lebling sviluppò degli algoritmi di scansione del testo per estrapolare da una frase il verbo, ossia l'azione da eseguire, l'oggetto diretto, destinatario dell'azione e l'oggetto indiretto, da utilizzare sul primo. Ma il parser di Zork non si limitava a questo: era in grado di comprendere senza grande sforzo frasi ben più complesse come "leggi il volantino e poi buttalo", mostrando una flessibilità raramente vista nelle avventure testuali, capace di innalzare la giocabilità a livelli altissimi e trascinando letteralmente l'avventuriero nel mondo virtuale creato dal gioco. Per inciso, se in Zork aprite la cassetta della posta che trovate all'inizio dell'avventura e provate poi a dare il comando che abbiamo utilizzato come esempio (in inglese è "read the leaflet then drop it"), noterete che Zork risponderà con "(Taken)" prima di descrivere l'esito delle azioni indicate, a sottolineare il fatto che ha ben capito le vostre intenzioni ed ha preso l'iniziativa per completare il comando prendendo innanzitutto il volantino, azione necessaria ma che non gli è stata specificata, prima di leggerlo e poi buttarlo.

Un parser avanzatissimo dunque, al servizio di una storia avvincente supportata da uno sterminato mondo di gioco e una grande quantità di testo scritto in ottima prosa resero Zork leggenda, facendolo diventare rapidamente un campione d'incassi capace di guadagnare le pagine di giornali come il Times ed il Rolling Stones, nonché il cuore di moltissime persone, compreso quello del compianto Robin Williams, da sempre appassionato di videogiochi al punto di chiamare sua figlia Zelda, che arrivò a telefonare nel cuore della notte ad uno degli autori del gioco per chiedergli consigli su come continuare l'avventura.

Un progetto così vasto comportava però una serie di problemi, in primis quello delle dimensioni. La versione completa di Zork

girava su un mainframe PDP-10 con 512K di memoria che non era proprio quel che si sarebbe definito un computer alla portata di tutti ed il programma occupava più di un megabyte di spazio. Benché oggi queste siano dimensioni talmente piccole da non essere neppure percepite, per i computer e, soprattutto, i microcomputer dell'epoca su cui era destinato a girare, con la loro disponibilità media in K di una sessantina per la RAM ed un paio di centinaia per lo storage su disco, rappresentavano un grattacapo non da poco. Per sciogliere questo nodo, i ragazzi della Infocom lavorarono su due fronti. Il primo consisteva semplicemente nel suddividere il gioco in tre parti che divennero così i primi capitoli della saga, i ben noti "Zork - The great underground empire", "Zork II - The Wizard of Frobozz" e "Zork III: The Dungeon Master". Il secondo fronte fu più articolato, ma permise sia di superare i limiti imposti dalla scarsità di memoria sia di semplificare significativamente la distribuzione del gioco su più piattaforme. Blank ed un altro membro Infocom della prima ora, Joel Berez, risolsero il problema seduti al tavolino del Café della madre di quest'ultimo, dove decisero che la soluzione migliore sarebbe stata quella di progettare una macchina virtuale in grado di eseguire codice contenuto in un file, conservato su qualsiasi supporto di memorizzazione di massa ad accesso casuale disponibile per l'elaboratore su cui la macchina virtuale avrebbe girato. Sì, avete letto bene, le macchine virtuali non sono un'invenzione moderna, furono teorizzate negli anni '60 ed iniziarono a diventare popolari a partire dal 1970 con il linguaggio Pascal, diventando poi di uso comune con il diffondersi di Java per arrivare ai sofisticatissimi ed efficienti software odierni, in grado di emulare più copie di un elaboratore virtuale su un unico sistema fisico. La macchina progettata da Blank e Berez era molto più semplice di quelle a cui siamo abituati oggi ed era specializzata per far girare una classe ben definita di software (non era cioè un calcolatore general-purpose), ma poteva essere eseguita su sistemi reali con risorse limitate e grazie ad essa la Infocom poté commercializzare i suoi giochi per le più diffuse piattaforme dell'epoca, raggiungendo una maggiore utenza con uno sforzo quasi nullo.

La Z-machine, questo il nome della macchina virtuale Infocom ("Z" sta per Zork), implementava una cpu ed un hardware minimale (in **fig.1** ne vedete uno schema), non era nemmeno previsto un framebuffer per la gestione del video che, se necessario, doveva essere fornito dal player, "interprete" nella terminologia Infocom, ma era in grado di eseguire un programma caricandolo a blocchi da un file alla bisogna, realizzando così un rudimentale sistema di memoria virtuale. In questo modo, una volta caricato nella memoria del dispositivo ospite l'interprete (di solito di qualche decina di K), si potevano far girare giochi ben più grandi della RAM disponibile poiché la virtual machine attingeva dal disco al pezzo di programma di cui necessitava in un dato momento. Per scrivere un gioco eseguibile da una Z-machine, venne creato lo Zork Implementation Language (abbreviato in ZIL), modellato sull'MDL in cui era stato scritto originariamente Zork, ed un compilatore che prendeva un sorgente ZIL e lo compilava creando un file detto "storyfile" che poteva essere eseguito dallo ZIP, lo Z-language Interpreter Program (da non confondere con il noto programma di compressione) ossia il player della Z-machine. Chi ha dimestichezza con gli emulatori di console può pensare allo storyfile come alla "rom" da caricare nell'emulatore per far girare

il gioco, la corrispondenza è assolutamente analoga.

Sia ZIL che ZIP meriterebbero di essere approfonditi con una trattazione che renda giustizia a questi due eccellenti esempi dell'ingegno geek ma, come detto in apertura, qui stiamo provando a mantenere il tema sul balneare andante, per cui vi risparmieremo complessi dettagli tecnici. Alcuni punti però meritano comunque di essere portati alla vostra attenzione. L'intero mondo di gioco definito in uno storyfile veniva incorporato in una struttura gerarchica ad albero, che da una "stanza" andava via via contenendo rami che rappresentavano oggetti, personaggi e tutto quello che popolava l'ambiente definito dal gioco. In questo modo era possibile spostare un oggetto (identificato da un ramo) con un unico comando, avendo la certezza che tutti gli oggetti in esso contenuti lo avrebbero seguito nella nuova collocazione. Un altro aspetto interessante era la gestione del testo utilizzato per le descrizioni ed i messaggi. ZIP impiegava cinque bit per indicare un carattere definito da una tabella simil ASCII ridotta. I caratteri erano impacchettati in word di due byte che quindi contenevano tre caratteri ciascuna. In questo modo si conseguiva un doppio scopo: si otteneva una leggera compressione dei testi e li si proteggeva dagli

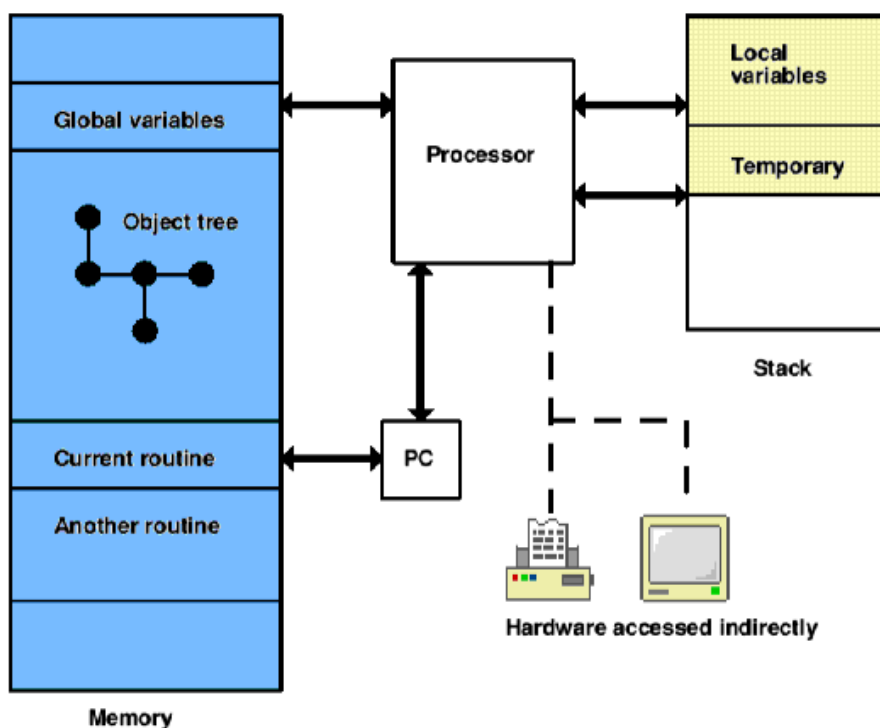


Figura 1 – Schema della Z-Machine

occasionali curiosi che tentavano di esaminare lo storyfile con un editor binario, alla ricerca di indizi per risolvere il gioco. E' altresì interessante osservare che, benché la Z-machine fosse specializzata nell'esecuzione di giochi testuali, la sua composizione era sufficientemente elastica da permettere lo sviluppo anche di altro tipo di software. A riprova di ciò, in rete è possibile trovare in formato ZIP un interprete Basic (baZic), una versione di Tetris (FreeFall) ed una del gioco Snake (Z-Snake) che spopolò sui telefoni Nokia, tutti in grado di girare egregiamente in una Z-machine.

Poiché implementazioni di ZIP sono disponibili ancora oggi per le più svariate piattaforme, la più nota è Frotz, ed esiste almeno un compilatore per ZIL (ZILF), le avventure testuali continuano ad essere prodotte e giocate da una nicchia di irriducibili appassionati, che continuano a tenere in vita questo genere di giochi per computer. Oltre a ZILF, esiste un altro tool per la generazione di storyfile, che implementa un suo linguaggio di programmazione più avanzato rispetto ad ZIL e che di fatto è il più utilizzato dagli aficionados di Interactive Fiction (IF). Il suo nome è Inform e ne parleremo nel prossimo paragrafo.

Inform

Creata nel 1993 da Graham Nelson, Inform è una piccola suite software che implementa un linguaggio per la scrittura di avventure testuali ed un compilatore in grado di produrre storyfile eseguibili da un interprete ZIP. Anche se non è l'unico strumento disponibile, ad oggi è il principale tool utilizzato per la produzione di IF, tanto che persino Marc Blank e Mike Berlyn l'hanno scelto per creare "Zork: The Undiscovered Underground", il nono capitolo della saga di Zork. Dalla sua nascita, Inform ha subito nel tempo notevoli trasformazioni che hanno cambiato radicalmente l'aspetto del suo linguaggio di programmazione tanto che, a seconda della versione, ci si ritrova con prodotti completamente diversi tra loro. La versione di Inform con cui sono state scritte centinaia di avventure è la 6 (Inform 6), caratterizzata da un linguaggio procedurale ad oggetti e fondata su due pilastri principali: il world model ed il parser. Il primo definisce l'insieme delle locazioni che compongono il

mondo di gioco, con tutte le relazioni necessarie per spostarsi coerentemente da un punto ad un altro. Il secondo tenta di interpretare l'input del giocatore per trasformarlo in comandi elaborabili dal software per farlo reagire correttamente alle istruzioni ricevute dall'umano alla tastiera. La compilazione o, se preferite, la configurazione di entrambi è affidata al programmatore, a cui è data grande libertà nel modellarli per rappresentare esattamente come le ha immaginate le vicende a cui sarete chiamati a partecipare. Completato questo compito, è sufficiente creare gli oggetti da inserire nel gioco e definirne i comportamenti per ottenere un prodotto finito e funzionante. Ovviamente, è anche necessario scrivere tutti i testi ed i messaggi per il giocatore, il che probabilmente occuperà la maggior parte del tempo di sviluppo, ma la cura posta in questa fase giocherà un ruolo importante nel successo prodotto finale. Non dimentichiamo mai che un'avventura testuale racconta una storia, anche se dinamica, e non ha il supporto di media audiovisivi per farlo, perciò una buona quantità di testo scritto con uno stile curato è fondamentale per creare l'atmosfera e risvegliare la fantasia di chi la affronterà. Per chi fosse curioso di dare uno sguardo al formato del codice utilizzabile in Inform 6, nel **riq. 2** ne può osservare un esempio tratto dal sorgente di "Salvate lo stregatto" di Marco Vallarino (recensito in questo stesso numero).

Nel 2006 Nelson riscrisse completamente Inform ed il linguaggio che implementava, abbandonando la programmazione procedurale a favore di quella naturale (cioè basata su uno stile che emula il linguaggio umano) ed arricchendo la gamma dei tool con editor dedicati, fondati sul paradigma della progettazione visuale. Anche se battezzato con il nome di Inform 7, esso è un prodotto completamente nuovo, il cui unico legame con la versione 6 è la capacità di produrre storyfile compatibili con le specifiche ZIP. Ancora una volta, se siete curiosi di vedere come appare il codice scritto con questo nuovo linguaggio, date un'occhiata al **riq. 3**. A chi volesse utilizzare Inform 7 per sviluppare giochi, consigliamo caldamente di utilizzare i tool visuali disponibili sul sito dell'autore; seppur sia perfettamente possibile programmare un gioco utilizzando un semplice editor di testi, con i nuovi strumenti creati da Nelson il lavoro di sviluppo viene

notevolmente semplificato, permettendovi di concentrarvi di più sulla storia e sulle interazioni che la faranno progredire.

Oltre Infocom

Finisce qui questa breve tour nello sviluppo di giochi di avventura, in cui abbiamo ripercorso le tecnologie create da Infocom a partire dal lontano 1979 e che vengono utilizzate ancora oggi. Come abbiamo già accennato, i tool basati su di esse non sono però gli unici disponibili per questo scopo. Strumenti come TADS o e-paper adventures rappresentano eccellenti alternative per raccontare storie interattive; se siete interessati a scrivere una vostra avventura, vi invitiamo a dargli un'occhiata.

