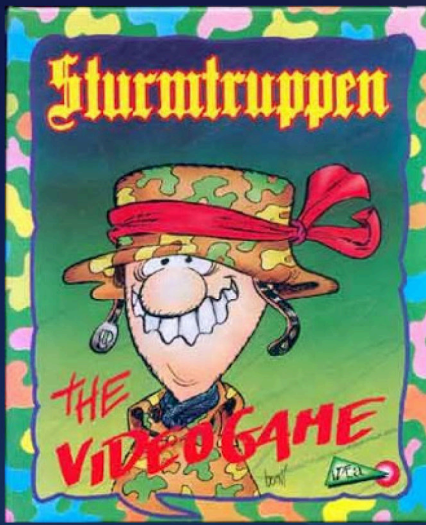




World RetroMagazine

future days are back

Numero 24  Anno 4 - Luglio 2020 - WWW.RETROMAGAZINE.NET - Pubblicazione gratuita



Recensione: Sturmtruppen (Amiga)



Ritorno al passato: Windows 2000



Recensione: Cyborg Z (MSX)



Game & Watch:
Intervista a Luca Antignano
"MADrigal's simulators"



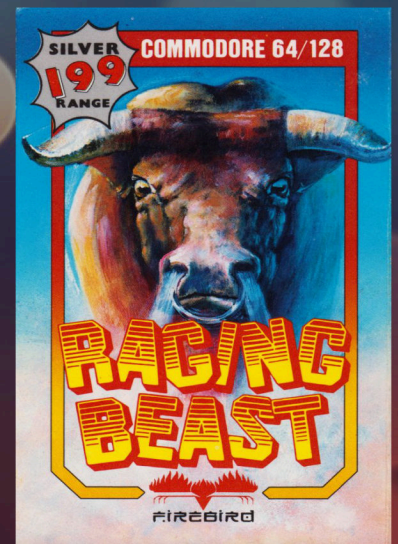
Recensione: Moon Cresta (Arcade)



Recensione: Wonderboy the dragon's trap (1989 & remake)



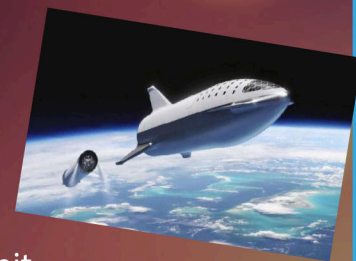
Anteprima di Ellenica (Megadrive)



Recensione:
Raging Beast (C64)



- Abbreviazioni per appunti sull'uso di una Interfaccia Grafica
- LM80C Color Computer: un computer autocostruito nel 2019 basato sullo Z80 - parte 4
- Uno splash screen per Amstrad CPC in formato SCR
- L'Olivetti M20 e la storia di un sito web
- Speciale Scacchi: il 1° torneo RMW per home computer a 8-bit



Tempo di vacanze, tempo di ricordi...

L'estate, con il suo caldo torrido e le notti bollenti, ha fatto nuovamente visita a tutti noi. Probabilmente mai come quest'anno il caldo e le temperature sopra la media erano attesi con tanta trepidazione. Dopo un inverno ed una primavera orribili, questa estate non è soltanto sinonimo di vacanze, ma anche di un lento ritorno alla normalità per molti di noi.

Chi vi scrive vive all'estero ormai da qualche anno e l'estate è uno dei momenti più attesi per poter far ritorno in Italia e riabbracciare amici e parenti. Potete immaginare quindi con quanto ardore abbia atteso quest'anno la possibilità di poter nuovamente viaggiare e tornare nei luoghi della mia giovinezza. Per chi vive lontano dal paese in cui è nato e cresciuto, la possibilità di farvi ritorno una o due volte all'anno è come sfogliare un album dei ricordi. Ritrovare luoghi e persone che non vedi da parecchio tempo, accende dentro di te la voglia di conoscere cosa è successo nel frattempo e parimenti la possibilità di fare un paragone con quello che è stato e quello che invece è... La cosa più divertente, almeno per me, è il notare come i piccoli cambiamenti siano molto più impressionanti di quelli grandi. Forse perché non sono così evidenti e quando te ne accorgi, scatta immediato il paragone.

Qualcuno di voi potrebbe chiedersi quale sia l'attinenza di questo editoriale con la nostra comune passione. E' presto detto! Causa la complessità del trasloco e la cronica mancanza di spazio nelle nostre case (quantomeno nella mia...) ho dovuto, almeno temporaneamente, rinunciare a trasferire una sostanziosa parte del mio materiale retroinformatico. Libri, riviste, quaderni, ancora fanno bella mostra di sé nella casa in Italia. Tornare e trovare questo materiale è come aprire uno scrigno dei ricordi. E' vero che, come ho scritto anche in altri editoriali, possiamo trovare riviste e libri a bizzeffe su internet, ma questi sono i miei personali. Rileggendone le pagine immediati riaffiorano i ricordi, i dubbi, le cose che all'epoca non mi erano chiare e che adesso invece comprendo facilmente, riportandomi istantaneamente indietro nel tempo.

Stranamente provo un misto di sentimenti; vorrei portare questo materiale con me, ma al tempo stesso mi piace l'idea di tornare e ritrovarlo qui a disposizione per un vero e proprio ritorno al passato.

E voi? Avete un vostro personale scrigno dei ricordi? Se volete raccontarci le vostre storie, potete contattarci tramite i riferimenti in calce.

Come di consueto, il numero che avete scaricato è pieno di contenuti interessanti e, speriamo, stimolanti, ma prima di lasciarvi a godere del numero 24, voglio augurare a nome di tutta la redazione una buona estate a tutti.

Francesco Fiorentini

Email: retromagazine.redazione@gmail.com

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

Twitter: twitter.com/RetroMagazineW

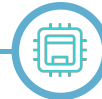
SOMMARIO

◇ L'Olivetti M20 e la storia di un sito web	Pag. 3
◇ LM80C Color Computer Parte 4	Pag. 7
◇ Giappone 12ma puntata: Game & Watch Vs MADrigal	Pag. 11
◇ Decuplicare facilmente i giochi disponibili nel THE64? Si può fare!	Pag. 18
◇ RITORNO AL... PASSATO - Tappa nr. 2: Windows 2000	Pag. 19
◇ Caratteri ridefiniti per Amstrad CPC	Pag. 21
◇ Uno splash screen per Amstrad CPC in formato SCR	Pag. 23
◇ Abbreviazioni per appunti sull'uso di una Interfaccia Grafica	Pag. 27
◇ Speciale Scacchi: 1° Torneo RMW per home computer a 8-bit	Pag. 30
◇ Introduzione ad ARexx – quarta parte	Pag. 34
◇ HEART CHASER 2 - un gioco in Locomotive Basic	Pag. 38
◇ YAMI NO RYŪŌ HADESU NO MONSHŌ (MSX)	Pag. 45
◇ WON-SI-IN (MSX)	Pag. 46
◇ CYBORG-Z (MSX)	Pag. 47
◇ ELLENICA (Megadrive)	Pag. 48
◇ STURMTRUPPEN (Amiga)	Pag. 53
◇ WONDER BOY DRAGON'S TRAP (PC)	Pag. 55
◇ MOON CRESTA (Arcade)	Pag. 57
◇ RAGING BEAST (C64)	Pag. 59
◇ LUCI DELLA FINANZA (Piattaforme varie)	Pag. 60
◇ WARDNER (Arcade)	Pag. 61

Hanno collaborato alla stesura di questo numero di RetroMagazine World:

- Alberto Apostolo
- Carlo N. Del Mar Pirazzini
- Daniele Brahimi
- Davide Bucci
- David La Monaca
- Ermanno Betori
- Flavio Soldani
- Francesco Fiorentini
- Gianluca Girelli
- Giorgio Balestrieri
- Leonardo Miliani
- Marco Fiaschi
- Marco Pistorio
- Mathias Lorenz
- Michele Ugolini
- Querino Ialongo
- Sakis Kaffesakis
- Supporto grafico Irene G. Valeri
- Copertina a cura di Flavio Soldani





L'Olivetti M20 e la storia di un sito web

di Davide Bucci (traduzione in italiano a cura di David La Monaca)

1. Introduzione

L'Olivetti M20 è stato un computer molto particolare, sviluppato in un'epoca in cui non era ancora scontato che i processori Intel e MS-DOS avrebbero avuto un ruolo predominante negli anni a venire. In definitiva, l'M20 non avrebbe avuto successori diretti e non avrebbe finito per creare un ecosistema abbastanza esteso da sopravvivere per più di qualche anno sul mercato dei personal computer. Tuttavia, ha sicuramente trovato una sua piccola nicchia, soprattutto in Europa ed ha fornito per anni il suo contributo al lavoro quotidiano di molte piccole imprese.

Quest'articolo, nella sua prima parte, presenta l'M20 ed il sistema operativo che Olivetti sviluppò per questo computer, il PCOS. Sono tuttora il webmaster di un sito web dedicato a questa macchina, che è andato per la prima volta online nel 2005, pertanto la seconda parte di questo articolo ha un carattere più personale e descrive la storia del mio M20 e quella del sito web ad esso dedicato.

2. Il ritmo di un altro tamburo, lo Z8001

Lo sviluppo dell'M20 (nome di progetto: PC1000) iniziò nel 1979 nel Centro di Tecnologia Avanzata Olivetti (ATC)

di Cupertino, California, non lontano dalla sede centrale della Apple. Olivetti presentò il computer al pubblico il 31 marzo 1982, nel magnifico castello di Agliè, non troppo lontano da Ivrea, in Piemonte [1]. Il nome M20 era già stato utilizzato per un vecchio modello di macchina per scrivere nel 1920 ed Olivetti evidentemente sperava di ripetere il suo successo per l'allora nascente mercato dei personal computer. Il neonato sistema era basato sul processore Zilog Z8001, come tutte le macchine della 'LINEA 1' (L1), che comprendeva anche i minicomputer M30, M40 e M60. L'M20 utilizzava un sistema operativo proprietario sviluppato appositamente: il Professional Computer Operating System o PCOS-8000. Un dettaglio curioso dell'evento della presentazione ufficiale fu che diversi ministri e rappresentanti politici italiani furono invitati al meeting di Agliè: all'epoca, presentare un nuovo computer in Italia non era considerato come un evento tecnologico di minore importanza, ma era anche un evento politico e (perché no?) culturale. Tuttavia il computer fu, di fatto, distribuito in tutto il mondo [2-4] suscitando ovunque un certo interesse. Il titolo di questo paragrafo, ad esempio, parafrasa il sottotitolo della prova pubblicata in una rivista americana [3].



Figura 1 - il case dell'M20, firmato da Ettore Sottsass

Olivetti era nota per il suo design industriale e l'M20 non faceva eccezione. Un case in plastica dall'aspetto molto moderno fu progettato da Ettore Sottsass (1917-2007) e fu molto elogiato all'epoca. Il progettista vantava una lunga storia di collaborazione con Olivetti. In particolare, fu responsabile del design industriale dell'Elea 9003 nel 1959, uno dei primi computer completamente transistorizzati al mondo. Nel 1969 Sottsass disegnò anche l'elegante macchina per scrivere Valentine, tutta rivestita da uno sgargiante color rosso. L'M20 ospitava una scheda madre di dimensioni rispettabili, una tastiera ben accessoriata (non rimovibile) e due unità floppy disk da 5,25" oppure un'unità floppy disk e un disco rigido. Un monitor separato da 12" poteva essere posizionato sopra l'unità centrale ed orientato a piacere. Il risultato si può apprezzare in Figura 1. Il look complessivo mantiene tuttora un certo fascino retro-futuristico.

3. L'hardware

La scheda madre, mostrata in figura 2, occupava praticamente tutta la base del computer, per cui la scelta di una tastiera integrata era dettata da vincoli tecnici. La scheda presentava un processore Z8001 a 4 MHz, che utilizzava un bus a 16 bit, 128 KB di RAM, un'interfaccia stampante e un'interfaccia seriale RS232 con una velocità massima di 9600 baud. La RAM poteva essere espansa fino ad un massimo di 512 KB utilizzando fino ad un massimo di tre schede di espansione. Due slot erano





disponibili per le espansioni come una scheda di interfaccia IEEE, un controller per hard disk e così via. La qualità costruttiva era eccellente e il sistema era pienamente modulare: l'assemblaggio e il disassemblaggio richiedevano soltanto poche viti ed alcuni incastri facili da chiudere o aprire.

La tastiera era leggermente rumorosa, ma veloce ed eccellente per la digitazione rapida. I lettori italiani possono notare in figura 3 il classico layout QZERTY delle macchine per scrivere, dato che il moderno layout QWERTY italiano più le lettere accentate venne introdotto, dalla stessa Olivetti, a partire dal modello M24. Olivetti voleva probabilmente che le persone si sentissero il più possibile a loro agio durante le sessioni di lavoro alla tastiera e si rivolgeva a chi era già abituato alle macchine per scrivere, perlomeno nella versione BC (Business Computer). Altre versioni riportavano sui tasti i comandi BASIC per agevolare gli utenti programmatori.

Le recensioni contemporanee [3] tendono a menzionare esplicitamente l'assenza di tasti funzione separati, ma comunque alcune funzioni speciali sono accessibili tramite due tasti colorati denominati Command e CTRL, che si dovevano tenere premuti prima di azionare un altro tasto. Una legenda rimovibile poteva essere inserita in una fessura sulla parte superiore della tastiera, per ricordare all'utente la funzione di ogni tasto. Ai due tasti aggiuntivi chiamati S1 e S2 potevano essere assegnate varie operazioni. L'omissione più evidente è il tasto backspace, normalmente ottenuto con CTRL+H (il tasto Tab si ottiene con CTRL+I). Il tasto S2 si trova esattamente nel punto in cui ci si aspetterebbe il tasto backspace, ma era configurato di default come Carriage-Return (tasto Invio). Qualsiasi tasto poteva essere riprogrammato: ad esempio, se si pensava di non poter vivere senza il tasto backspace, al suo posto si poteva digitare 'CK &C3, 8' per assegnare il tasto S2 (codice di scansione C3 in esadecimale) a quella funzione (codice ASCII 8 = CTRL+H). Si poteva anche rendere questa funzione permanente, con il comando PSAVE, in modo da ritrovarla già pronta all'apertura di ogni sessione. Un piccolo cicalino montato sul PCB della tastiera era l'unico dispositivo sonoro disponibile.

Olivetti era molto attenta ai mercati internazionali e diverse versioni nazionalizzate dell'M20 sono state rese disponibili nel tempo. Questo in realtà causò alcuni problemi in quanto i set di caratteri erano diversi per ciascuna versione; infatti alcuni caratteri devono essere adattati se si digita un sorgente BASIC da un manuale in inglese (un esempio è il carattere "#" del set inglese che diventa "£" in francese o in italiano). Il problema fu notato già all'epoca. In una recensione di una rivista francese [4] fu detto che una macchina nazionalizzata fu consegnata con i manuali in inglese perché quelli in francese non erano ancora pronti e la cosa creò qualche grattacapo all'interno della redazione durante i test. Occorre prestare attenzione a questo dettaglio ancor oggi, poiché molti dei manuali disponibili

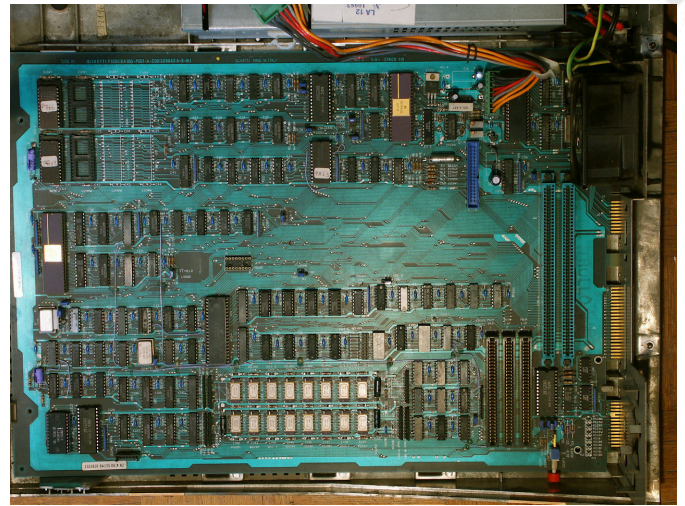


Figura 2 - L'ampia motherboard dell'M20

sono solo in inglese [5].