A chi invece si sta chiedendo se oggi vale ancora la pena di sviluppare un gioco simile, invece di dare una risposta argomentata suggeriamo di fare un giro sugli store iOS ed Android per osservare quanti libri game ci sono in circolazione e quanti giochi di questa categoria e siti a tema vengono fuori da una rapida ricerca su Google. Siamo sicuri che non fatterete a risolvere i vostri dubbi. Aldilà degli aspetti educativi coinvolti nello sviluppo di questo tipo di opera interattiva, una nutrita nicchia di "vecchi" appassionati continua a divertirsi con le IF e un discreto numero di nuove leve inizia a riscoprire un genere mai scomparso del tutto perciò, a chi volesse saperne di più, segnaliamo questi siti:

- **The Z-Machine Standards Document**
<http://inform-fiction.org/zmachine/standards/z1point1/index.html>
- **Learing ZIL**
<http://www.xlisp.org/zil.pdf>
- **ZIP Manual**
<http://xlisp.org/zip.pdf>
- **ZILF - The ZIL Compiler**
<https://bitbucket.org/jmcgrew/zilf/wiki/Home>
- **Manuale Inform 6**
<http://inform-fiction.org/manual/DM4.pdf>
- **Sito ufficiale di Inform 7**
<http://inform7.com/>

- **baZic**
<http://cowlark.com/bazic/>
- **Z-Snake**
<http://ifdb.tads.org/viewgame?id=dlo4yl d75dljqia7>
- **FreeFall (Tetris)**
<http://ifdb.tads.org/viewgame?id=ohjrd vowx6cd7zzx>
- **Zeugma - A Z-machine for 65xx processors**
<http://www.linusakesson.net/software/zeugma/index.php>
- **TADS**
<http://www.tads.org/>
- **e-paper adventures**
<http://www.epaperadventures.qlmagic.com/>

Approfondimento

Nei linguaggi di programmazione, con "serializzazione" si intende la capacità di trasformare un oggetto definito in memoria in un formato che può essere salvato in un file, trasmesso ad un altro programma (via rete o con altro meccanismo di comunicazione) o condiviso con altri oggetti al di fuori del programma che l'ha creato. Poiché la serializzazione agisce sull'oggetto così com'è al momento in cui viene eseguita (agisce cioè sul suo stato), questa tecnica può essere utilizzata per "congelare" un oggetto che può essere ripreso alla prossima esecuzione del programma. Ad esempio, immaginiamo un oggetto che apra finestre di dialogo ed ipotizziamo che dal programma si possano variare i colori ed i font di ogni istanza. Serializzando l'oggetto che rappresenta una particolare finestra, possiamo salvarlo su disco e riprenderlo quando lanceremo di nuovo il programma, ritrovando così i colori ed i font scelti nell'esecuzione precedente.

Questo tipo di operazione viene utilizzata anche da ZIP per salvare il gioco. Poiché ZIP è una macchina virtuale, invece di un singolo oggetto viene salvato l'intero stato della Z-machine, in maniera analoga a quanto accade negli emulatori di console al momento in cui viene richiesto il salvataggio dello stato.

Riquadro 1 – La serializzazione

```
Class Room has light;

Class bancarelle with scenic 'sole' 'cielo' 0 "Niente di interessante.",
    before [; "Non sei qui per le bancarelle della fiera, ma per ritrovare lo Stregatto
perduto!"; ],
    has pluralname female scenery;

Class Interno with before [; "Niente di interessante."; ], has scenery;

Interno "muri"
    with name 'muri' 'pareti' 'parete' 'muro',
    found_in Bar Negozio Retro;

Bancarelle "bancarelle"
    with name 'bancarelle' 'bancarelle' 'stand' 'gazebo' 'libri' 'volumi',
    found_in Via_della_Repubblica_Sud Via_della_Repubblica_Nord Incrocio Via_Genova
Via_Milano;

Room Bar "Barcobaleno"
    with name 'barcobaleno' 'bar' 'locale',
    scenic 'tavolino' 'tavolo' 'sedie' 'bancone' 'banco' 'sedia' 0 "L'arredamento del
bar non ha alcuna importanza per lo svolgimento della tua missione.",
    description "Sei al Barcobaleno, popolare ritrovo di via della Repubblica in cui
ogni scusa @`e buona per abbuffarsi di salatini e brindare alla salute di amici e
parenti, ubriacandosi oltre misura. L'uscita @`e a est.",
    cant_go "L'uscita del bar @`e a est.",
    e_to Via_della_Repubblica_Sud;
```

Riquadro 2 - Esempio di codice in Inform 6 tratto da "Salvate lo Stregatto" di Marco Vallarino

```
"Cave Entrance"

The Cobble Crawl is a room. "You are crawling over cobbles in a low passage. There
is a dim light at the east end of the passage."

A wicker cage is here. "There is a small wicker cage discarded nearby."

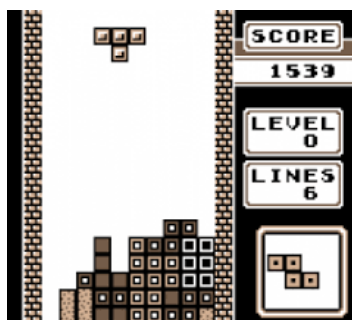
The Debris Room is west of the Crawl. "You are in a debris room filled with stuff
washed in from the surface. A low wide passage with cobbles becomes plugged with mud
and debris here, but an awkward canyon leads upward and west. A note on the wall
says, 'Magic word XYZZY'."

The black rod is here. "A three foot black rod with a rusty star on one end lies
nearby."

Above the Debris Room is the Sloping E/W Canyon. West of the Canyon is the Orange
River Chamber.
```

Riquadro 3 - Esempio di codice in Inform 7 tratto dal manuale del programmatore.

TETRIS



Colonna sonora indimenticabile

Korobeiniki fu un componimento del XIX secolo appartenente alla tradizione popolare russa e fa la sua entrata nel mondo dei videogames grazie a questa versione che una volta imparata non uscirà mai più dalla nostra mente.

GIUDIZIO SUL GIOCO

GIOCABILITA'

99%

Intuitivo, immediato, assuefacente. Il gioco per chiunque, giovane o anziano che sia. Senza istruzione alcuna ci impiegherete una partita a comprenderlo e già ne sarete dipendenti.

LONGEVITA'

80%

Le modalità di gioco sono varie ma non abbastanza da renderlo eterno. Una partita comunque non gliela negherete mai, neanche ad anni di distanza.

Tetris

Bullet-Proof Software / Nintendo - Anno 1989 - Piattaforma GameBoy

E' stato considerato tra i 10 videogames più importanti di sempre.

E' il primo caso di gioco che piaceva indistintamente a ragazzi e ragazze.

E' uscito su tutti i sistemi esistenti da quando ha fatto la sua comparsa nel lontano 1984, nell'allora comunista Russia... e, nonostante tutta la sua celebrità, ne parlerò ugualmente perchè per diventare davvero celeberrimo ci volle la sua pubblicazione in bundle con il GameBoy Nintendo a fine anni 80.

Alexey Pajitnov è il nome del creatore reale del gioco, un ingegnere e game developer russo che non si vide attribuire alcuna royalty prima del 1996 (ma come si suol dire "questa è un'altra storia"). Il gioco girava su Electronika 60, computer sovietico clone dell'occidentale LSI-11, e solo nel 1986 vide la sua conversione in paesi esterni al blocco comunista. PC IBM, Commodore 64, Amiga, Atari ST e via via tutti gli home computer dell'epoca videro la loro conversione del popolare gioco, ma ci volle la versione portatile Nintendo perchè lo si potesse finalmente vedere su una console da gioco, fatta eccezione per l'unlicensed creata per Sega Master System.

Nintendo sapeva di avere una bomba per le mani ma le cose andavano fatte per bene. Mostrare al mondo cosa poteva fare la sua nuova console con un titolo di sicuro impatto? Perfetto, vediamo allora di implementarne le potenzialità.

Le meccaniche di gameplay base del Tetris sono semplicissime: una serie di 7 figure composte da tetramini (figure piane composte da quattro quadrati identici connessi tra di loro lungo i lati) cadono dall'alto in maniera casuale, una per volta, all'interno di una sorta di maxi-contenitore. Il giocatore potrà spostarli lungo l'asse orizzontale e farli ruotare in senso orario o anti-orario tramite la pressione dei tasti A e B. Cliccando in basso sulla croce direzionale si accelererà la caduta mentre nel lato destro dello schermo ci verrà anticipato quale sarà il prossimo pezzo che cadrà dall'alto una volta posizionato l'attuale. Tutto qui? Manca una specifica: una volta completata una riga questa scomparirà. Ne più ne meno.

In questo porting esistono tre modalità di gioco. La prima possiamo definirla classica appunto, poichè segue semplicemente le regole scritte sopra accelerando la velocità di caduta dei pezzi ogni dieci righe completate, andando avanti finchè il giocatore non verrà sconfitto e questo prima o poi accadrà per forza. Diversi matematici si sono interessati alla questione e pare che sul lungo periodo le probabilità giochino contro anche al più concentrato ed esperto dei giocatori, motivo per cui anche mantenendo un'attenzione perfetta, la sconfitta sarà sempre inevitabile...però volete mettere a fare un record eccellente che soddisfazione?

La seconda modalità di gioco è al contrario terminabile. Una serie di schemi già pronti con diversi elementi già piazzati a schermo per renderci meno agibile la sfida. Completiamo 25 linee ed avremo terminato la missione, così da passare al quadro successivo.

La terza ed ultima modalità di gioco è quella multiplayer dato che, come anticipato in precedenza, la Nintendo sentiva il bisogno immediato di mostrare cosa il GameBoy potesse fare con quel suo cavo link (anch'esso presente nel bundle con la console). I due giocatori si sfideranno, interpretando l'uno Mario e l'altro Luigi, nella sfida classica di cui vi ho parlato prima, ma con la possibilità di "infastidirsi" a vicenda. Ogni riga che un giocatore completerà significherà alzare (da basso) le linee che l'altro dovrà completare, presentandosi come una sequenza incompleta di quadratini. Più righe completate contemporaneamente e più sarà "l'alzata" che l'avversario dovrà subire. Il primo che perderà finirà per cedere la vittoria al sopravvissuto. Certo, il tutto a patto di possedere due console e due copie del gioco.

Semplice ed infinito, come solo i migliori giochi sanno essere!

di Starfox Mulder

GREEN BERET



Comincia tutto così...

All'inizio del gioco un breve video ci mostra subito lo scopo della nostra missione



Quattro livelli per quattro boss

Alla fine di ogni livello ci aspettano soldati speciali che sono pronti a farci fuori con ogni mezzo.



Quasi alla meta

Siamo ormai infiltrati nella base russa nemica: anche gli altri prigionieri si agitano e vorrebbero essere salvati.

GIUDIZIO SUL GIOCO

GIOCABILITA'

90%

Il suo successo fu dovuto soprattutto all'ottimo gameplay in cui il nostro berretto verde poteva muoversi.

LONGEVITA'

80%

Se giocato con continuità, Green Beret non è un titolo impossibile da finire...

Green Beret

Konami - Anno 1985 - Piattaforma Arcade

Ci sono avvenimenti storici che con il tempo si affievoliscono e si ricordano meno, altri che invece rimangono indelebili, come una cicatrice sulla pelle. Mi sto riferendo alle grandi guerre come i due conflitti mondiali, la guerra nel Vietnam o a quella nel Golfo.

Ma ci sono anche altri tipi di guerre che non vengono combattute direttamente sui campi di battaglia ma in modo silenzioso nelle stanze dei bottoni.

Appartiene a questa seconda categoria la guerra fredda che si è combattuta tra gli Stati Uniti d'America e la Russia a partire dalla fine del secondo conflitto mondiale fino agli anni 80. Queste due nazioni non sono mai scese direttamente in campo, ma ognuna ha esercitato la propria influenza, soprattutto verso gli stati meno forti, che diverse volte ha sfiorato il pericolo di una terza guerra mondiale.

Questo clima di tensione ha ispirato libri di spionaggio, documentari, film e anche videogiochi. Tra questi ultimi c'è un titolo che è un ricordo vivissimo della mia infanzia da videogiocatore e cioè il coin op Green Beret, il mitico soldato statunitense che doveva liberare i suoi patrioti prima che venissero giustiziati.

Lanciato nel 1985 dalla Konami, Green Beret è un tipico platform a scorrimento laterale in cui il nostro eroe è inizialmente armato di un semplice coltello e deve addentrarsi tra le linee nemiche per salvare alcuni soldati prigionieri dei sovietici.

Il gioco è costituito di soli (si fa per dire) quattro livelli il che non lo rende un titolo lunghissimo, ma non dobbiamo farci ingannare perché le difficoltà sono dietro ogni angolo, compresi i boss finali che sono sempre più difficili da eliminare.

Nel primo livello dobbiamo attraversare una base missilistica al termine delle quale ci aspetterà un camion pieno di soldati nemici. Come arma bonus possiamo trovare un lanciafiamme che incenerisce i sovietici.

Nel secondo livello ci troviamo invece vicino un porto e dobbiamo eliminare i soldati nemici tra palazzi e container prima di vedercela con un branco di cani inferociti e i loro addestratori. Come arma in nostro soccorso abbiamo un lanciamissili.

Nel terzo livello siamo sotto le mura della base russa nemica e, dopo esserci difesi con il nostro inseparabile coltello e con bombe a mano che troviamo lungo il cammino, dobbiamo eliminare alla fine tre soldati che si muovono su mini elicotteri.

Infine nel quarto ed ultimo livello siamo ormai infiltrati nella base nemica e, prima di poter

liberare i nostri commilitoni e far ritorno in patria, dobbiamo affrontare soldati sempre più agguerriti e numerosi e il temibile boss finale, costituito da tre soldati muniti di lanciafiamme che dobbiamo eliminare uno per volta.