La maggior parte delle macchine era dotata di un doppio lettore di floppy disk da 5.25" la cui capacità nominale di 320 KB diventava di 272 KB quando un disco (chiamato "volume" nel gergo del PCOS) veniva formattato. Una particolarità del sistema operativo era che si utilizzava una diversa formattazione della testina 0/traccia 0 (MF, 128 byte/settore) rispetto alle altre tracce (MFM, 256 byte/settore). Esistono diverse configurazioni: un drive Winchester da 11,5 MB (simile all'MFM ST251 che in effetti si può adattare) era disponibile per sostituire uno dei drive per dischetti ed erano acquistabili anche drive per dischi da 160 KB e da 640 KB. In modo un po' insolito, nella configurazione hardware standard, l'M20 identificava i due drive per dischetti come 1: a sinistra e 0: a destra. Un'intera gamma di stampanti era disponibile presso Olivetti: la stampante PR1450S, la PR1450G (G sta per "grafica"), la PR1471, a matrice di punti e la PR2400 termica. L'M20 ha un'interfaccia parallela standard Centronics ma, come per l'interfaccia RS232 e lo schermo, impiega un connettore proprietario a pettine. Una presa di corrente collegata al computer permetteva di accendere automaticamente la stampante o il monitor a colori, il solo che richiedesse un'alimentazione separata al momento dell'accensione del computer.

L'M20 non fa distinzione tra i modi grafici e i modi di testo, tutto è disegnato in uno schermo grafico con una risoluzione di 512x256 pixel. Ciò offriva una grande flessibilità a scapito di un leggero ritardo in alcune operazioni, come lo scorrimento del testo. Il sistema operativo utilizza una griglia di 64x16 caratteri o una più convenzionale disposizione 80x25. Il modello base era dotato di un display in bianco e nero, ma un'opzione costosa (e quindi abbastanza rara) era il già citato monitor a colori che richiedeva speciali schede di espansione RAM dotate di shift register per effettuare una conversione parallelo/seriale. Questi agivano come una sorta di semplice circuito DMA in grado di inviare i bitplane R, G e B direttamente all'uscita verso lo schermo.

Lo schermo in bianco e nero era alimentato dal computer





con 12 V e utilizza un unico cavo per fornire l'alimentazione e i segnali video. Non è standard, i segnali sono R, G, B, B/N, più V-Sync e H-Sync (tutti con livelli TTL). La sincronizzazione verticale funziona tramite impulsi positivi a 68,2 Hz, quella orizzontale a 18,7573 kHz. Queste frequenze non standard rendono difficile trovare un monitor compatibile se quello originale non funziona correttamente o è assente.

4. Il sistema operativo

Il sistema operativo PCOS appare singolare ed alquanto oscuro se osservato da un punto di vista moderno, ma non era inferiore al DOS 1.0 di cui era fornito il primo PC 5150 IBM, presentato alla stampa solo 7 mesi prima dell'M20. Il PCOS non disponeva di directory e sottodirectory, ma permetteva di controllare l'accesso ai file con una password, anche se la chiave di protezione non era considerata molto robusta. Si potevano usare abbreviazioni per comandi come 'VF' (per 'VFORMAT'), utile per formattare un nuovo disco. En passant, vale la pena ricordare che questo comando a volte non riusciva a formattare un dischetto già formattato per MS-DOS, per cui una soluzione radicale ma efficace era quella di distruggere tutti i dati precedenti usando una calamita. Una cosa interessante era che il sistema operativo era facilmente configurabile e permetteva di decidere quali comandi mantenere residenti, a seconda della configurazione della memoria e delle proprie esigenze. Sistemi operativi alternativi al PCOS sono stati messi a disposizione da Olivetti. Nell'ottobre 1982, infatti, propose una scheda di espansione chiamata Alternate Processor Board (APB), contenente un Intel 8086 che permetteva all'M20 di eseguire MS-DOS. Degno di nota era il sistema operativo CP/M-86, un adattamento del ben noto CP/M per il processore Intel, che ovviamente richiedeva la scheda APB.

L'Olivetti M20 utilizzava dischetti standard DS-DD e l'accesso era ragionevolmente veloce, la formattazione completa su un'unità da 320 KB richiede 1 minuto e 18 secondi mentre una copia completa del disco (usando il comando 'VCOPY') veniva terminata in 1 minuto e 24 secondi. L'avvio del PCOS 4.1a richiede 27 secondi sulla mia macchina da 512 KB di RAM, compreso il self-test iniziale. In generale, l'M20 è stato ampiamente riconosciuto come una macchina relativamente potente: lo Z8001 era un processore abbastanza avanzato per l'epoca ed il computer fu venduto da Olivetti ad un prezzo interessante. Una versione del Microsoft BASIC era compresa nell'acquisto

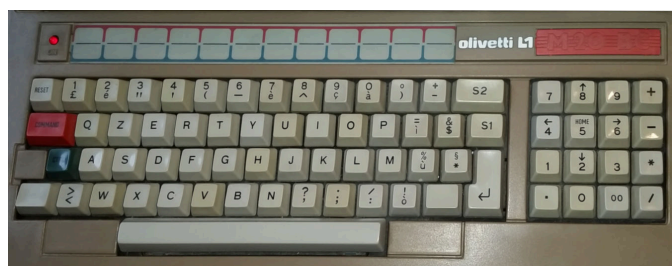


Figura 3 - il layout QZERTY italiano della tastiera

e la velocità di esecuzione dei programmi sull'M20 era competitiva con l'originale IBM 5150.

Il successore dell'M20 prese il nome di M24 ed uscì nel 1983, ma questo modello era basato sul processore 8086 e sul sistema MS-DOS. Era una macchina molto efficace e riscosse un vasto e meritato successo. Olivetti offrì anche per l'M24 una "Alternate Processor Board" con microprocessore Z8001 a bordo e ciò permise a questo nuovo modello di eseguire il PCOS ed il software sviluppato fino ad allora per l'M20. Qualche anno dopo, divenne chiaro, tuttavia, che la campana a morto stava suonando per il PCOS. La versione 4.1a, sviluppata nel 1984, è l'ultima versione di cui sono a conoscenza.

5. Il mio M20 e la nascita di un sito web

La primavera del 1993 fu piuttosto soleggiata nella mia piccola città, vicino a Torino. Durante uno di quei pomeriggi apparentemente interminabili che un adolescente si trova ad avere per le mani, dopo la scuola, andai a trovare un amico. Will era un ragazzo molto intelligente ed un chitarrista incredibilmente abile, se si pensa che aveva da poco compiuto solo 14 anni. Il più delle volte, a casa sua, lui suonava ed io cercavo di imparare qualcosa da lui. Quel giorno mi salutò con un sorriso e mi disse: "Dave, ho un computer che potrebbe interessarti", mostrandomi un oggetto squadrato marrone e beige che giaceva immobile nella sua stanza. "Mio padre l'ha recuperato dal negozio di ricambi per auto qui vicino, volevano gettarlo via perché è vecchio. Non possiamo usarlo: se vuoi, prendilo". Inutile dire che accettai.

"Cos'è questo... coso?", mio padre non era certo entusiasta quando quella sera venne a prendermi in auto e mi vide con quello strano computer. Avevamo già un PC 286, ma avevo visto alcune foto sulle riviste di computer vecchie di più di dieci anni, che leggevo all'epoca. Quella fu comunque la prima volta che vidi un Olivetti M20 di persona e mi piacque subito la sua forma stranamente spigolosa ma elegante.

Scoprii presto che il mio "nuovo" computer era, in effetti, inutile senza un sistema operativo adeguato. Si accendeva, ma aveva solo due unità disco e nessun disco rigido. Una volta terminato il self-test rimaneva quindi completamente inerte, in perenne attesa. Dopo qualche settimana, chiesi aiuto a Ugo, uno dei nostri vicini di casa dell'epoca. Abitava nella casa accanto a quella dove vivevo con i miei genitori ed era stato lui a insegnarmi praticamente tutto quello che all'epoca sapevo sui computer. Avevo già ricevuto da lui in regalo quelle vecchie riviste di computer che leggevo con tanta soddisfazione ed Ugo venne nuovamente in mio soccorso con la sua solita pazienza e gentilezza, offrendomi dei vecchi dischetti per questa strana macchina. Venne fuori che aveva usato una macchina simile dieci anni prima e la conosceva quindi abbastanza bene.

Appresi così che l'M20 utilizzava uno strano sistema operativo chiamato PCOS e che potevo giocare un po'



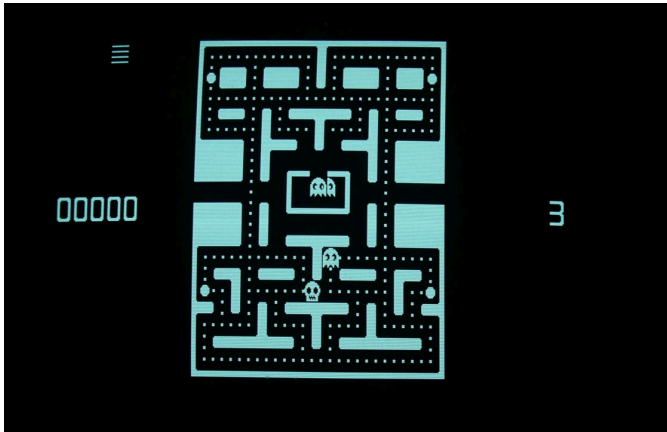


Figura 4 - un clone di Pac-Man

con il Microsoft BASIC in dotazione. Conoscevo già questo "dialetto" del linguaggio BASIC, così passai alcune ore a sperimentare e a disegnare linee e cerchi sullo schermo. Trovai anche alcuni semplici giochi (come quello della figura 4), essenziali ma affascinanti.

Era divertente, ma senza manuale e informazioni dettagliate, non c'era molto all'epoca che potessi fare. Ho conservato il mio M20 con cura, ma non l'ho usato spesso, almeno fino al 2004. A quel punto vivevo già in Francia e, essendo diventato un utente abituale di Internet, ho pensato che sarebbe stato interessante cercare informazioni sul mio vecchio e bizzarro computer. Scoprii, con piacere, che le scansioni di alcuni manuali erano già disponibili, ma che molte informazioni erano sparse in diversi siti web e anche in modo frammentato.

Così decisi di agire e scelsi una soluzione che era ovvia in un momento in cui su Internet i social network erano ancora di là dal diffondersi: costruii un sito web, scrivendo tutto l'HTML necessario e un po' di PHP io stesso. È andato online nel 2005 ed esiste ancora oggi [6], con lo stesso look vecchio stile che scelsi allora. Grazie al sito sono entrato in contatto con molte persone interessanti e incredibilmente competenti. L'Olivetti M20 ha una piccola ma devota comunità internazionale. A questo proposito, desidero ringraziare in modo particolare Roberto Bazzano che ospita il sito sui suoi server dal 2007.

Intorno al 2006, ho cercato di migliorare la mia macchina: aveva infatti bisogno di alcune riparazioni ed era ancora nella configurazione base a 128 KB di RAM. L'alimentatore originale non era più all'altezza e, dopo più di 20 anni, le testine dei drive dovevano essere riallineate. Sono riuscito a contattare un ex-tecnico Olivetti al telefono, che mi ha spiegato la procedura di fabbrica per riportarli in vita. Ho anche studiato gli schemi delle espansioni di memoria e ne ho costruito una con due moduli SIMM recuperati qua e là, portando così la mia macchina ai fatidici 512KB di RAM. Olivetti deve aver usato un sistema CAD primitivo all'epoca, gli schemi elettronici sono incredibilmente difficili da leggere e letteralmente pieni di errori. Sospetto (e spero) che quelli utilizzati per la produzione non fossero quelli che appaiono nel manuale dell'hardware.

In tutti questi anni ho cercato di aiutare altri utenti dell'M20, cercando di imparare cose nuove e documentando tutto quello che ho fatto con la mia macchina. Pertanto il sito ospita anche i contributi di coloro che sono stati tanto gentili da inviarmi il racconto delle loro esperienze. Non sarebbe stato possibile redigere quest'articolo senza l'aiuto che ho ricevuto attraverso il sito web.

6. Conclusione

In questo articolo ho descritto un computer piuttosto particolare, l'Olivetti M20. Inizialmente molto competitivo rispetto al PC 5150 IBM, fu in definitiva penalizzato sul mercato dalla mancanza di software e forse ancor di più da un mancato riconoscimento commerciale. Ho descritto sia l'hardware che il sistema operativo PCOS-8000 e nell'ultima parte ho raccontato la storia della mia macchina (rappresentata nelle figure) e come ho deciso di costruire un sito web ad essa dedicato. Questo mi ha permesso di entrare in contatto con molte persone di diversi paesi che sono appassionate dell'Olivetti M20.

Un'ultima nota: ho scritto la prima versione di questo articolo sul mio Olivetti M20 con Oliword 1.2, trasferendo i file con un link RS232 e usando un semplice script bash per rimuovere i codici di controllo. L'editing finale è stato poi completato su un più comodo MacBook Pro.

Riconoscimenti

Vorrei ringraziare calorosamente il mio amico William Barbero, che più di 25 anni fa mi ha permesso di recuperare il mio M20. Grazie di cuore anche a Ugo Garombo, che mi ha salvato con i floppy disk e mi ha insegnato a tredici anni cosa fosse un sistema operativo. Ringrazio anche Roberto Bazzano che ha gentilmente ospitato il sito dell'M20 per moltissimo tempo e continua a farlo tuttora. La versione in inglese di questo articolo (pubblicato su RMW #02) sarebbe stata probabilmente imbarazzante da leggere senza la gentile e attenta correzione di bozze di Chris Carter, che ha anche scovato alcune imprecisioni. I restanti errori sono miei.

Bibliografia e riferimenti

- [1] M. Marinacci "Anteprima dell'Olivetti M20", MC MicroComputer n. 8, Aprile 1982
- [2] W. Marett "Italian micro enters New Zealand business market", Bits & Bytes n. 9, Giugno 1983
- [3] S. Mello-Grand "Review of the Olivetti M20", Byte vol. 8 n. 6 Giugno 1983
- [4] X. Bonfils, A. Pinaud, J.-P. Brunerie, "Au banc d'essai Olivetti M20", L'ordinateur individuel, n. 41, Settembre 1982
- [5] Vedi ad es.: <http://www.z80ne.com/m20/index.php?argument=sections/manuals/manuals.inc>
- [6] <http://www.z80ne.com/m20>





LM80C Color Computer

Un computer autocostruito nel 2019 basato sullo Z80 - parte 4

di Leonardo Miliani

L'argomento di questo articolo è leggermente diverso dagli altri perché andremo ad affrontare l'analisi del computer più da un punto di vista software mentre le scorse volte ci eravamo concentrati sull'hardware. Il computer LM80C è una macchina completa e funzionante, dotata di interessanti caratteristiche che, come abbiamo visto negli scorsi articoli, sono di tutto rispetto e gli avrebbero permesso di non sfigurare fra gli 8 bit in voga negli anni '80, potendo affiancare alla pari i sistemi più diffusi di quel periodo. Ma ciò che rende l'LM80C un computer divertente da usare è la sua facilità d'uso: integra infatti un sistema operativo che permette di usare già all'accensione, senza bisogno di software aggiuntivo, quelle sue caratteristiche distintive come la grafica ed il suono. Per far sì, infatti, che gli home computer potessero essere usati non appena tolti dalla scatola anche dagli utenti più inesperti, essi venivano dotati di un linguaggio di programmazione integrato che assolveva anche alle funzioni di interfaccia di comando, grazie a cui era possibile non solo inserire programmi ma anche pilotare l'hardware senza ricorrere a routine in linguaggio macchina. Il complesso software che sovrintende alla gestione della macchina è quindi, in un home computer, tanto importante quanto l'hardware stesso: è stata proprio la loro facilità d'uso che, tra le altre cose, ne ha permesso una grande diffusione.

Il sistema operativo di un home computer ad 8 bit è generalmente strutturato su più livelli. Il livello più alto è, come detto, rappresentato da un'interfaccia di input/output grazie alla quale l'utente può digitare dei comandi in un linguaggio pre-caricato ed istruire il computer per eseguire delle operazioni, ad esempio stampare il risultato di un calcolo matematico, leggere un programma da una periferica di massa oppure emettere un suono. Il livello sottostante è composto dal parser del linguaggio installato, che analizza i comandi inseriti e li traduce in una serie di compiti ancora più elementari che sono eseguiti dal sistema per svolgere l'operazione richiesta dall'utente. Se l'operazione da richiedere coinvolge anche l'hardware vengono chiamate delle routine di livello ancora più basso che comunicano direttamente con le componenti del sistema. Anche l'LM80C è strutturato così, come andremo a vedere. Ma partiamo dal linguaggio.