Quando fu lanciato dalla Konami non c'erano tanti titoli nelle sale giochi sul tema della guerra e questo contribuì sicuramente a far breccia nei cuori dei videogiocatori. Ma il suo successo fu dovuto soprattutto all'ottimo gameplay in cui il nostro berretto verde poteva muoversi, oltre che da sinistra a destra, anche in verticale, scalando muri, tetti e palazzi e raccogliendo le armi bonus che venivano lasciate lungo il percorso se uccidevamo un particolare soldato russo.

Anche la grafica era ben curata con l'ottima combinazione dei colori che venivano usati in quel periodo. Per non parlare del sonoro con musiche ed effetti sempre adatti ai vari livelli di gioco e con quelle due inconfondibili melodie che accompagnavano il video iniziale e quello finale.

Personalmente sono molto legato a questo titolo perché è stato uno dei pochi coin op che sono riuscito a terminare in sala giochi con un solo gettone e con tanto di pubblico alle spalle pronto ad applaudire la mia impresa. Infatti, se giocato con continuità, Green Beret non è un titolo impossibile da finire e possiamo pian piano imparare le sequenze dei vari nemici ed arrivare a sfidare i boss dell'ultimo livello.

Green Beret ha avuto anche un seguito dal titolo M.I.A. (missing in action) ambientato stavolta nella guerra del Vietnam che però, nonostante la possibilità di muoversi su più livelli di gioco e quella di poterlo giocare in modalità cooperativa, non è riuscito a mantenere né le dinamiche di gioco e nemmeno il fascino del suo predecessore, il cui successo è testimoniato anche dalle diverse conversioni per le varie piattaforme dell'epoca tra le quali ricordiamo quelle per Commodore64, Amstrad, Amiga e MS-DOS.

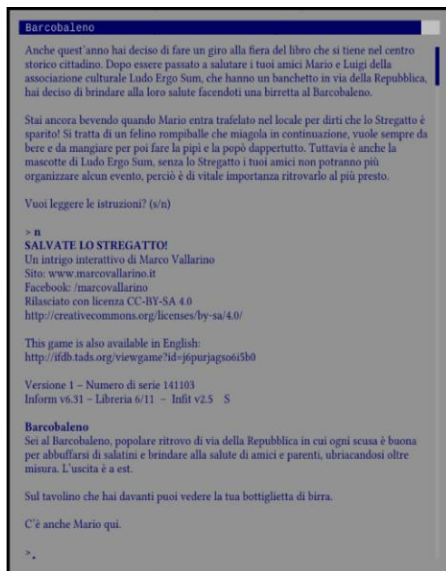
Voglio chiudere questa recensione così come l'avevo iniziata e cioè con un riferimento alla guerra fredda. Il titolo americano infatti era *Rush'n Attack* che tradotto in modo letterale significava scatta e attacca. In realtà era un gioco di parole visto che la pronuncia era *russian attack*, cioè attacco russo, e questa la diceva lunga sui rapporti in quegli anni tra queste due super potenze, che anche nella scelta di un titolo per un videogioco o per un film non perdevano occasione per provocarsi.

di Querino laLongo

SALVATE LO STREGATTO

Un'avventura per chi inizia...

...in tutti i sensi! Questa avventura è dedicata ai più piccoli ed a coloro che si avvicinano al genere per la prima volta, come a chi vuole imparare a programmare un proprio gioco, imparando le basi di Inform.



Il potere dell'immaginazione

Niente grafica, solo testo. Questo apparente limite, permette di liberare appieno l'immaginazione del giocatore e di fargli vivere un'esperienza assolutamente personale, creata dalla sua mente.

GIUDIZIO SUL GIOCO

GIOCABILITA'

83%

Una tastiera e la capacità di leggere e scrivere in lingua italiana è tutto quello che vi serve. Il parser ed il gioco sono ottimamente congegnati e difficilmente vi ritroverete in una "caccia alla parola", terrore di ogni avventuriero.

LONGEVITA'

??%

Parlare di longevità nelle avventure testuali ha già poco senso di suo, in questo caso non ne ha nessuno. Stregatto ha lo scopo di fungere da introduzione per giocatori e aspiranti game designer, in cui l'aspetto didattico è predominante sugli altri che caratterizzano i titoli "tradizionali". Una volta completato, è poco probabile che lo rigiocherete, se non per nostalgia a distanza di anni. Chi lo userà per apprendere i concetti fondamentali del game design e di Inform ci passerà su molto più tempo, giocandolo più e più volte per studiarlo o per testare nuove idee.

Come di consueto, prima di occuparci del gioco a cui questo articolo è dedicato, parleremo un po' del suo autore e delle vicende che l'hanno portato ad occuparsi, in prima persona, di giochi d'avventura.

Marco Vallarino, classe '77, giornalista, scrittore ed autore di avventure testuali, nasce ad Imperia ed in teoria avrebbe dovuto fare l'ingegnere, ma la passione per la scrittura prese velocemente il sopravvento e il buon Vallarino si ritrovò a condurre una carriera completamente diversa. Il primo incontro con le avventure testuali avvenne in edicola, dove il nostro scoprì le collane Viking e Explorer pubblicate da un'altra nostra conoscenza, Bonaventura Di Bello. L'amore per il genere fu del tipo "a prima vista" e ben presto Vallarino passò a titoli più impegnativi e maturi come Zork e The Pawn (Magnetic Scrolls). Ormai appassionato di avventure testuali e libri game (su tutti, la saga di Lupo Solitario) e con una propensione alla scrittura, il passo che lo portò a scrivere giochi fu abbastanza breve. Le sue prime avventure le scrisse in Basic per PC ma presto passò ad Inform 6, un linguaggio specialistico per software di avventura, capace di produrre codice per le Z-machine inventate da Infocom, che offriva grandi vantaggi per gli sviluppatori di avventure testuali, in primis il supporto di un linguaggio ad hoc per questo genere e la capacità di poter distribuire i giochi per più piattaforme grazie all'uso di una macchina virtuale. Con questo linguaggio ha prodotto piccoli capolavori come Enigma o la saga di Darkiss e "Salvate lo Stregatto" che, pur non essendo il più rappresentativo delle capacità dell'autore, è forse il gioco più noto grazie alla sua natura didattica ed all'utilizzo che Marco ne fa durante gli eventi pubblici sul tema, come quelli tenuti all'università di Imperia, al Gizmark di Genova o all'accademia della comunicazione di Milano.

Il gioco

La trama del gioco, piuttosto breve, è contenuta nell'introduzione che compare all'inizio dell'avventura. Siete in visita alla fiera del libro che si tiene nel centro storico della città, dove sono presenti con uno stand anche i vostri amici dell'associazione "Ludo ergo sum". Mentre vi rilassate al bar di zona, il vostro amico Mario arriva con una nefasta notizia: è sparito lo Stregatto! L'odioso felino, le cui principali occupazioni sono mettere a dura prova i nervi di chi gli sta intorno con continui miagolii, lasciare rifiuti solidi e liquidi in ogni dove e mangiare a sbafo qualunque cosa commestibile gli capiti a tiro, non si trova più e sembra che anche molti altri gatti della zona siano misteriosamente scomparsi negli ultimi giorni. Per quanto consideriate la cosa un vantaggio per il bene comune, il detestabile quadrupede è purtroppo la mascotte del "Ludo ergo sum", per cui non vi resta altro che accettare la richiesta di Mario e di mettervi alla ricerca dello Stregatto.

Disponibile in due lingue, italiano ed inglese, questo gioco racchiude il suo punto di forza nella sua duplice natura didattica, dal momento che è stato pensato sia per permettere a chi non ha mai visto

Salvate lo Stregatto

Marco Vallarino - Anno 2013 - Multiplatforma

un'avventura testuale di familiarizzare ed apprezzare il genere, sia a chi voglia imparare a scrivere giochi di questo tipo di muovere i primi passi, grazie al rilascio in forma open source del sorgente completo in Inform.

Per gli avventurieri di lungo corso, completare il gioco richiederà una ventina di minuti, senza nemmeno dover disegnare la mappa delle locazioni tanto è semplice, ma per i neofiti richiederà qualche ora visto che, pur nella sua semplicità, "Salvate lo Stregatto" contiene tutti i meccanismi tipici delle avventure testuali, come il dover esaminare oggetti e ambienti più volte a seconda del momento di gioco, combinare più oggetti per raggiungere un obiettivo, eseguire azioni per provocare altre utili all'avanzamento della trama ecc.

Per gli aspiranti scrittori/programmatore, avere a disposizione il codice di un gioco semplice ma completo rappresenta un preziosissimo aiuto nella comprensione di Inform ed una notevole base di partenza per iniziare a sperimentare con questo linguaggio.

Giusto a titolo di cronaca, segnaliamo che Vallarino ha utilizzato lo Stregatto e l'Inform per insegnare ai ragazzi dell'I.I.S. "G. Marconi" di Imperia a scrivere avventure, lavorando con gli studenti ed i loro professori durante tutte le fasi dello sviluppo, dalla scrittura della trama dal punto di vista artistico alla progettazione delle meccaniche di gioco. L'esperimento ha dato vita a tre giochi, "Visita al Marconi", "In cerca di Tesla" e "Fuga dall'Acropoli", tutti scaricabili gratuitamente. Se volete saperne di più, se ne parla su Old Games Italia a questo indirizzo:

<http://www.oldgamesitalia.net/listaavventuretestuali/fuga-dallacropoli>

Conclusioni

Anche se siamo dinanzi ad un gioco facile da giocare e breve da concludere, "Salvate lo Stregatto" è assolutamente godibile e merita senza dubbio di essere giocato. Anzi, proprio per la sua leggerezza, è perfetto come passatempo estivo, magari sotto l'ombrellone, giocato sul vostro tablet al posto (o in aggiunta) al classico Sudoku da spiaggia. Per ottenerlo, basta andare a questo indirizzo

<http://avventuretestuali.xoom.it/stregatto.html>

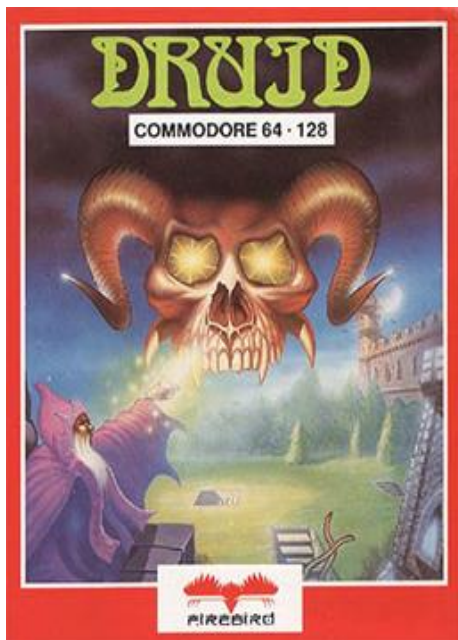
e scaricare il pacchetto completo. Dopodiché non dovrete fare altro che scegliere un interprete di Z-code, come ad esempio Frotz, ed iniziare a giocare. Gli interpreti per Z-machine sono disponibili per praticamente qualsiasi piattaforma, tablet e smartphone compresi, per cui non avete scuse: giocate e divertitevi!

di Giorgio Balestrieri

DRUID

Druid

Firebird - Anno 1986 - Piattaforma: Commodore 64



Fantasma più scalinata per i sotterranei, bene!

GIUDIZIO SUL GIOCO

GIOCABILITA'

70%

Non c'è pace, bisogna sempre scappare, attaccare i nemici, scegliere l'incantesimo adatto e studiare bene la mappa e la strategia da adottare, non per ultimo bisogna riconoscere che un poco di fortuna sarà molto utile!

LONGEVITA'

50%

Amstrad CPC, Atari 8-bit family, Atari ST, Commodore 64, ZX Spectrum, Amiga, MSX2, Tatung Einstein, Famicom Disk System. Queste sono tutte le piattaforme sul quale Druid è stato sviluppato, voi mi chiederete, "ma che significa relazionare la quantità di piattaforme con la longevità del gioco?" Ebbene, l'inversa proporzionalità che li accomuna è netta! Un gioco di una difficoltà incredibile che ha dettato fama e rabbia, quindi un 100% e un 0% potrebbero essere plausibilmente commentabili, non resta che dare un motivato 50% e provare, almeno per chi non ci ha mai giocato, a dedicargli, cinque, sì, solo cinque minuti e rendersi conto che o lo si ama o lo si odia!

Hasrinaxx, un nome tipico per un personaggio che si vuol far ricordare per secoli e secoli, un nome da Druido che finirà per contagiare numerose piattaforme, non solo il nostro carissimo Commodore ma anche Atari, Zx Spectrum, Amstrad CPC e Tatung Einstein. Se dobbiamo ricordare un gioco della nostra infanzia che ci ha lasciato un retrogusto di claustrofobia, agitazione e sensazione di impotenza nel completare gli schermi, probabilmente potremo annoverare Druid in questa pole position di pura angoscia.

Il nostro Hasrinaxx è un Druido, poco si sa nell'arcano mondo del web, pertanto rimane un misterioso personaggio avvolto nel suo mantello, che si muove in modalità multidirezionale lungo una visuale isometrica nonché a scorrimento. I livelli sono numerosi, così come i nemici da sconfiggere, direi che gli angusti spazi di movimento nel labirinto e l'improvvisa apparizione di nemici da ogni dove, potrebbe evocare perfettamente un attacco di panico tipico dell'ascensore in blackout.

I nemici sono di vario tipo, giganti pseudo-insetti, fantasmi e chiaramente demoni. Le armi in nostra dotazione sono suddivise in elettricità, fuoco ed acqua. Ogni arma è chiaramente in quantitativi a scalare, ogni nemico richiede l'arma giusta. In più Hasrinaxx può evocare il mostruoso ed antropomorfo Golem in caso di bisogno, oppure può diventare invisibile per un breve tempo. Potremo anche usare una sola volta l'incantesimo del teschio che distruggerà tutti i nemici dello schermo.