L'interprete BASIC integrato

Nell'LM80C è pre-installato l'LM80C BASIC, un dialetto del popolare linguaggio BASIC (in figura 1 potete vedere un listato di esempio). Prendiamo ad esempio le due seguenti istruzioni in LM80C BASIC:
SCREEN 2:CIRCLE 128,96,50,8



Esse dicono al computer di passare in modalità grafica 2 e disegnare un cerchio di raggio 50, centrato nello schermo, usando il colore rosso. Come fa il sistema ad interpretare queste istruzioni ed a generare il risultato sullo schermo? Ci riesce perché nella sua ROM (la memoria non volatile, che non perde i dati quando viene tolta l'alimentazione elettrica alla macchina) è stato inserito un interprete che analizza i comandi digitati dall'utente e li trasforma in una serie di operazioni in linguaggio macchina eseguibili dalla CPU. Il BASIC, come altri linguaggi di programmazione, è in realtà un'agevolazione per noi umani: ci permette di utilizzare un sistema di comunicazione facile da capire e da scrivere perché la CPU capisce solo il linguaggio "macchina" (non a caso si chiama così), ossia i valori 0 e 1. Per semplificare la programmazione, al posto di far inserire una serie interminabile di numeri senza significato per noi essere umani è stato inventato inizialmente l'assembly (da non confondere con "assembler", che è invece il programma assemblatore, che trasforma cioè il sorgente assembly in linguaggio macchina), un linguaggio di bassissimo livello di tipo "mnemonico" perché composto da sigle che aiutano a "ricordare" l'istruzione che esse rappresentano. Un esempio di codice assembly Z80 con, accanto il "significato" per il programmatore:

```
INC A → INC(rement) A
CP $10 → C(om)P(are to) $10
JP NZ,$8000 → J(um)P (if) N(ot) Z(ero to) $8000
```

Le istruzioni significano, nell'ordine: incrementa il contatore A; compara il suo valore con 16 (il corrispondente in decimale del valore esadecimale \$10); se non è uguale, salta all'indirizzo di memoria 32768 (decimale per \$8000). Ho parlato di "assembly Z80" perché ogni CPU, siccome è differente dalle altre, ha un ben preciso set di istruzioni e, conseguentemente, il proprio assembly. Ad esempio, l'istruzione assembly 6502 LDX #\$10, che significa "carica nel registro X il valore decimale 16", non ha nessun senso in assembly Z80 sia per la sintassi diversa delle istruzioni stesse sia perché essa non ha il registro X. È ovvio che scrivere in assembly è molto complesso: per assegnare semplicemente un valore ad una variabile bisogna scomporre il compito in tante piccole operazioni (si veda in figura 2 un esempio di una porzione del firmware dell'LM80C in assembly Z80). Per venire incontro agli





```
10 A=INKEY(10)
20 SCREEN 2:GPRINT "Press any key for next demo...",1,23
30 DRAW 127,95,RND(1)*256,RND(1)*192,RND(1)*13+1
40 IF INKEY(0)=0 THEN 30
70 CLS:GPRINT "Press RUN/STOP to stop",5,23,15,1
80 FOR I=0 TO 255
90 DRAW 127,95,I,0,RND(1)*13+1:NEXT
100 FOR I=0 TO 191
110 DRAW 127,95,255,I,RND(1)*13+1:NEXT
120 FOR I=255 TO 0 STEP -1
130 DRAW 127,95,I,191,RND(1)*13+1:NEXT
140 FOR I=191 TO 0 STEP -1
150 DRAW 127,95,0,I,RND(1)*13+1:NEXT
160 GPRINT "Press any key to exit",5,23,1,15
170 A=INKEY(5):IF A=0 THEN 170
180 SCREEN 1
```

Figura 1: Un esempio di programma in LM80C BASIC

utenti alle prime armi, ma anche per velocizzare la scrittura dei programmi a quelli più esperti, furono sviluppati i cosiddetti linguaggi di alto livello, ossia linguaggi più semplici da scrivere e più intuitivi da leggere e comprendere, che venivano anche integrati nelle memorie dei computer di quel periodo.

Il linguaggio che tra la fine degli anni '70 ed i primi anni '80 del XX secolo andava per la maggiore era il BASIC! Il linguaggio nacque nel 1964 (ebbene sì, ha più di 50 anni!) presso l'università di Dartmouth (USA) ad opera dei professori John Kemeny e Thomas Kurtz, che svilupparono questo linguaggio per facilitare l'apprendimento dell'informatica ai loro studenti. Il BASIC si diffuse molto rapidamente grazie a diversi fattori ma, principalmente, perché i suoi inventori decisero di non brevettarlo e tenerlo confinato nella loro università: fecero altresì un'intensa opera di divulgazione anche presso altri istituti così che molte persone iniziarono ad usarlo. Il fatto che fosse un linguaggio che non richiedesse molte risorse (i primi interpreti occupavano pochi KB di memoria) lo rese perfetto per l'uso sui microcomputer che iniziavano a diffondersi negli anni '70 ed il suo essere un linguaggio di tipo generico portò le persone a scrivere programmi per i più svariati compiti ed a pubblicarli anche sulle riviste e sulle newsletter dei club di appassionati di computer dell'epoca. La Microsoft stessa iniziò la sua ascesa commerciale vendendo un interprete BASIC, inizialmente per l'Altair 8800 e successivamente distribuendolo per ogni tipo di macchina. Il Microsoft BASIC divenne così popolare che alla fine si affermò come uno standard e molti costruttori di computer, come Comodore, Apple e Atari, decisero di adottarlo direttamente oppure di usare

un interprete BASIC derivato da esso. Usare un interprete BASIC era una scelta scontata, all'epoca: permetteva di avere all'accensione della macchina un linguaggio molto diffuso che potesse essere usato dopo un breve periodo di apprendimento.

L'M80C BASIC deriva dal Nascom BASIC, a sua volta derivato dal Microsoft BASIC. Il Nascom BASIC era un interprete BASIC che Nascom Computers, un costruttore inglese attivo alla fine degli anni '70 del XX secolo, distribuiva per i suoi computer e terminali. Derivava dalla versione per Z80 del Microsoft BASIC e con esso manteneva una grande compatibilità. E così è anche per l'M80C BASIC: a parte le differenze legate alle differenti architetture hardware, un programma in Microsoft BASIC può girare con poche modifiche sull'M80C. Ho scelto il Nascom BASIC come partenza perché i computer della Nascom erano così diffusi che agli inizi degli anni '80 si era formata una fiorente comunità con diverse pubblicazioni dedicate: su una di esse fu pubblicato il sorgente completo e commentato dell'interprete. L'M80C BASIC, come il Microsoft BASIC, è un linguaggio interpretato (anche se il BASIC nacque come linguaggio compilato): questo significa che dietro non c'è un compilatore che trasforma prima il sorgente in codice macchina eseguibile dalla CPU ma esiste un interprete che analizza ogni comando inserito e, una volta riconosciuto, esegue le corrispondenti routine in linguaggio macchina.

Tornando al nostro esempio iniziale, l'interprete incontrerà per prima l'istruzione SCREEN 2. Una volta riconosciuto il comando come uno di quelli validi, l'interprete chiamerà la routine corrispondente che, verificata la correttezza





della sintassi (ad esempio verrà controllato che la modalità video richiesta sia valida), imposterà la modalità video richiesta mediante l'esecuzione di specifiche sub-routine, sempre in linguaggio macchina, grazie alle quali verranno spediti i dati necessari al chip video. Ecco quindi, i diversi strati software del sistema a cui abbiamo accennato in precedenza. Terminata l'esecuzione dell'istruzione, l'interprete continuerà l'analisi della riga di comando ricevuta, e troverà il carattere dei 2 punti: questo viene interpretato come una specie di avviso che segnala all'interprete che "c'è dell'altro", vale a dire che l'interpretazione della riga non è terminata. Passando oltre, troverà l'istruzione CIRCLE 128,96,50,8. Qui la situazione è leggermente più complessa perché i parametri non solo sono in numero superiore ma l'ultimo è anche opzionale, ossia può essere passato oppure no: in quest'ultimo caso il computer deve usare il colore predefinito. Terminata l'analisi sintattica, inizia l'esecuzione di una routine in linguaggio macchina molto complessa che calcola le coordinate di ogni singolo punto costituente la circonferenza. Trovate le coordinate, viene chiamata una seconda routine, di un livello più basso, che esegue la visualizzazione di un punto data una coppia di coordinate X,Y. La prima analisi che essa fa riguarda il controllo della visibilità o meno del punto, se cioè le coordinate passate ricadono sulla porzione di schermo visibile oppure no: in questo caso il punto non viene disegnato. Se invece le coordinate sono valide, un'altra routine si occupa di trasformare le coordinate nell'indirizzo della cella della memoria video che contiene il pixel da visualizzare e poi di accendere il bit corrispondente ed impostare il colore richiesto. Tutto questo viene ripetuto per tutti i punti della circonferenza. Alla fine di questo comando l'interprete tornerà ad analizzare la riga passata e, non trovando

nessun altro comando, restituirà il controllo all'utente. Sarà tornato in quella che viene detta "modalità diretta", una speciale modalità dove ogni comando inserito viene eseguito immediatamente. In realtà le cose sono un po' più complesse di così...