Nella schermata del gioco troveremo disseminati anche degli scrigni, contenenti determinati elementi che, dopo una rapida, rapidissima riflessione, dovremo decidere se raccogliere o meno. Sicuramente le chiavi apriranno le porte chiuse, queste le prenderemo! Infine sul terreno potremo trovare la celeberrima ed intramontabile magica stella a cinque punte che ricarica l'energia persa durante i combattimenti.

La strategia da studiare in ciascun schermo ci fa ben intuire l'arguzia del programmatore, creatore di un gioco che a grandi linee potrebbe assomigliare a Gauntlet. Nella realtà Druid è dotato di una propria personalità che ne ha decretato un successo sia a livello di multiplatforma sia a livello di giocabilità in due. Sì, Druid permetteva di utilizzare entrambe le porte dei nostri amati Joystick: uno poteva essere usato dal giocatore che muoveva Hasrinaxx mentre

l'altro muoveva il Golem evocato dal Druido. Perspicace il programmatore, oserei dire anche rivoluzionario per quei tempi.

Una piccola curiosità, l'autore si chiama Dene Carter, ha sviluppato il gioco chiedendo i soldi per finanziare il progetto a... papà! Si perché lo sviluppatore aveva solo 15 anni. Il padre non ha esitato a contribuire al progetto, aveva infatti individuato doti di spiccato talento nel proprio figliolo.

Ritornando al gioco, il nostro Hasrinaxx deve sconfiggere il perfido Acamantor e la sua armata di demoni. Il mondo sarà liberato quando tutti i livelli saranno completati. Il primo livello è a piano terra. I seguenti vanno sottoterra. Ognuno al piano sotto e sotto ancora. Ecco che ritorna l'analogia dell'ascensore in blackout. Quel senso di claustrofobia che insieme ad insetti, streghe e fantasmi ci condurrà belli carichi di ottime prospettive a combattere, ad uno ad uno, i quattro principi demoni.

Una breve guida strategica di Druid la possiamo trovare in:

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

Sarà veramente utile seguire questi preziosi consigli, soprattutto perché non esiste la possibilità di salvataggio del livello.

Se, come me, avete amato le musiche del gioco allora non potete perdervi il sito dei remix amatoriali:

<http://remix.kwed.org/index.php?search=Druid>

Un'altra curiosità, il porting di Druid è stato eseguito, nel 1988, anche per il Famicom Disk System dalla Jaleco e, non per ultimo, anche per MSX.

Che dire? Dene Carter ha sviluppato un videogioco veramente accattivante e complicato nonostante i suoi 15 anni, il tutto servito da una colonna sonora molto orecchiabile curata da David M. Hanlon. Il tutto impiattato attraverso il codice di Andrew E. Bailey. Un gruppo niente male.

Oltretutto il gioco Druid ha partorito Druid II (Enlightment) ed infine Warlock (The Avenger). Una terna di giochi fantasy che sebbene abbiano preso spunto da Gauntlet, godono di una propria giocabilità e personalità, provare per credere.



Zero teschi, zero Golem e zero chiavi



Il porting minimal-affascinante degli 8bit



L'occhio e' posizionato su "fire" ma il teschio?



Immagine da piattaforma MSX

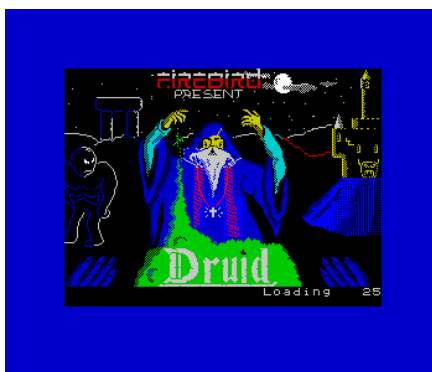


Immagine da ZX Spectrum



Immagine da piattaforma CPC Amstrad

La critica degli anni ottanta si era espressa con votazioni ottime, basti pensare che Zzap! aveva dato 88 su 100, Amstrad Action 93 su 100, Computer game 18 su 20. Tranne alcune riviste come Tilt su piattaforma Zx Spectrum che aveva dato un misero 12 su 20. Io nel punteggio della giocabilità darei un 70% di voto, chiaramente l'aspetto nostalgia è uno dei vari ingredienti ma non è quello principale. Ebbene, Druid è stato il classico gioco che una volta conosciuto, o lo si ama o lo si odia.

Riassumendo Druid, lo possiamo annoverare tra i grandi giochi del passato per tutta la serie di peculiarità elencate sopra, sommata ad elementi degli anni ottanta di indiscutibile pedigree: pubblicato dalla Firebird Software, multiplatforma, il genere del gioco si può inserire in "gioco di azione" con prospettiva classica di quei tempi "Bird's-eye", uno scrolling 2D con un gameplay di tipo arcade e ambientazione fantasy.

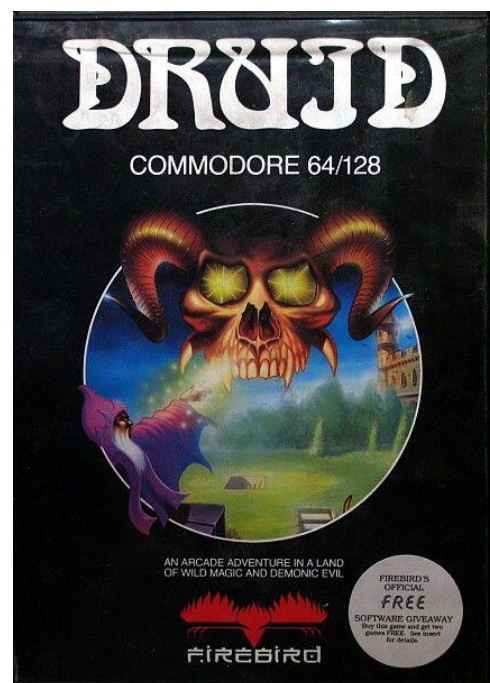
Ecco, questo è Druid, ho voluto scrivere questo articolo per non far svanire un bellissimo e geniale gioco nell'oblio dell'abbandono. Mi riprometto di perseguire in questo scopo, riportare al presente del fantastico retrogaming le perle del passato e invitare tutti noi a rivivere quelle entusiasmanti esperienze. Quindi nel prossimo articolo, con questo spirito di rivisitazione, vi proporrò il bellissimo Rimrunner su piattaforma C64.

Eccovi infine una serie di schermate del gioco che confrontano le varie piattaforme del porting. E' un po' una mia fissazione il porting, più ne avviene più significa che il gioco è stato

gradito? Non solo, il porting ha svariate peculiarità di interpretazione tra le righe, una potrebbe forse essere che spostarsi di piattaforma porta a sfociare in grafiche e gaming di reciproca e costruttiva battaglia.

Concludo invitandovi tutti a giocarci. Sebbene il gioco sia stato creato da un quindicenne, vi sfido ad uscire "vivi" da questa particolare avventura!
Ciao!

di Michele Ugolini



SUPER BURNOUT



GIUDIZIO SUL GIOCO

GIOCABILITA'

90%

Super Burnout è un gran titolo esclusivo per Jaguar ed è un peccato che sia passato così inosservato al tempo della sua uscita.

LONGEVITA'

80%

Il gioco prevede diverse modalità: corsa singola, time attack, testa a testa a due giocatori e la modalità campionato!

Angolo Oscurita': Super Burnout

Shen Technologies - Anno 1995 - Piattaforma Atari Jaguar

Siamo nel 1995. L'Atari Jaguar scivola verso l'oblio, mentre Saturn e Playstation escono a livello mondiale. Le vendite della console sono estremamente scarse; tra le mille colpe imputate ad Atari c'è soprattutto quella di aver creato un sistema potente ma con un hardware troppo complicato.

La casa americana, in un ultimo disperato colpo di coda, affretta l'uscita dell'unità CD, sperando invano di poter recuperare credibilità presso il pubblico. Nonostante il palese fallimento, ci sono ancora diversi sviluppatori interessati a questa macchina. Uno di questi è Shen Technologies, un piccolo team francese composto da tre giovani programmatori, che da anni lavora e crea demo su Atari ST.

Il primo progetto videoludico di questo gruppo è proprio Super Burnout, la cui produzione ha inizio nel 1994 e si conclude in poco meno di un anno.

I ragazzi sono costretti per la maggior parte del tempo a lavorare nei fine settimana, quando un'altra casa francese, tale "Virtual Xperience", permette loro di usare il kit di sviluppo per Jaguar. Tutto questo impegno porta però buoni frutti, perché il gioco si rivela una delle migliori esclusive uscite sulla macchina Atari.

Super Burnout è un titolo di corse motociclistiche, in cui dobbiamo affrontare otto circuiti sparsi in tutto il mondo a cavallo di sei diversi tipi di moto.

La grafica è davvero spettacolare; il gioco è totalmente bidimensionale e sfrutta la grande capacità di scaling del Jaguar per rendere efficacemente il percorso e gli oggetti che lo contornano: moto, alberi, recinzioni, nuvole, cartelloni pubblicitari, palazzine, gradinate gremite di pubblico.... tutto ciò che vedremo su schermo è bello e ben definito. Ma la cosa che fa davvero rimanere a bocca aperta è la sensazione di velocità. Il gioco infatti fila costantemente a 60 frame al secondo senza alcun incertezza, persino nella modalità a due giocatori.

Benché all'epoca tutto il mondo si fosse ormai lanciato sui titoli poligonali, Super Burnout dimostrava che la buona vecchia grafica bidimensionale poteva ancora dire la sua.

Come ciliegina sulla torta, inoltre, abbiamo una grande cura riposta nella caratterizzazione dei percorsi, che sono a tema con il paese rappresentato (sia nel paesaggio che in alcuni elementi scenici, come i ciliegi in fiore in Giappone) e che, in alcuni casi, prevedono anche gare al tramonto ed in notturna.

Per il sonoro valgono gli stessi elogi; uno dei punti di forza principali del Jaguar è proprio la grande resa del chip audio, capace di riprodurre musiche di qualità cd, anche se su cartuccia. I titoli che lo hanno saputo sfruttare davvero bene sono pochi, e Super Burnout è fra questi.

Le musiche sono di ottima qualità e ci accompagneranno sia nei menu che sulle piste vere e proprie, risultando sempre evocative e ben fatte. Ottimi anche gli effetti sonori, così come la qualità della voce dell'annunciatore, chiara e ben campionata.

La vera bellezza del titolo risiede ad ogni modo nella giocabilità. Affrontare i percorsi con la nostra moto si rivela estremamente divertente, grazie ad un modello di guida semplice all'apparenza ma profondo nella sostanza. Ognuno dei 6 bolide (più uno segreto) a nostra disposizione, infatti, si comporterà in modo molto diverso in pista, e starà a noi trovare la due ruote più adatta alle nostre esigenze. I tracciati si dividono in tre tipi principali: "tecnico", "semi-tecnico" ed "alta velocità" e vanno affrontati varie volte prima di ottenere risultati ottimali.

Il gioco prevede diverse modalità: corsa singola, time attack, testa a testa a due giocatori e la modalità campionato. In quest'ultima dovremo correre di seguito tutte ed otto le gare, e non sarà semplice riuscire ad arrivare nelle prime posizioni. Il segreto diventa trovare una moto ben bilanciata, che possa rendere efficacemente in tutti i percorsi. Tra le varie opzioni che potremo modificare troviamo il numero di giri, la difficoltà degli avversari che affronteremo ed il tipo di trasmissione. Potremo inoltre cambiare la configurazione del pad, a seconda dei nostri gusti. Il gioco salva automaticamente le opzioni ed i tempi migliori, pertanto potremo tornare a correre quando vorremo, senza dover ripartire da zero.

Il pad del Jaguar, spesso criticato per la poca reattività nei giochi, qui si comporta in maniera assolutamente perfetta risultando fluido e rapido nella risposta.

Tirando le somme, Super Burnout è un gran titolo esclusivo per Jaguar ed è un peccato che sia passato così inosservato al tempo della sua uscita. Riscoprirlo è dovere di ogni appassionato del felino di casa Atari, ma non posso che consigliarlo anche a tutti gli amanti di giochi di corse.

Non ne rimarrete delusi!

di Federico Gori

CAPTAIN BLOOD



GIUDIZIO SUL GIOCO

GIOCABILITA'

80%

Stiamo parlando di uno dei titoli più controversi ma al tempo stesso profondi di un lontano 1988.

LONGEVITA'

85%

Uno dei punti forti più evidenti del gioco firmato da Ere Informatique per il publisher Infogrames era la trama

Ere Informatique, Infogrames - Anno 1988 - Piattaforma Olivetti Prodest PC 128

Di tutti i giochi usciti (anche) su Olivetti Prodest Pc 128 è probabilmente quello che ha sfruttato al meglio, se non al massimo, le sue potenzialità grafiche. Inoltre, grazie alla localizzazione in italiano, era possibile goderselo fino in fondo punto per punto.

Stiamo parlando di uno dei titoli più controversi ma al tempo stesso profondi di un lontano 1988. Ecco dunque Captain Blood che potremmo definire un'avventura grafica dai toni decisamente cyber. Uno dei punti forti più evidenti del gioco firmato da Ere Informatique per il publisher Infogrames uscito principalmente su Atari ST (dove è stato sviluppato originariamente), su Amiga, C64, Sepctrum ed Amstrad CPC, era la trama.

La storia, infatti, era splendidamente riassunta nel libretto di accompagnamento che descriveva anche per filo e per segno le razze presenti nel gioco. Ed ecco che facciamo la conoscenza di Bob Marlock, uno sviluppatore di videogiochi, che assillato dal lavoro sta ultimando la sua opera. Ma accade un fatto imprevisto: egli è così preso ed assorto che, dopo aver digitato il "run" per eseguire la sua ultima creazione, si ritrova intrappolato in essa e prende il nickname di Captain Blood che dà anche il nome a questo interessante gioco ideato e realizzato da Didier Bouchon e Phillipe Ulrich.

I più attenti avranno anche notato l'omaggio degli autori all'omonimo romanzo di Rafael Sabatini dal quale venne tratto un film di Michael Curtiz del 1935 che fu anche candidato ai premi Oscar dell'anno successivo grazie anche ad Errol Flynn protagonista di quella pellicola. Altri ricorderanno questo gioco su Amiga ed Atari ST anche per la presenza del brano del maestro della musica sintetizzata Jean-Michel Jarre. Il pezzo è una versione ripresa di Ethnicolor presente nell'album Zoolock del 1984.

Ma di cosa parla il gioco? Bob Marlock, dicevamo, intrappolato nel suo stesso videogioco, vaga nella nebulosa dell'Hydra al comando di una poderosa arca bio-meccanica ma in seguito ad un incidente durante l'iperspazio viene clonato 30 volte. Queste copie tolgono a lui energia vitale e lo rendono sempre più simile ad un droide. Parte, dunque, la caccia ai suoi cloni per eliminarli perché questo è l'unico modo di mantenere ciò che resta della sua umanità. Una caccia lunghissima ci porta ad 800 anni dopo gli eventi iniziali. Il nostro eroe è riuscito a trovare 25 dei 30 cloni. A questo punto entriamo in gioco noi per l'ultima parte di questa missione che non è altri che una lotta contro il tempo. Dovremo, così, vagare per questa nebulosa (che include oltre 32.000 pianeti) seguendo anche le indicazioni delle

Captain Blood

varie razze presenti nei tantissimi pianeti. Ne nasce un'avventura ricchissima anche di sfaccettature sociali. Le tante popolazioni che incontreremo avranno molto da raccontare.

La parte più profonda di Captain Blood è il dialogo con queste creature per prendere indizi utili alla nostra ricerca. Il gioco stesso ci fa iniziare da un pianeta abitato per farci iniziare a prendere confidenza con l'interfaccia di dialogo universale UPCOM composta da circa 150 icone. Ci sono tutti i verbi e le parole principali per imbastire frasi di senso compiuto e sarà fondamentale usare i toni giusti. Essere accomodanti può non essere efficace con chi è guerrafondaio, al contrario essere aggressivi potrebbe offrire delle chance per ottenere preziose indicazioni con popolazioni meno battagliere. Bisogna quindi imparare ad adattarsi al linguaggio di ogni razza elargendo favori, anche, imparando anche a comportarci per ottenere il meglio. Scopriremo anche che ci sono popolazioni in guerra tra loro ed in questo turbinio dovremo trovare questi cloni per convincerli a farli salire a bordo dell'arca, ibernarli e distruggerli. Captain Blood offre un colpo d'occhio notevole per l'epoca e se questo era lecito attendersi su Amiga e su Atari ST, la conversione per Olivetti Prodest Pc 128 non aveva nulla da invidiare a livello visivo alle controparti a 16 bit ma era completamente muta, il che visto l'hardware della macchina in questione, era anche un bene. La avveniristica interfaccia funzionava al meglio mentre la navigazione attraverso l'immensa Hydra era immediata grazie ad un enorme touch screen dal quale scegliendo un punto preciso nella mappa si davano le coordinate per la destinazione desiderata. Sveltava il braccio bio-sintetico di Captain Blood. Dopo un viaggio per l'iperspazio si giungerà al pianeta e bisognerà lanciare una sonda biologica, la OORXX, ed esplorare la superficie del pianeta fatta di paesaggi frattali evitando anche eventuali ostacoli per arrivare in fondo a quella che è una valle frastagliata e vedere se questo pianeta è abitato o meno. Se abitato si vedrà un alieno e si comincerà ad imbastire una conversazione.

L'arca offre anche la possibilità di scansionare la superficie dei pianeti ed addirittura distruggerli o di accogliere a bordo alieni per ibernarli e rilasciarli in altri pianeti. Questo dipenderà dal tenore dei dialoghi che poi scaturiranno delle "missioni" per il nostro protagonista per avere le preziose informazioni e coordinate per trovare i cloni. Captain Blood ci è rimasto nel cuore come uno dei titoli più innovativi di fine anni '80 dalla forte atmosfera ed in grado di trasmettere un senso di solitudine nel vagare nello spazio.

di Edoardo Ullo

Wimbledon Championship Tennis

WIMBLEDON CHAMPIONSHIP TENNIS



Superficie in erba

Proprio come a Wimbledon la pallina schizza veloce



Superficie in terra rossa

Su questo tipo di campo la pallina ha un rimbalzo particolare, come se fossimo al Roland Garros



Superficie in cemento

Infine non manca il campo in cemento dalla velocità media. Se poi lo giochiamo in doppio è ancora meglio

GIUDIZIO SUL GIOCO

GIOCABILITA'

90%

Un gioco perfetto sotto tutti i punti di vista...

LONGEVITA'

90%

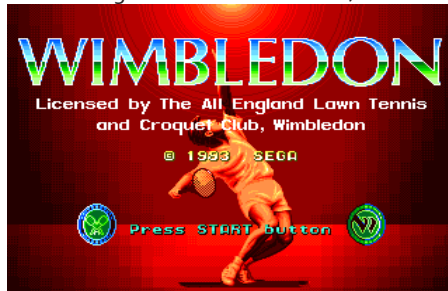
... e soprattutto piacevole da provare ancora oggi!

Quando l'inverno decide di tornarsene in letargo per dare spazio alle belle giornate a risvegliarsi non è soltanto la natura, ma anche alcuni sport come ad esempio il tennis. Infatti dall'erba di Wimbledon fino alla terra rossa del Roland Garros questo sport vive, dalla primavera all'autunno, la sua stagione più bella ed entusiasmante, che spesso fa venire voglia anche a noi di impugnare la racchetta e sfidare i nostri amici come novelli Agassi o Sampras.

Ma per chi non ha voglia di sudare ci sono sempre i videogiochi tennistici che, a livello sportivo, forse sono secondi solo ai titoli dedicati al calcio.

Sin da quando ero bambino ho sempre amato questo genere di simulazioni, partendo dal mitico commodore64 fino alle console di ultima generazione, non dimenticando poi i vari titoli arcade che sono passati nelle sale giochi.

Tra i vari videogames quello che secondo me però ha raggiunto un ottimo livello sia di simulazione che di giocabilità rimane Wimbledon Championship Tennis, prodotto nel 1993 da casa Sega per la sua piattaforma più riuscita e cioè il Megadrive (chiamato Genesis negli Stati Uniti d'America).



In questo gioco, dopo aver impostato la modalità esibizione o coppa Wimbledon, possiamo scegliere tra tanti tennisti maschili o femminili, ognuno con le proprie caratteristiche, qualità, punti di forza e anche lacune. Poi, dopo aver impostato il livello di gioco, possiamo decidere su quale superficie affrontare il nostro avversario, scegliendo tra il classico campo in erba (il terreno più veloce), quello in cemento (dalla velocità media) e infine quello in terra rossa con il suo caratteristico rimbalzo della pallina.

Una volta che iniziamo la partita ci accorgiamo subito che Wimbledon Championship Tennis ha davvero un ottimo gameplay, perfetto mix tra un gioco arcade ed uno di simulazione, in cui con il nostro tennista possiamo davvero provare qualsiasi tipo di colpo, dal rovescio al dritto, dal colpo in top spin allo smash, senza dimenticare palle corte e colpi sotto rete.

Sega - Anno 1993 - Piattaforma Megadrive

Il gioco si avvale di tre pulsanti con i quali possiamo provare tre tipi diversi di colpi come quello normale, quello veloce ed il pallonetto. Inoltre tutte queste combinazioni le possiamo utilizzare nella battuta, potendo quindi scegliere tra diversi stili di gioco e strategie per sorprendere il nostro avversario di turno.



Il gioco si presenta anche con una grafica ben curata e molto colorata per lo standard di quel periodo e il sonoro accompagna ogni fase di gioco, sia con effetti tipici del mondo del tennis che con le acclamazioni del pubblico pronto ad accendersi davanti a colpi spettacolari.

Inoltre il tutto è arricchito da belle animazioni e da una schermata che compare ogni cambio di campo che, oltre a ricordarci il punteggio, ci permette di fare una breve pausa per tirare il fiato e concentrarci sul prossimo game.

In questo gioco non mancano nemmeno i raccattapalle e il mitico tie break che ci permette di decidere un set finito in parità.

Se poi siete amanti come il sottoscritto della modalità multiplayer sappiate che in questo titolo si può giocare con un amico (contro o in doppio) e collegando il mitico multi tab addirittura fino a quattro giocatori, possibilità davvero innovativa per quel periodo.



Se proprio devo trovare una pecca a questo videogame è la mancanza delle licenze ufficiali dei nomi, in parte sostituito dalla possibilità di mettere il nostro nome reale. Ma questo non può certo intaccare un gioco perfetto sotto tutti i punti di vista e soprattutto piacevole da provare ancora oggi, da solo o in compagnia.

Di Querino Ialongo

SID MEIER'S PIRATES!



La schermata iniziale...

...con una musicchetta "rinascimentale" inizia la nostra avventura come pirati del mar dei Caraibi.



La navigazione

La navigazione per mare è controllata dal joystick ed è influenzata dalla direzione e dalla forza dal vento che, oltre che con il suono, è intuibile dallo spostamento delle nuvole. Attenzione alle secche e agli scogli!

GIUDIZIO SUL GIOCO

GIOCABILITA'

95%

I controlli di gioco sono molto semplici, tramite joystick per la navigazione e le battaglie; diverse scelte testuali nelle taverne, con i governatori o abbordando le navi.

LONGEVITA'

90%

Quattro livelli di difficoltà, un gameplay avvincente e rilassante allo stesso tempo, per un gioco che resterà negli annali della storia videoludica.

Sid Meier's Pirates!

Microprose - Anno 1990 - Amiga

A.D. 1992, nelle camerette di 4 giovani sottotenenti dell'esercito italiano campeggiano un Amiga 500 e un Amiga 2000, il mio...in quel periodo, le attività preferite dai 2 PC (e dai 4 S.Ten. 😊) si dividevano tra i mitici "cartellino gaillo" e "calcio d'angolo" di KickOff e le avventure caraibiche di Pirates!

non influiscono sulla difficoltà del gioco. Infine possiamo scegliere il livello di gioco da "apprentice" (il più semplice) a "swashbuckler" (il più difficile), che cambierà il livello di collaborazione dell'equipaggio e ovviamente influenzerà di conseguenza la difficoltà del gioco, e una delle "abilità" personali, personalmente ho quasi sempre scelto l'abilità nella schermata.



Pirates! è uno dei giochi di maggior successo del mitico Sid Meier, noto ideatore di giochi come Railroad Tycoon e Civilization (che magari recensirò in uno dei prossimi numeri)...solo per citarne altri due. E' nato nel 1987 per Commodore 64 e nel 1990 viene pubblicata l'edizione per Amiga, con grafica e musica migliorate. Nominato come miglior gioco dell'anno da almeno 3 riviste dell'epoca, è stato pubblicato per diverse piattaforme e sono stati fatti anche 2 remake: Pirates Gold del 1993 e Sid Meier's Pirates! del 2004.

Il gioco si svolge durante l'epoca d'oro dei pirati nel mar dei Caraibi, con il giocatore che impersona non un vero pirata, ma più un capitano corsaro di una delle 4 "grandi potenze" del XVII secolo: possiamo scegliere infatti di ricevere la lettera di corsa da Inghilterra, Spagna, Francia o Olanda.

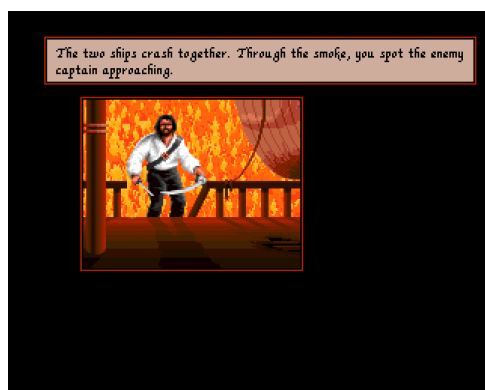
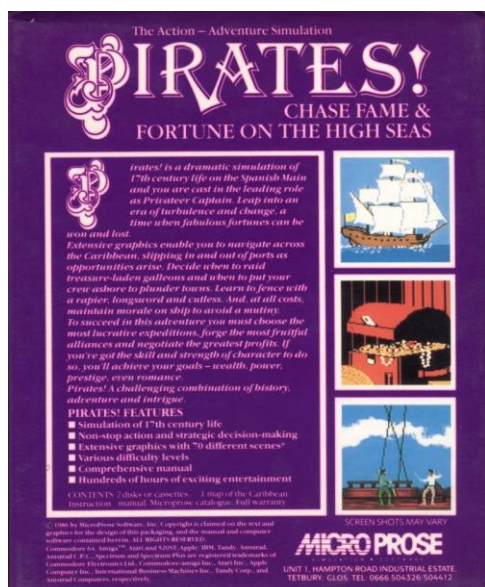
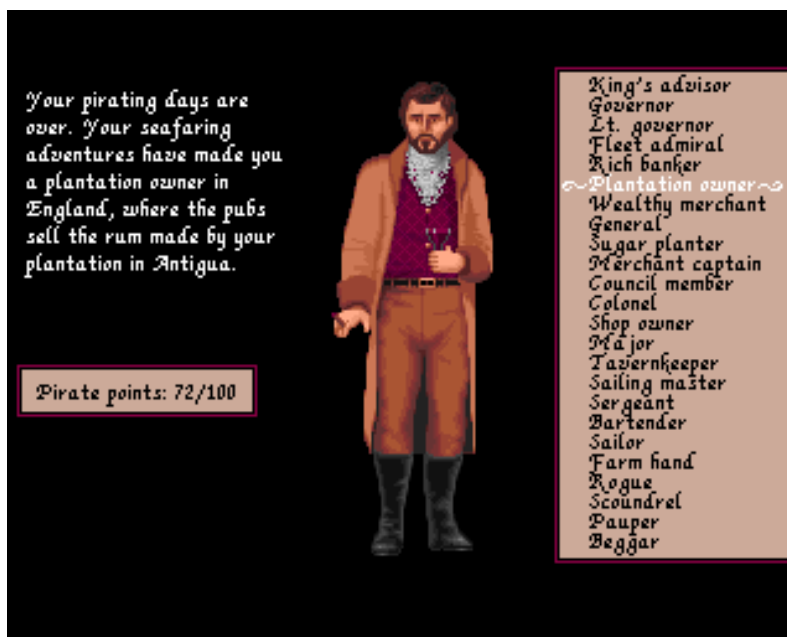
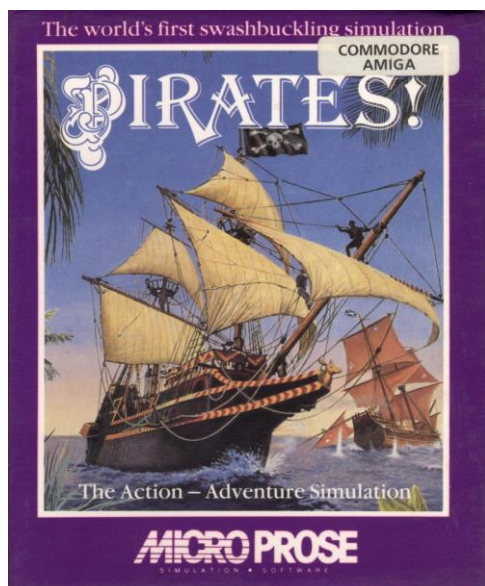
L'interfaccia del gioco è completamente in inglese, si inizia scegliendo, o meno, un particolare periodo storico, si sceglie poi la nazionalità del nostro giocatore e il nostro cognome, sceglie, però, che