Il firmware

Non è vero che solo alla pressione del tasto "RETURN" (o "ENTER" sulle macchine che provenivano dal Regno Unito) il sistema esegue i comandi inseriti, perché esiste un programma che gira continuamente, ed è quello che gestisce proprio l'interfacciamento con l'utente: esso analizza continuamente tutte le periferiche di input in attesa di un dato proveniente dall'esterno. Ad esempio, il sistema legge ad intervalli regolari la tastiera e visualizza sullo schermo il carattere corrispondente al tasto premuto. Nel caso venga richiesto l'avvio del programma in memoria il controllo passa all'interprete, che eseguirà riga per riga il listato. Ma anche in questo caso il firmware resta "sull'attenti" perché continua a gestire la visualizzazione dei caratteri stampati sullo schermo, sposta eventuali sprite, genera i suoni quando richiesto, legge la tastiera per l'input e così via. Il complesso software di un computer memorizzato nella ROM che serve al suo funzionamento è detto firmware: questa parola inglese, che deriva dall'unione di "firm" e "software" e che in italiano suona pressappoco come "parte logica permanente", indica il software preinstallato in un sistema che, di solito, l'utente finale non può o non è in grado di modificare e che si occupa dell'avvio della macchina e della gestione delle sue funzioni base.

Il firmware dell'LM80C non è composto solo dall'interprete BASIC che, comunque occupa gran parte del codice

```

KEY:  dec    HL          ; dec 'cos GETCHR INCs
      call  GETCHR     ; Get next character
      jp    Z,LSTKEYS ; jump if nothing follows
                          ; change FN keys
      call  GETINT     ; get a number
      and  A          ; is it 0?
      jr   NZ,KEYCH   ; no, jump over
      push HL         ; yes - reset FN keys to defaults
      push DE         ; store HL & DE
      ld   HL,DEFFNKS ; pointer to default FN keys texts
      ld   DE,FNKEYS  ; pointer to destination
      ld   BC,$0080   ; 128 chars to be copied
      ldir          ; restore default texts
      pop  DE         ; retrieve DE
      pop  HL         ; retrieve HL
      ret          ; return to caller
KEYCH: cp    $09      ; is it >= 9?
      jp   NC,SNERR  ; yes - syntax error
      dec  A         ; FN key in range 0~7
      add  A,A       ; multiply A...
      add  A,A       ; ... times 4...
      add  A,A       ; ... to get the correct...
      add  A,A       ; ... offset fo FN key text
      ld   (TMPBFR1),A ; store FN key offset...
      xor  A         ; ...in a...
      ld   (TMPBFR1+1),A ; ...16-bit register

```

Figura 2: l'inizio della routine relativa al comando KEY che permette di gestire i compiti assegnati ai tasti funzione



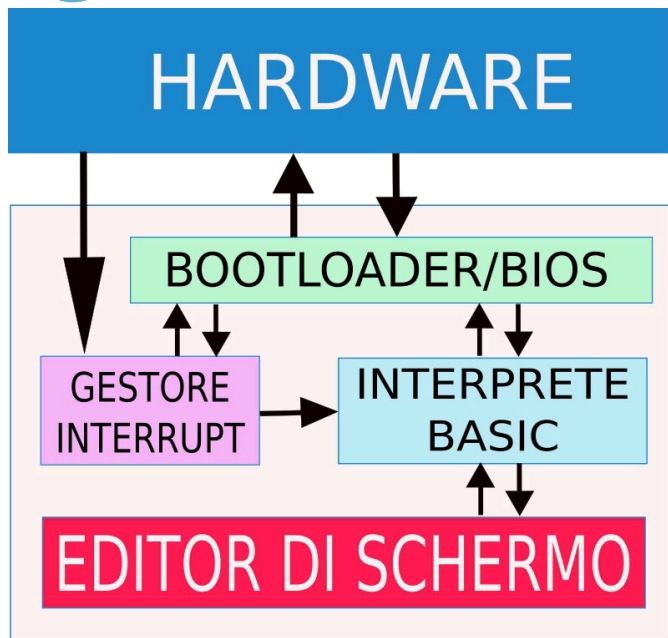


Figura 3: La struttura del firmware

integrato. Esistono più porzioni (figura 3):

- il bootloader, che viene chiamato al reset del computer e che si occupa di inizializzare l'hardware;
- il BIOS, composto da tutta una serie di funzioni di basso livello che permettono l'accesso diretto all'hardware del computer;
- il gestore degli interrupt, che gestisce sia l'interrupt di sistema, che viene richiamato ad intervalli regolari e che si occupa di tutte le operazioni temporizzate del sistema come la lettura periodica della tastiera, la disattivazione della riproduzione dei toni sonori, il lampeggio del cursore, ed altro, nonché gli interrupt sollevati dalle periferiche del computer;
- l'interprete BASIC vero e proprio che può funzionare sia in modalità diretta che indiretta;
- l'editor di schermo, che gestisce la stampa dei caratteri sullo schermo e lo spostamento del cursore, agendo come vera e propria interfaccia tra utente e macchina.

Il bootloader sono poche righe di codice. Si occupa di indirizzare la CPU verso un preciso punto della ROM dopo un reset e da lì inizia l'avvio del sistema: viene visualizzato il logo, viene impostato il timer di sistema usato come temporizzatore per le operazioni da ripetere ad intervalli regolari, vengono impostati i vettori di interrupt per i chip periferici e la modalità video standard, e poi il controllo viene lasciato all'interprete. L'interprete non dialoga direttamente con l'utente ma questi interagisce con il computer tramite l'editor di schermo. L'editor permette di spostare il cursore e di inserire dei comandi in ogni punto del video. I tasti vengono letti da una routine richiamata dal gestore di interrupt, che ad intervalli regolare scansiona la tastiera per vedere se l'utente ha premuto qualcosa. Se si tratta di un tasto comune, il corrispondente carattere viene visualizzato nella cella occupata dal cursore: per fare questo vengono richiamate diverse routine del BIOS che dialogano direttamente con il chip video. Queste routine si occupano di trasformare

le coordinate X e Y del cursore nell'indirizzo della cella di memoria video corrispondente e poi di scrivere in essa i dati da visualizzare. Se viene premuto il tasto "RETURN" allora tutto il testo presente sulla riga viene passato all'interprete che lo analizza ed esegue eventuali comandi. Anche i comandi fanno affidamento al BIOS per eseguire le loro funzioni: ad esempio, per suonare una nota o per scrivere un carattere.

Il nucleo dell'interprete BASIC era disponibile in rete ma, come detto, era stato sviluppato per un computer di tipo terminale, ossia con cui è possibile interfacciarsi tramite un collegamento seriale e che accetta e restituisce dati su tale canale. Per questo motivo il firmware iniziale era poco meno di 8 KB non offrendo molte funzionalità. La stesura dell'editor di schermo ha richiesto diverse settimane di lavoro e di KB di codice (comprese le mappe caratteri per i font a 6x8 e 8x8 pixel, disegnati completamente a mano). A ciò si è aggiunto il lavoro per rendere il sistema indipendente dalla seriale e trasformarlo in un vero home computer, con l'input passato dalla tastiera e l'output visualizzato sullo schermo. Oltre a questo si aggiunge tutto il codice scritto per gestire tutti i nuovi comandi grafici e audio del computer che, ovviamente, in un terminale non erano presenti. Alla fine il lavoro finale ha portato ad avere 19 KB di firmware, e il lavoro di stesura del codice è stato più lungo e impegnativo di quello per la realizzazione della parte hardware. Questo per far capire che se decidete di intraprendere la strada della realizzazione di un computer partendo da zero dovete considerare che questo processo non riguarda solo la scelta dei chip e il disegno di una scheda madre ma vi attende anche un bel po' di lavoro di stesura del codice per farlo funzionare e poterlo usare e sfruttare al meglio.