Oltre che iniziare una nuova partita, o continuarne una salvata, è possibile scegliere di comandare una famosa spedizione, ognuna con un differente livello di difficoltà.

Se non si sceglie di comandare una famosa spedizione, il gioco si svolge in maniera abbastanza libera, alla ricerca di fama e fortuna, magari trovando la flotta del tesoro o il treno d'argento. Possiamo navigare nelle acque del mar dei Caraibi, visitare città dove incontrare commercianti o il governatore che potrebbe assegnarci una missione speciale.

The governor presents you with an elaborately written proclamation. "In recognition of your brave and loyal service to the English crown, I do gladly confer upon you the title of Ensign!"





Navigando sarà possibile incrociare altre navi e, a questo punto, inizieranno le scelte che decideranno la nostra fortuna nel gioco. Si può decidere se investigare e scoprire che tipo di nave sia oppure navigare via. Una volta scoperto il tipo di nave possiamo decidere se attaccarla oppure no. Se decidiamo di attaccarla inizia la fase di battaglia navale, potremo condurre la nostra nave (in caso di più navi, dobbiamo scegliere prima quale utilizzare) tramite il joystick, prestando attenzione alla direzione e intensità del vento, e poi colpire l'avversario con i cannoni, premendo il tasto di fuoco, ovviamente ci sarà un tempo di ricarica più o meno lungo tra una cannonata e l'altra. Una volta indebolito a sufficienza il "nemico" possiamo abbordarlo e iniziare il duello con il suo capitano. Anche in questo caso il duello si svolge tutto tramite il joystick e a seconda della posizione dell'asta sferreremo fendenti d'attacco o assumeremo posizioni di difesa, il numero dei componenti del nostro equipaggio che avremo al seguito influenzerà sulla forza dei colpi che riusciremo ad infliggere.

Una volta sconfitto il capitano avversario, potremmo decidere se affondare la sua nave o se incorporarla nella nostra flotta, in ogni caso dovremo decidere cosa fare del carico; in caso di capitano famoso potremmo tenerlo come ostaggio e, in alcuni casi, alcuni dei suoi sottoposti potrebbero chiedere di unirsi al nostro equipaggio. In caso di sconfitta verremo catturati e portati in prigione, o abbandonati su un'isola deserta.

Anche durante la navigazione bisogna prestare particolare attenzione, infatti c'è il rischio di finire su degli scogli e quindi danneggiare o distruggere la nostra nave e, se questa è l'ultima rimasta nella nostra flotta,

finiremmo su una spiaggia isolati per mesi prima che possano venire a salvarci... e la nostra reputazione ne risentirebbe negativamente.

Il gioco non ha una vera fine o un percorso prestabilito, sta a noi scegliere con attenzione quando e quali navi e città attaccare, quando commerciare, quando visitare governatori ed eventualmente accettaene le missioni, quando tradire un governatore e, soprattutto, quando dividere il bottino con l'equipaggio. I mesi e gli anni passano, la nostra reputazione muterà in conseguenza delle nostre azioni e potremmo anche guadagnare fama e pezzi di terra dove trascorrere la nostra vecchiaia... inutile dire che se troviamo il tesoro della flotta o il treno d'argento, la nostra vita migliorerà decisamente...

Insomma, per me uno dei migliori giochi a cui abbia mai giocato, con grafica e musica piacevole, ci ho perso veramente ore e mesi a giocarci...e ancora oggi rischio di fare lo stesso, purtroppo su un emulatore.

*"Veri pirati noi siamo
contro il sistema lottiamo
ci esercitiamo a scuola
a far la faccia dura
per fare più paura"*

di J.R.

Intervista a Gideon Zweijtzter

il designer della cartuccia 1541 Ultimate e della motherboard Ultimate64

di David La Monaca (Cercamon)

Uno dei lati più interessanti del mondo del retrocomputing è sicuramente quello di poter essere parte di una comunità molto vasta che, praticamente ovunque in Europa e nel mondo, conta un gran numero di appassionati. Oltre a retrogamer, collezionisti puri, nostalgici utenti di emulatori e retrocoder, possiamo contare anche su un numero corposo di esperti di elettronica e di design cui potremmo forse dare l'appellativo di "retrodesigner". Si tratta di progettisti di soluzioni ed accessori hardware per i nostri amati home computer, che fanno largo uso della tecnologia oggi disponibile. E così non è raro di questi tempi leggere di novità hardware come espansioni di memoria, acceleratori, periferiche di massa, nuovi case e persino vere e proprie nuove versioni dei computer completi.



Fig. 1 – Gideon Zweijtzter

Per la folta schiera degli amanti del Commodore 64, fra i prodotti più ambiti ci sono quelli ideati e realizzati dalla Gideon Logic di Amsterdam: la cartuccia 1541 Ultimate e la motherboard Ultimate64, entrambi a firma di Gideon Zweijtzter, un ingegnere elettronico di 44 anni con un cuore che batte decisamente a 8-bit. Gideon è una persona davvero gentile e disponibile con il quale abbiamo conversato di passato e presente del mondo C64 con un occhio al futuro prossimo del retrocomputing, per il

quale la Ultimate64, date le sue caratteristiche, rappresenta una concreta speranza. Di seguito il resoconto completo della nostra intervista esclusiva.

Salve Gideon e grazie mille per aver accettato l'invito per questa intervista esclusiva. Lo staff editoriale ed i lettori di RetroMagazine sono davvero entusiasti di avere l'opportunità di rivolgerti qualche domanda sulla tua esperienza di progettazione di due dei più famosi (se non i più famosi in assoluto, soprattutto in virtù delle loro indiscusse qualità funzionali) accessori per il Commodore 64: la cartuccia 1541 Ultimate e la motherboard Ultimate64.

Durante il 2017, stando a quanto si apprende dai tuoi siti web ufficiali, gli ultimi ritocchi al design della motherboard Ultimate64 sono stati completati. La tanto attesa scheda madre che sostituisce completamente quelle originali del C64 ed aggiunge molte funzionalità moderne (doppio SID, integrazione della 1541 Ultimate II+, piena compatibilità delle porte standard, uscita video HDMI ed RGB, porta Ethernet e modulo Wi-Fi, ecc.) è finalmente entrata in produzione ed i primi lotti sono stati spediti agli utenti finali già nella prima parte di quest'anno. Il feedback dei primi utenti della Ultimate64 è assolutamente positivo mentre Gideon, con l'aiuto della comunità, continua a lavorare all'aggiornamento del firmware e a sveltire l'intero processo di produzione.

Moltissimi fan del Commodore 64 in Italia e nel mondo conoscono bene i tuoi prodotti, ma sono abbastanza sicuro che non sanno come tutto è cominciato.

DLM: Allora, Gideon, prima di passare alle domande, che ne dici di presentarti brevemente e dirci qualcosa di te e della tua storia personale e professionale?

GZ: Ciao David! Grazie a voi per l'invito. Parlare un po' di me? Certo... Sono nato ad Amsterdam nel 1974 in una famiglia molto stabile e con un fratello maggiore. Fin da piccolo sono sempre stato interessato ai

tecnicismi. Prima dell'avvento degli home computer, me ne stavo sempre a giocare col mio Lego tecnico, anche se mi piaceva anche correre in bici in giro per il quartiere. Ho giocato spesso con i circuiti composti da interruttori, piccoli motori e lampadine, ma purtroppo non c'era nessuno nel vicinato che avesse conoscenze di elettronica. Ho fatto un normale corso di studi e dalla scuola secondaria sono passato all'università tecnica TU Delft, dove ho studiato Ingegneria Elettronica.

DLM: Suppongo che tu sia sempre stato un fan ed un utente di computer fin da bambino. Che cosa ti ha portato nel campo dell'informatica e qual è stata la tua prima esperienza con un computer? Scommetto che il primo su cui hai messo le mani fosse un Commodore 64.

GZ: Ero abbastanza piccolo quando in casa prendemmo una console Atari 2600. In realtà fu mio fratello a cominciare ad interessarsi ai computer e alle possibilità di programmarli. All'epoca c'era un interprete BASIC per l'A2600, ma alla fine comprò un Sinclair ZX81. Non mi era permesso di toccarlo, ma qualche volta mi intrufolavo nella sua camera quando lui non c'era e facevo qualche prova, ma da bambino e senza alcun aiuto, non andavo molto lontano. Qualche tempo dopo, mio fratello ottenne un Commodore 64 dai nostri genitori e divenne tutto fantastico. Quel computer era così popolare in quel periodo! I computer club, gli incontri, i party per copiare giochi e programmi! Mio fratello mi contagiò con la sua curiosità per la programmazione e, anche se non voleva che lo disturbassi, potevo starmene seduto sul pavimento fra la sua enorme scrivania e la vecchia TV a colori che si trovava sopra un altro tavolo lì davanti. Finché me ne stavo lì zitto e buono, potevo restare a guardare quello che faceva. E così vidi codice BASIC, listati per assembler, ecc. Ad un certo punto sapevo dirgli da dietro la scrivania quando dimenticava un'istruzione... Ma soltanto quando raggiunsi l'età di 11 anni, ebbi un Commodore 64 tutto per me.

DLM: Come hai cominciato a sperimentare sul C64, a parte giocarci? Ti sei interessato rapidamente alla programmazione e alla scoperta del funzionamento interno della macchina?

GZ: In realtà non ho mai giocato molto con i videogame. Con qualche eccezione, come The Great Giana Sisters, ma in generale non passavo molto tempo sui giochi. Così, mentre mio fratello si concentrava molto sul software, il mio interesse per l'hardware cresceva. Ad un certo punto delle mie sperimentazioni costruii un semplice termometro utilizzando una termistore NTC sulla porta paddle del C64. Avevo ancora bisogno di mio fratello per via delle sue maggiori abilità matematiche per calcolare la curva di conversione. Al computer club di Amsterdam, di solito passavo il tempo vicino al banco delle riparazioni, dove potevo osservare alcuni ragazzi che dissaldavano e sostituivano i chip dei C64 rotti. Secondo quanto dice mia madre, avevo appeso alla parete della mia piccola camera l'intero schema elettronico del 64. Ma in tutta onestà non me lo ricordo affatto.

DLM: Dopo il C64, sei passato come molti di noi direttamente al mondo dei PC? Hai mantenuto il tuo C64 sulla scrivania accanto al PC?

GZ: No, in effetti no. Con la tua domanda stiamo saltando parecchi anni avanti. Il mio interesse per il C64 svanì quando uscì l'Amiga 500 e più tardi il PC. In effetti il mio primo PC fu un Pentium 120 MHz, quindi puoi facilmente immaginare che ho cercato di resistere all'avvento dei PC per un bel po' di tempo... Ma l'amore per il C64 non è mai davvero tramontato, smisi soltanto di usarlo per un po'. Non ho neppure mai fatto parte di un demo-group o della "scena C64" in generale e non sono mai stato un vero e proprio coder... Quindi da allora non ho mai praticamente avuto bisogno di una periferica per il C64 più avanzata, come le SD2IEC che cominciarono a girare più tardi negli anni.

DLM: Qual è stata l'ispirazione iniziale che ti ha spinto sulla strada della progettazione della prima versione della cartuccia 1541 Ultimate e quando hai cominciato?



Fig. 2 – La cartuccia 1541 Ultimate II+

GZ: La prima versione della 1541 Ultimate è stata completata nel 2007. Tutto cominciò con alcune implementazioni del 6502 mentre imparavo e facevo esperienza con il design di circuiti FPGA utilizzando VHDL. E questo accadeva nel 2001 o giù di lì. Stavano succedendo molte cose a quell'epoca. Per esempio, Jeri Ellsworth stava lavorando alla sua C-One, che in seguito divenne il famoso C64-DTV, se non sbaglio. Io avevo comunque già fatto molto sul C64 con FPGA a quel tempo, ma non vedevo la ragione per ripetere un'esperienza simile. Volevo fare qualcosa di nuovo. Così pensai che avrei costruito un'alternativa moderna del floppy drive. In uno degli incontri del club di Maarssen, feci una dimostrazione del mio primo prototipo configurato su una scheda Xilinx Spartan 3. C'era bisogno di un laptop o di un PC per scaricare l'immagine di un floppy via Ethernet nella memoria della scheda, dopodiché la scheda si comportava esattamente come un floppy drive. Nessun menu, nessun'altra emulazione, soltanto le funzioni di un drive. In seguito, grazie ad una conversazione con uno dei miei colleghi di lavoro, venne fuori l'idea di costruirne una versione su cartuccia, in modo tale che il (chip video) VIC potesse essere utilizzato per visualizzare un'interfaccia utente. Quest'idea si concretizzò nel 2007.

DLM: Hai progettato la parte hardware ed il software/firmware per la cartuccia 1541 Ultimate in totale solitudine?

GZ: Sì, in pratica sì. Ci sono stati alcuni importanti contributi di altre persone nel corso degli anni, però. Ma in sostanza, il

design dell'hardware, della FPGA, del firmware e del framework necessario per lo sviluppo della parte software (dal momento che non ho fatto uso delle librerie standard C++) è tutta opera mia.

DLM: Qual era il sistema che hai utilizzato per sviluppare e testare le prime versioni della cartuccia?

GZ: Soltanto un PC ed un Commodore 64. E sì, il mio sistema non includeva affatto un drive 1541! In seguito compresi che ciò di cui disponevo non era sufficiente per fare i test necessari, ma non avevo altro hardware, quindi feci visita ad alcuni amici del Commodore club che possedevano collezioni impressionanti di macchine e periferiche con cui effettuare i test di compatibilità. E infatti, proprio in quell'occasione scoprii che il primissimo prototipo di 1541 Ultimate (versione cartuccia) non era molto compatibile. Questo mi fu molto utile per modificare il progetto prima di mandare la prima versione della scheda nella fase di produzione.

DLM: Hai frequentato dei corsi specifici per cominciare nel campo dell'elettronica e se sì, quanto tempo hai investito per imparare a livello pratico? Oppure come molti altri progettisti/programmatore dei primi anni Ottanta sei stato un autodidatta?

GZ: Molte cose le ho imparate da solo, anche se studiare all'Università Tecnica di Delft mi ha fatto comprendere le basi e molti altri concetti avanzati. Ma per essere onesti, credo

di aver imparato molto al lavoro che ho iniziato subito dopo la fine degli studi. Ho cominciato a lavorare come progettista junior presso un'azienda chiamata Technolution B.V. e lì ho imparato la maggior parte delle conoscenze pratiche che possiedo oggi, soprattutto in termini di design di circuiti elettronici. Curiosamente, ho finito per portare all'interno dell'azienda le mie conoscenze riguardo al design di circuiti basati su FPGA come se fossi uno dei fondatori di questa tecnologia!

DLM: Come si è articolato il processo di sviluppo? Ad esempio, hai prima abbozzato i concetti base, poi hai progettato il circuito principale ed il firmware, ecc.? Ancora oggi intraprendi un nuovo progetto allo stesso modo?

GZ: Oooh, questa è una domanda piuttosto difficile a cui rispondere. Poiché ho sempre guardato a queste attività come ad un hobby, per lo più lascio che tutto accada da sé. Sono il tipo di designer che fa un sacco di lavoro come se si trattasse di un 'processo in background'. Non sono il classico ingegnere molto strutturato che segue un metodo preciso, passo dopo passo (al lavoro ho fatto impazzire parecchi project manager, dal momento che non mi vedevano da subito all'opera sui nuovi compiti che mi avevano assegnato...).

Fondamentalmente lavoro con le iterazioni. Ma tutto accade soprattutto nella mia mente. A volte sotto la doccia, a volte quando guidavo. Una volta che il progetto la parte essenziale di esso 'suona bene' nella mia testa, comincio a provare qualche implementazione pratica. E a volte, dopo un esperimento, mi rendo conto che la cosa non 'suona più bene'. Non ho paura di buttare via tutto il lavoro fatto o parte

di esso e ricominciare da capo. Naturalmente, sempre tenendo in conto della lezione imparata nella fase precedente.

DLM: A proposito dei tuoi progetti più importanti (la cartuccia 1541 Ultimate e la nuovissima motherboard Ultimate64), quale sfida tecnica ti ha fornito la sensazione di maggiore soddisfazione?

GZ: Sì, beh se ci limitiamo a delle 'sfide tecniche', di sicuro direi scovare e risolvere dei bug davvero difficili da trovare... Sai, quelli davvero fastidiosi e complessi che spesso costringono altri ad abbandonare i loro progetti...? Esatto, proprio quelli!

Al secondo posto metterei il momento in cui dai corrente ad una nuova scheda e tutto funziona alla grande fin da subito (e questo non accade di rado nel mio caso... [faccia compiaciuta]).

DLM: Quali sono stati i maggiori ostacoli tecnici o logistici che hai dovuto abbattere durante le varie fasi di realizzazioni (progettazione, produzione, test, vendita) della 1541 Ultimate o della Ultimate64?

GZ: Ostacoli... [fammi pensare]... Beh, dipende anche un po' da cosa intendi per ostacoli. Molte fasi del lavoro sono tuttora processi che implicano soltanto un gran dispendio di tempo. Ma in effetti ci sono stati parecchi ostacoli e momenti critici da superare. Penso che nel caso della 1541 Ultimate sia stato il dover creare un'interfaccia utente facile da usare senza poter disporre di un framework di sviluppo standard; ho dovuto costruire tutto da zero.

Su una piattaforma embedded come la Ultimate non si possono usare framework standard come quelli comunemente

disponibili per Java o C#, quindi bisogna costruirsi uno praticamente dal nulla.

Uhm, e poi... vediamo... Un altro ostacolo è stato certamente quello di dover sviluppare un sistema di test di fabbricazione per la Ultimate II+. Ha richiesto parecchio tempo, ma d'altro canto adesso me ne fa risparmiare un sacco. Un altro momento critico è stato quello di passare ad un sistema automatizzato di acquisti via web invece di continuare a raccogliere ordini ed elaborarli manualmente.

DLM: Qual è il tuo gioco preferito per C64? Trovi ancora il tempo per giocarci di tanto in tanto?

GZ: Il gioco è sicuramente The Great Giana Sisters... e mmmh, assolutamente no, niente tempo per giocare! :-)

DLM: Immagino che tu possieda una buona collezione di C64 (cioè un po' tutte le versioni, dal classico "biscottone", con le varie versioni di motherboard ASSY, al C64c, al C64g fino al C128) per poter effettuare i test dei tuoi prodotti Ultimate. Sei anche un collezionista di retrocomputer in genere, non soltanto a marchio Commodore?

GZ: Mia moglie mi ucciderebbe se davvero ne collezionassi ancora. Possiedo solo mainboard C64 funzionanti e di queste ne utilizzo una soltanto in una scocca del tipo C64c. Questa è sempre la stessa macchina che ho usato per testare oltre 3.000 cartucce 1541 Ultimate nel corso degli anni. Il pulsante di accensione e la porta cartridge sono ormai comprensibilmente un po' usurati. Ho anche un C128 ed un C128D ma non li uso mai. Ed ho anche parecchi floppy drive.

DLM: Riesci ad immaginare di calcolare quante ore hai passato a progettare e a lavorare attivamente sulle varie versioni di 1541 Ultimate? E che mi dici della Ultimate64?

GZ: E' molto difficile fare questo calcolo. Come detto, molte attività di progettazione spesso sono effettuate in background nella mia mente. Se contassi solo le ore che passo



Fig. 3 – La mainboard Ultimate64 in tutto il suo splendore

davanti al PC, verrebbe fuori una cifra falsamente bassa. Quello che posso dire con certezza, però, è che il design delle parti hardware, degli schemi dei circuiti e di roba simile, di solito non richiede molto tempo. Penso di aver creato il design della motherboard U64 in circa tre settimane, ma naturalmente soltanto durante la sera ed i fine settimana. La realizzazione dei diagrammi tecnici ha richiesto una quantità di tempo simile. La maggior parte del tempo speso nei tecnicismi serve per il design, la messa in opera, il debug e l'implementazione del firmware della FPGA.

Rileggendo la tua domanda, comprendo che stai mettendo molto in evidenza gli aspetti tecnici, ma posso assicurarti che anche tutti i compiti amministrativi, inclusi gli ordini di spedizione ed il supporto (le risposte alle e-mail degli utenti), purtroppo portano via gran parte del tempo.

DLM: Hai mai lavorato (o pensi di lavorare) su altri progetti che riguardano il C64 o anche altre macchine a 8/16-bit?

GZ: Al momento proprio no..., non ne avrei il tempo! :-)

DLM: Quante persone attualmente lavorano alla produzione, al testing ed alla vendita dei due principali prodotti di Gideon Logic, la tua ditta? Hai mai lavorato in team o ti sei semplicemente consultato con altri esperti di elettronica per raggiungere un risultato particolare o per risolvere un bug fastidioso?

GZ: La produzione dei pezzi è ovviamente affidata esternamente ad alcune aziende specializzate. Il test di produzione della Ultimate-II+ viene eseguito anch'esso in fabbrica. Quello per la U64 al momento non è previsto ma lo sarà presto per poter accelerare tutto il processo. Per quanto riguarda l'assemblaggio delle Ultimate-II+ nei case in plastica, quest'operazione è spesso compiuta da mia moglie... quando ne ha voglia. Lei riveste anche un ruolo importante per la preparazione degli ordini da spedire. Tutto il resto lo faccio io, non ci sono collaboratori in questo momento. Se si possa continuare in questo modo è discutibile. Penso di aver bisogno d'aiuto esterno



Fig. 4 – Un recente meeting del Commodore Club di Maarssen (Paesi Bassi)

nell'immediato futuro vista la quantità di Ultimate64 che sono state ordinate!

Per quanto riguarda l'aspetto puramente tecnico, qualche volta parlo con i miei colleghi a proposito di certi bug e naturalmente faccio buon uso delle loro risposte e delle richieste che arrivano dalla comunità. Ne fanno parte alcuni ragazzi molto bravi e capaci e anch'essi a volte mi aiutano a trovare e risolvere i bug. E questo col tempo porta agli aggiornamenti effettivi utili per tutti gli utenti dei prodotti Ultimate. Per arrivare a realizzare una particolare funzione tecnica, spesso applico dei modelli che sommamente acquisisco o imparo da altri progetti su cui lavoro.

DLM: Se ripensi al tempo in cui tutto iniziò, c'è qualcosa di cui ti penti nel design dei prodotti Ultimate o di qualunque altro dettaglio? Faresti qualcosa in modo diverso ora, se potessi?

GZ: Mi rammarico soprattutto di non aver portato ad un livello di produzione il codice FPGA per C64 che composi anni prima di lavorare alla realizzazione della Ultimate64. Infatti avevo messo insieme un C64 completo tramite FPGA già nel 2011. All'epoca credevo che nessuno fosse interessato ad acquistare una scheda madre C64 basata su FPGA, perché le macchine originali Commodore si potevano ancora trovare a prezzi irrisori o in mancanza di queste i potenziali utenti avrebbero comunque usato uno dei tanti emulatori disponibili.

Rimorsi a proposito di altri aspetti? Beh, con lo sguardo al passato, di molte cose ci si può

pentire, ma penso che non sia giusto guardare alle cose in questo modo, perché come persona e come ingegnere s'impara tutto strada facendo. Se pensi che le cose debbano essere modificate, allora hai sempre la libertà di farlo in un secondo momento. Credo che questo sia uno dei buoni motivi per gestire in completa autonomia le proprie creazioni. Ma suppongo che lo stesso principio si possa applicare a molte altre cose della vita, non credi...?

DLM: Sono certo che hai lavorato molto duramente su entrambi i progetti durante gli ultimi anni ma anche che ti sei divertito parecchio durante tutto questo tempo. Qual è stato il momento più strano e divertente o l'episodio più buffo che ti è capitato in questa tua avventura?

GZ: Oh, mi sono davvero divertito a fare tutto questo! In termini tecnici, la parte più divertente è stata programmare la FPGA, poi l'hardware stesso e infine il firmware.

Credo che un momento davvero buffo sia stato quando mi sono davvero reso conto della mia grande ingenuità. Durante l'intero processo di sviluppo della 1541 Ultimate, non ho MAI pensato di farne un prodotto da commercializzare. O meglio, diciamo che questo non era il mio obiettivo: fino ad allora era stato puramente un hobby per me. A dire la verità fu uno svedese della scena C64, TwoFlower, che capitò in visita al Commodore Club di Maarssen proprio il giorno in cui io stavo facendo la dimostrazione di una cartuccia che incorporava un floppy drive. Mi disse che avrei

dovuto metterlo in produzione, ma io ero esitante e pensavo che non sarebbe stato realmente fattibile. Così mi chiese quanti pezzi era necessario produrre in base alle richieste ed io balbettai qualcosa come "Forse 40 o 50?". Allora lui sorrise e disse: "Allora fallo e basta... Ti assicuro che io da solo ne venderò oltre 40 in Svezia!" Ed è così che cominciò la mia avventura!

DLM: Gideon, grazie ancora per il tuo tempo. La nostra intervista termina qui, ma vorresti aggiungere o dire qualcosa ai nostri lettori?

GZ: Sì, c'è una cosa importante che vorrei dire. Vorrei esprimere un enorme 'grazie' all'affettuosa comunità Commodore. Uno degli aspetti più gratificanti dei miei progetti è il grandioso feedback, tutti i commenti positivi che ricevo quotidianamente. In breve: senza tutti voi, ragazzi, non sarei mai stato capace di compiere tutto questo. Vi ringrazio.