Link utili

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





Giappone 12^ puntata: Game & Watch Vs MADriginal

di Michele Ugolini

Cari lettori, diamo il benvenuto a MADriginal, alias Luca Antignano. (cfr. figura 1)

Sapevamo tutti che sarebbe arrivato questo momento.

Stiamo vivendo un anno particolarmente nefasto: il 2020 è sia anno "bisesto" che anno "lustrò".

In Italia siamo ormai quasi completamente usciti dal dramma planetario causato dal Covid19.

Stiamo oltretutto vivendo un periodo sovraffollato di mini console, remake e progetti più o meno riusciti e visto il recente arrivo del "PC Engine CoreGrafx mini", comincio a pensare di essermi smarrito nel mezzo del famoso cammino.

Nelle scorse puntate ho parlato del concepimento dei Game&Watch, dei principali personaggi operanti nella sala parto e soprattutto dei magici ingredienti nipponici che hanno potuto tramutare la follia, in un oggetto ad oggi entrato nella sacralità di noi collezionisti. Ho sempre definito i Game&Watch come un prodigio elettronico dotato di una stilizzata anima immortale. E' incredibile pensare a quante figure nipponiche dovremmo ringraziare per la loro genialità divenuta ormai eterna. Altrettanto incredibile è pensare quanto il genio italiano, dall'altra parte del mondo, condivida ideali creativi e brillanti!

Il mio amore infinito per il Giappone è motivato anche da questa spazialità asincrona, animata da

una personalità degna di confronto, similmente all'universo: generato da una lieve ed asincrona fluttuazione tra opposti stadi di materia.

Bene, iniziamo l'intervista, oggi gli onori vanno tutti al geniale Luca Antignano.

RMW: "Luca, benvenuto, un caloroso abbraccio virtuale, raccontaci qualcosa di te e del tuo nickname, della tua passione per i G&W che anche io come tantissimi lettori condividiamo assieme a te!"

LA: "Figurati, grazie a te, sono davvero felice di poter condividere qualche retroscena dei miei progetti con i tuoi lettori. Non sono propriamente un giovanotto, sono nato nel 1974 e la mia prima esperienza videoludica risale probabilmente al 1981, quando vidi per la prima volta un cabinato di Pacman in un bar della mia città, Sassari. Da allora i videogame sono diventati prima una ossessione, poi una passione ed infine una professione (seppur temporanea). Ora che ho smesso di giocare, per via dei tanti impegni e delle vicissitudini della mia vita, comunque coltivo l'interesse per i videogame, prevalentemente retro. E' giusto un modo per rimanere legato ad un passato vissuto con tante emozioni. Vivo a Sydney da 5 anni, dove lavoro come ingegnere.

Il mio nickname e' una scelta piuttosto casuale.

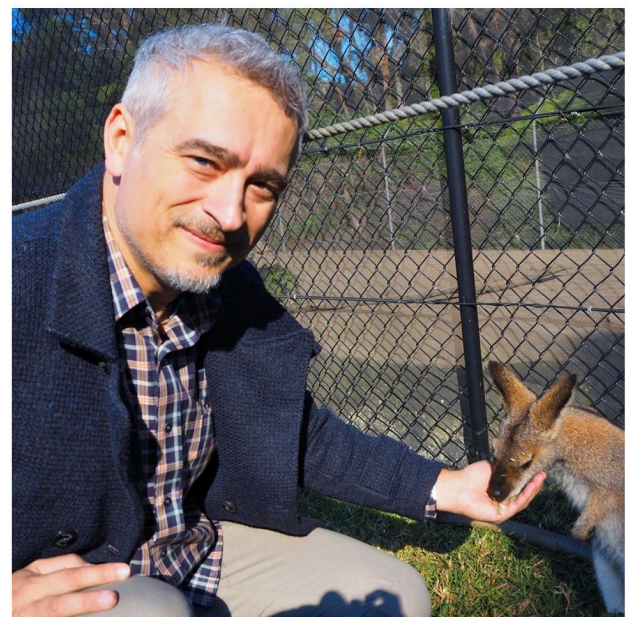


Figura 1





Madrigal era il nome del personaggio elfo che usavo quando giocavo di ruolo, ai tempi dell'università. Era semplicemente un nome che per me suonava bene, non aveva alcun significato particolare. Quando ho avviato il mio progetto, ho scelto il nickname MADrigal, con 'MAD' maiuscolo quasi a rafforzare la 'pazzia' che sentivo permeare il mio progetto. Leggi MADrigal come 'mad madrigal' (mad = pazzo, appunto).

La passione per i G&W ha radici profonde per me. Da piccolo non avevo mai avuto alcun gioco elettronico, per quanto li desiderassi parecchio. Giocavo con i giochi dei miei amici, fino a quando nel 1984 ricevetti il C64 come regalo di Natale dai miei genitori. Da allora non mi sono più interessato a quei semplici, banali giochi tascabili... fino a quando, nel 1999, non ne vidi alcuni in vendita a poche migliaia di lire in un mercatino dell'usato. Li comprai e poi mi chiesi come mai non ci fossero emulatori disponibili per questi giochi. E mi dissi: beh posso provare io a fare qualcosa, sarebbe divertente. E così per sfida e per passione verso il mondo dell'emulazione che in quegli anni viveva la sua fase più prolifica, decisi di intraprendere il mio primo progetto: '_MADrigal's Handhelds Simulators' (ora cambiato più semplicemente in 'MADrigal's Simulators'). (cfr. figura 2)

Negli anni ho collaborato attivamente a molti progetti quali MAME, Emuita.it, N! Zone (Nintendo Zone, che ho fondato e gestito per diversi anni), Zzap! Raine, Game & Watch Mania (il mio sito italiano sui G&W), Retroedicola Videoludica per cui ho realizzato lo speciale Zzap! 2015 ed altre pubblicazioni."

RMW: "Entriamo nel vivo dell'azione, potresti parlarci dei problemi più importanti che hai incontrato nei vari progetti?"

LA: "Inizio i miei progetti sempre giocando i giochi che, di volta in volta, mi procuro. Se il gioco non è divertente, lo metto via o lo scambio con altri, nella speranza di trovarne uno (a poco prezzo) che sia divertente da giocare, e stimolante per me da programmare.

Poi provo il giochino per tanto, tanto tempo, per cercare di capirne completamente le dinamiche e vedere quel che succede nei vari eventi (vincita di una vita, bonus, fine partita, intermezzi animati, etc.) e questo è abbastanza lungo, a volte tedioso, e complicato specialmente se il gioco è difficile. Una volta fatto questo, passo ad acquisire la grafica del gioco, solitamente tramite scanner.

Qui si aggiunge un'altra complessità. A volte i giochi non svelano tutta la grafica nello stesso momento, per cui devi scannerizzarlo tante volte, e ogni volta



» Read carefully and accept before downloading...

■ Attention webmasters!

Ask the author's permission before mirroring any games and files to your website. They are distributed freely from this site only, and a very few selected authorized/legal mirrors.

■ More questions?

Please take your time to [read the FAQs](#).

» Available Versions

■ Original MADrigal Releases

Produced by Luca MADrigal Antignano.

Work on every 32 and 64-bit version of **Microsoft Windows**.
32 MBytes RAM required as a minimum. Sound card is optional.

Available packages

[A] MADrigal CD collection

All games and a nice graphics interface in a single archive, available for download at the below "[Game Collections](#)" section.

[B] Standalone games

Single game archives are available for download at the below "[Single Games](#)" section.