©2018 RetroMagazine / DLM & GZ



Fig. 5 – Un primo piano di Gideon sorridente

Riferimenti

Sito ufficiale della cartuccia 1541 Ultimate
<http://www.1541Ultimate.net>

Sito ufficiale della motherboard Ultimate64
<http://www.ultimate64.com>

The Great Giana Sisters – Il gioco preferito da Gideon è un famosissimo platform game del 1987 uscito per C64 da Rainbow Arts. Il

titolo era spudoratamente, volutamente e ironicamente un clone del Super Mario Bros di Nintendo che all'epoca spopolava sulla console NES. Tanto che la Rainbow Arts fu costretta a ritirare il gioco dal mercato dopo poche settimane sotto la minaccia di causa legale da parte di Nintendo.

https://it.wikipedia.org/wiki/The_Great_Giana_Sisters

VHDL – Acronimo di VHSIC Hardware Description Language. È, insieme a Verilog, il linguaggio più usato per la programmazione di sistemi e circuiti elettronici digitali.

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

FPGA – Acronimo di Field Programmable Gate Array, un circuito integrato le cui funzioni sono programmabili mediante uno dei linguaggi di descrizione dell'hardware (Verilog o VHDL)

https://it.wikipedia.org/wiki/Field_Programmable_Gate_Array

Spartan 3 – Una famiglia di circuiti FPGA a basso costo, pronte da programmare e prodotte da Xilinx

<https://www.xilinx.com/products/silicon-devices/fpga/spartan-3.html>

C-One – Un computer a singola scheda (SBC, Single Board Computer) costruito nel 2002 da Jeri Ellsworth e da Jens Schönfeld per emulare un intero C64. Fu in seguito impiegato per produrre l'originale console / joystick C64-DTV.

<https://en.wikipedia.org/wiki/C-One>

Jeri Ellsworth – Inventrice e imprenditrice nel campo dell'elettronica. Nel 2004 mise a punto la console C64-DTV, un emulatore C64 integrato in un joystick con 30 giochi inclusi da collegare direttamente alla TV

https://en.wikipedia.org/wiki/Jeri_Ellsworth

TwoFlower – Grafico e musicista svedese della scena 64, già fondatore del gruppo Twilight, attualmente attivo con i Triad

<https://csdb.dk/scener/?id=8074>

La cartuccia Ultimate II+ in breve

- Emulazione completa di 2 drive C1541
- Legge file .D64 and .G64 su chiavetta USB
- ID dispositivo configurabile
- ROM configurabile (1541 / 1541C / 1541II oppure custom)

- Esp. di memoria RAMBoard fino a 40KB
- Speaker incorporato per i suoni del drive
- Supporto di dispositivi USB (chiavette, lettori di memory card e persino HardDisk)
- Porta Ethernet 100Mbps (Telnet o FTP)
- Supporto FAT/FAT32, ISO9660
- Real Time Clock con batteria di backup
- Modulo d'interfaccia UltimateDOS
- Implementazione Dual SID, player incorporato, suona su SID reale o emulato
- Emulazione cartucce C64 (Final Cartridge III, Action Replay, Retro Replay, Epyx, ecc.)
- Emulazione Commodore 1750/1764 REU
- Menu di configurazione
- Emulazione Tape supporto .TAP files
- Firmware aggiornabile

Specifiche hardware

- Dimensioni scheda: 62 x 90 mm
- Case in plastic robusta
- FPGA 1M-gate
- 6-pin DIN connector (IEC)
- Flash ROM 5 MB
- 64 MB DDR2 SDRAM
- 3 porte USB disponibili
- 1 porta Ethernet 100Mbps
- Uscita audio stereo
- Ingresso audio stereo (non in uso)
- Porte USB 3.0 port per dispositivi di massa o per adattatore tape (emulazione C2N) o altre funzioni disponibili in futuro.

La Ultimate64 in breve

- Motherboard di un completo C64
- Implementazione HW FPGA di un C64
- Tutte le funzioni di una U2+ incluse
- Compatibilità prossima al 100%
- Firmware aggiornabile nel tempo
- Alimentatore 12V DC incluso
- Supporto totale periferiche standard C64
- Supporto SID 6581 e 8580
- Uscita video standard DIN e HDMI (RGB)
- Supporto PAL, Luma Chroma separati
- Supporto NTSC disponibile a breve
- User port tramite pin-header su scheda
- Porta Ethernet 100Mbps (U2+)
- 2 porte USB 3.0 (U2+)
- Modulo Wi-Fi on-board disponibile
- Pulsante reset/menu U2+

Sito Web: CPCrulez.fr - <https://cpcrulez.fr>

di Francesco Fiorentini

I nostri lettori sono la nostra fonte principale di ispirazione, perché non si limitano soltanto a leggere la rivista, ma ci forniscono anche utili suggerimenti per ampliarla! E' capitato che qualcuno ci abbia suggerito di recensire, oltre a giochi ed applicazioni, anche i siti web dedicati alla nostra passione. Ovviamente ci è sembrata subito un'ottima idea ed abbiamo deciso di provare a cimentarci in questo nuovo esercizio. Se l'esperimento dovesse piacere, la recensione dei siti web potrebbe diventare una rubrica fissa all'interno di RetroMagazine.

Senza ulteriori esitazioni andiamo quindi a parlare del sito **CPCrulez.fr**.

Come facilmente intuibile dal nome stesso, trattasi di un sito francese interamente dedicato al computer **Amstrad CPC**. L'Amstrad CPC è una serie di home computer a 8 bit prodotta dalla Amstrad tra la metà degli anni '80 e quella degli anni '90. L'acronimo CPC, *Colour Personal Computer*, sta ad indicare la possibilità del computer di generare colori. Non che gli altri computer dell'epoca fossero in bianco e nero, ma a differenza degli altri computer del periodo, i computer Amstrad erano venduti come sistemi completi che includevano, oltre a lettori di cassette o floppy anche un monitor a colori.

Dal punto di vista estetico dobbiamo ammettere che il sito non si presenta benissimo, la pagina iniziale ci accoglie con una lunga lista di aggiornamenti, utili sicuramente, ma un po' caotici per come sono presentati. Forse il lato estetico andrebbe un po' rivisto, soprattutto per quanto riguarda la homepage del sito, vero e proprio biglietto da visita di ogni sito web.

Ma noi non siamo superficiali e non ci lasciamo scoraggiare ne' dalla mancanza di effetti speciali e colori ultravioletti, ne' dalla lingua francese. Eh si', trattandosi di un sito francese, il webmaster ha optato per scrivere principalmente nel suo idioma e noi non vogliamo assolutamente fargliene una colpa.

Come dicevo poco prima, nonostante il comparto grafico non sia di prim'ordine, nella parte sinistra del sito è presente un banner con un comodo menu', in inglese, che ci permette di navigare agevolmente in tutte le pagine del sito. Generalmente i siti dedicati ad uno specifico computer cercano di coprire a 360 gradi ogni aspetto della macchina e CPCrulez non fa eccezione.

Si parte dalla demoscene per arrivare alle utility, passando per i giochi, l'hardware, la programmazione e gli emulatori. Il tutto suddiviso in comode sezioni ricche di software, articoli, interviste e contenuti di vario genere.

Ovviamente la parte del leone la fa la sezione **Giochi**, con più di **7700** titoli catalogati in 9 sottosezioni e ricercabili tramite il comodo motore di ricerca presente sul sito. Purtroppo non sono un esperto del computer Amstrad CPC, ma credo di poter dire senza timore di smentita, che si tratta di una delle collezioni più complete dedicate a questa macchina. Quasi tutti i giochi sono corredati da una breve descrizione ed alcune schermate del gioco e per alcuni di essi il sito mette a disposizione anche contenuto aggiuntivo (manuali, inserti pubblicitari, recensioni, cover...).

La sezione dedicata alle **Utility** è curata esattamente come quella dei giochi. Ovviamente il numero delle applicazioni è minore, si tratta 'soltanto' di circa **3600** titoli, ma anche qui abbiamo la possibilità di ricercare il software tramite la funzione di ricerca ed usufruire di tutte le informazioni a corredo come per i giochi. Sicuramente un lavoro certosino che merita di essere segnalato e riconosciuto. Personalmente trovo le utility una parte fondamentale di ogni macchina e per alcuni versi anche più interessanti dei giochi stessi. Come dicevo prima non sono un esperto di questo computer in particolare, ma vi prometto che cercherò di studiarlo e di recensire qualcuna delle utility più interessanti (come fatto per datatool del C64 nel numero scorso).

Passiamo adesso alla sezione dedicata alla **Demoscene**. Va da sé che la stessa cura prodigata per i giochi e le utility è stata applicata anche nella ricerca, catalogazione e recensione delle demo. Onestamente non credevo che esistessero così tante demo programmate sull'Amstrad, ma il sito ne raccoglie ben più di **2700**! Di nuovo corredate da immagini, descrizione e persino interviste agli autori. Che dire... chapeau!

Anche le sezioni dedicate all'**Hardware** ed al **Coding dello Z80** sono ricche di contenuti e di materiale interessante. Peccato che lo stesso sia quasi completamente in lingua francese, e



quindi poco fruibile per chi non si trova a proprio agio con l'idioma dei cugini d'oltralpe. Comunque si tratta di materiale ormai facilmente reperibile su Internet con una semplice ricerca con Google, quindi non mi soffermerò oltre.

L'unica nota dolente del sito è la sezione dedicata agli **Emulatori**; sicuramente ricca e ben fornita, come nello standard del resto del sito, ma questa volta pesantemente penalizzata dalla grafica caotica del motore di rendering adottato. Per l'amor di Dio, c'è di tutto e di più, ma occorre guardare attentamente altrimenti si rischia di perdersi nelle sottosezioni. Sicuramente da rivedere.

In conclusione ci troviamo di fronte ad un lavoro ben fatto. L'amore del webmaster per l'Amstrad CPC è evidente, come evidente è la sua missione per preservare il software sviluppato e tuttora in sviluppo per questo computer. Condividendo la passione e la missione del sito non potevamo esimerci dal darle risalto nelle pagine della nostra rivista. Dovete visitarlo se siete amanti dell'Amstrad CPC ma ancor di più se non conoscete affatto questa macchina. Sicuramente un'ottima opportunità per riscoprire un computer forse un po' troppo sottovalutato in Italia!

URL: <https://cpcrulez.fr>

Lingua: Francese

Soggetto: Amstrad CPC

Contenuto: Giochi, utility, demo...

Status: Attivo

Aggiornamenti: Frequenti

Aspetto esteriore: Rivedibile

Valutazione: 8/10

In Breve: un ottimo sito, ricco di contenuti, dedicato all'**Amstrad CPC**. Peccato per la veste grafica rivedibile.

Chiusura e chiamata alle armi...

di Francesco Fiorentini

Piu' lavoro per augurarvi Buone Vacanze!

Come già avrete avuto modo di intuire dall'editoriale a firma di Starfox, questo che avete appena terminato di leggere e' un numero speciale per la nostra redazione. Ogni redattore ha profuso un impegno addizionale per fare in modo che questa uscita di RetroMagazine fosse sufficiente a garantire abbastanza materiale da leggere per i mesi piu' caldi dell'anno. Ed a mio modesto parere il risultato ha premiato lo sforzo profuso, ma come sempre e' il parere dei lettori che riveste maggiore importanza quindi aspetto fiducioso i vostri feedback.

Prima pero' di lasciarvi alle meritate vacanze, voglio porre l'attenzione su una novita' che abbiamo introdotto in questo numero.

Come anticipato qualche numero fa, esattamente nel numero 6, finalmente ha visto la luce la rubrica **RetroGiochiAmo** (sì il nome ha subito un cambiamento da quanto annunciato) e per farci perdonare dell'attesa, questo numero abbiamo deciso di parlare di ben 2 giochi: **Grimrock - The series** e **Minesweeper**. La scelta dei giochi non e' stata banale e con questi due titoli abbiamo voluto sottolineare il messaggio di un ponte temporale tra presente e passato. Grimrock e' infatti un titolo moderno ma con un gameplay classico, mentre Minesweeper e' un gioco classico (la prima versione risale addirittura agli anni '60) ma con un gameplay moderno (e' proprio di questi giorni la pubblicazione su Google Play di Minesweeping (ad-free) - classic minesweeper game).

We want you!

Noi ci siamo divertiti a rigiocare i giochi per scrivere gli articoli e forti di cio', vogliamo gettare l'esca. Perche' non partecipate alla stesura di una di queste rubriche? Se siete appassionati di un gioco e volete coinvolgere altri retrogiocatori, perche' non vi proponete per scrivere un articolo per RetroGiochiAmo? Dovete solo scegliere un gioco e contattarci, vi aiuteremo noi alla stesura dell'articolo. Non siate timidi!

We want you bis!

Chi ci segue da tempo sa che abbiamo un sito web di riferimento www.retromagazine.net e si sara' anche accorto che il sito e' povero dal punto di vista grafico e di contenuti. Ci piacerebbe tanto migliorare il suo aspetto e caricare del contenuto, ma il tempo e' quello che e', soprattutto per noi che scriviamo articoli ed impaginiamo la rivista. Siamo quindi alla ricerca di un **webmaster** che si prenda l'impegno di gestire il sito. Se avete voglia di partecipare al nostro progetto ma non avete modo di scrivere articoli questa potrebbe essere la vostra occasione. Come ho sempre detto la rivista e' di tutti, quindi pensateci sotto l'ombrellone e fatevi avanti!

Chiudo, come ormai di consueto, con i **ringraziamenti** a tutti i **gruppi Facebook** ed al sito **OldGamesItalia** che ci aiutano a condividere la rivista ad ogni uscita. A nome di tutta la redazione rinnovo l'invito a chi volesse dare il suo contributo, in ogni modo, a farsi avanti.

Al prossimo numero di RetroMagazine!

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 8

Direttore Responsabile
Francesco Fiorentini

Immagine di copertina
Flavio Soldani

Luglio/Agosto 2018