Key features

- * Full compatibility with arcade cabinets
- * Fullscreen mode features custom-made FullHD wallpapers.
- * Windowed mode available (optional)

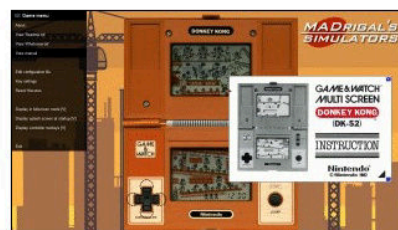


Figura 2





sperare di 'acchiappare' qualche sprite in piu'. Questo e' dovuto al fatto che i giochi con schermi a cristalli liquidi hanno i vari segmenti accesi o spenti a seconda dell'occasione. Nella migliore ipotesi, all'accensione del gioco, tutti i segmenti vengono accesi (modalita' test o 'ACL'), in altri casi no e devi farli accendere tu un po' alla volta giocandoci. O nel caso di schermetti a LED, devi aprire il gioco, tirare fuori lo schermo e scannerizzarlo separatamente, il che e' laborioso. (cfr. figura 3)

Dopodiche' le scansioni vanno ripulite e devi dare il tocco grafico alle immagini per farle sembrare reali. Sono tutte procedure lente, laboriose e complicate a volte, che vanno avanti per giorni. Il tutto ancora prima di iniziare a programmare, per cui puoi immaginare quanto impegno ci sia dietro. Una cosa particolarmente difficile da replicare e' l'intelligenza artificiale di alcuni giochi. A parte quelli in cui il gioco si comporta in modo piu' o meno casuale, ci sono giochi in cui per esempio i fantasmi inseguono Pacman, o il tuo avversario computerizzato si difende o ti attacca in base a come tu ti comporti. Ho programmato 3 giochi in cui ho dovuto replicare l'IA, ed e' sempre piuttosto complicato perche' non puoi permetterti di 'inventarla', devi farla come il giochino originale, per cui hai tanti paletti e regole da seguire. Ma e' divertente comunque!"

RMW: "Simulazione, emulazione, clonazione: sinonimi solamente apparenti! Ci vorresti descrivere meglio questi aspetti?"

LA: "Si hai ragione, in realta' sono cose talmente diverse, potremmo quasi definirle opposte! Una

interessante definizione viene data dal mitico Nicola Salmoria nella sua tesi di laurea dedicata al MAME, in cui cita il mio progetto come un esempio di 'simulazione' appunto opposto alla 'emulazione'.

Simulazione, in questo ambito, significa non curarsi dell'hardware dei giochi, ma solo del gameplay. Significa spremere il giochino originale fino a cercare di conoscerne tutti i segreti, e poi programmare (per esempio con Visual Basic o Java) tutte le routine ricreando un giochino che gli assomigli quanto piu' possibile. E questo ovviamente implica anche usare grafica per quanto possibile simile all'originale. In questo caso non mi servo delle 'ROM' del gioco originale, ma creo io le routine del gioco, programmandole ed inserendole nel simulatore stesso.

La differenza e' proprio questa: col simulatore scarichi un file ed e' pronto all'uso. Con gli emulatori invece ti servono le ROM.

Emulazione significa creare un programma che riproduce il comportamento hardware, ovvero CPU e altri processori e periferiche, del videogame originale. Una volta che questo 'ambiente virtuale' e' emulato, ci si fa girare sopra il programma (la cosiddetta 'ROM') che tipicamente era stato creato per quel particolare computer o console da gioco. E' questo il caso del MAME: ti servono le ROM perche' il MAME e' una collezione di emulatori di componenti hardware - ma senza software incluso. Senza le ROM, e' come avere tantissimi computer ma nessun programma per farli funzionare.

In senso lato, puoi chiamare 'clone' un simulatore se ti va. Ma non si 'clona' niente, si programma qualcosa che cerca di assomigliare, ma un clone significherebbe 'identico', mentre il simulatore non sara' mai identico all'originale. E neanche

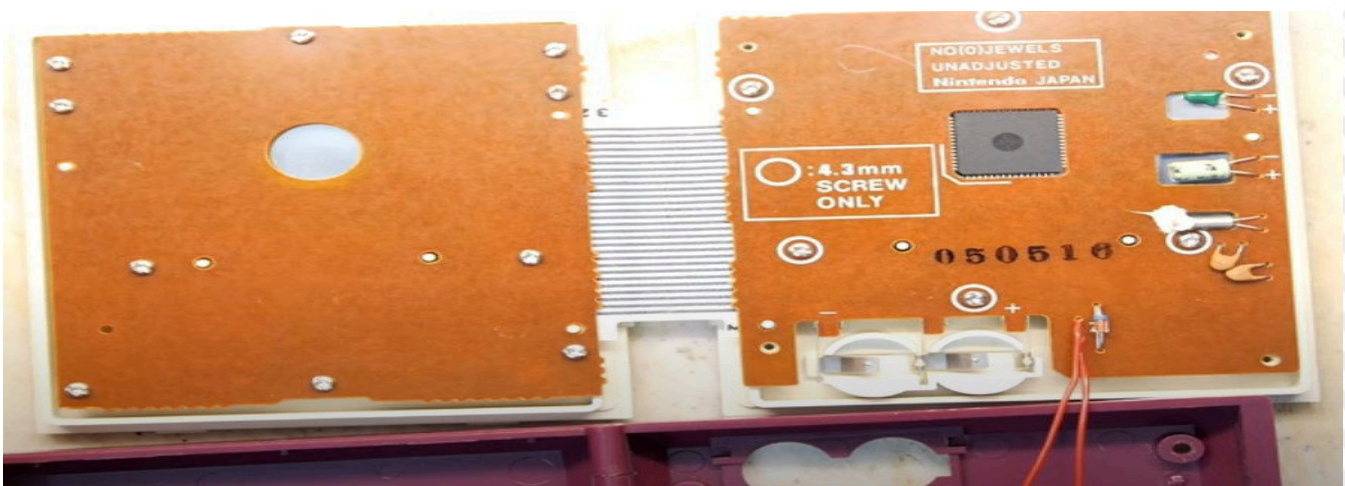


Figura 3





l'emulatore lo sarà mai al 100%, per quanto ci si avvicini parecchio di più."

RMW: "Puoi illustrarci il modello grazie al quale riesci a gestire le interazioni tra le varie componenti coinvolte? Con quale metodo riesci a trattare in maniera astratta ciascun progetto e con quali meccanismi ne gestisci le varie interazioni e/o azioni utente? Il sistema è replicabile e, se sì, in che modo e con quali eventuali limiti dal punto di vista sia delle piattaforme che della interfaccia utente?"

LA: "Programmare un simulatore di gioco elettronico non è diverso dal programmare qualsiasi altro videogame, almeno a livello di concetto. Poi naturalmente ci sono delle caratteristiche specifiche per il tipo di gioco. Un gioco elettronico ha pochi componenti rispetto ai giochi tradizionali: si tratta di segmenti a cristalli liquidi già sagomati, per esempio una fila di omini che corre, ciascuno in una posizione ben precisa sullo schermo. Tu ne accendi uno alla volta, nella sua relativa posizione, e sembra che lui stia correndo.

Nel programma, semplicemente ti preoccupi di 'decidere' quale segmento accendere e quali spegnere, ripeti questo varie volte e lo combini con la pressione dei tasti/joystick in modo che diventi interattivo. Aggiungi le musiche (i 'beep beep') ed il gioco è pronto. Per fare questo, prepari anticipatamente le routine e delle 'matrici' in cui individui i vari segmenti, questo ti rende possibile collegare gli sprite (i segmenti) ad un modello matematico di solito abbastanza semplice da gestire con un sistema di coordinate.

La caratteristica dei giochi elettronici è che non ci sono 'collisioni' tra elementi. Quando commetti un

errore giocando, è solitamente perché non hai spostato il tuo personaggio in tempo, verso una posizione salva, entro il tempo consentito. Il tempo viene scandito dai 'beep', per cui più o meno sai quanto tempo hai per agire e se non lo fai, viene segnalato un errore. Sembra complicato da spiegare, ma facilissimo una volta che lo vedi in funzione o in un video.

Il sistema certamente è replicabile, non conta il linguaggio di programmazione o dove lo giochi (cellulare, computer o console), ci sono simulatori che puoi giocare premendo direttamente i pulsanti del giochino sullo schermo (per replicare il feeling dei giochini plastici coi pulsantoni) oppure su tastiera o joystick. Questo dipende da come il simulatore è stato programmato."

RMW: "Preparazione del codice, preparazione delle immagini, preparazione dell'audio, gestione del random, ottimizzazione, ad ogni progetto permane una mole imponente di lavoro sul tavolo. Senza un rigido, anzi, oserei dire nipponico stile organizzativo, sembrerebbe impossibile poter fondere le numerose figure giapponesi che hanno fatto nascere un solo G&W. Eppure tu nel progetto MADriginal sei riuscito, pressochè da solo, a partorire decine di lavori meravigliosi (cfr. figura 4 e 5). Ti prego, parlacene del tuo stile!"

LA: "Hai ragione, oggi siamo abituati a grossi team di programmazione, grafici, musicisti, registi e produttori. Ma per un gioco elettronico tutto questo non serve. Avendo non solo programmato simulatori di giochi elettronici classici ma anche creato nuovi giochi elettronici (per una ditta americana di videogame), ho potuto in qualche modo assaporare



Figura 4



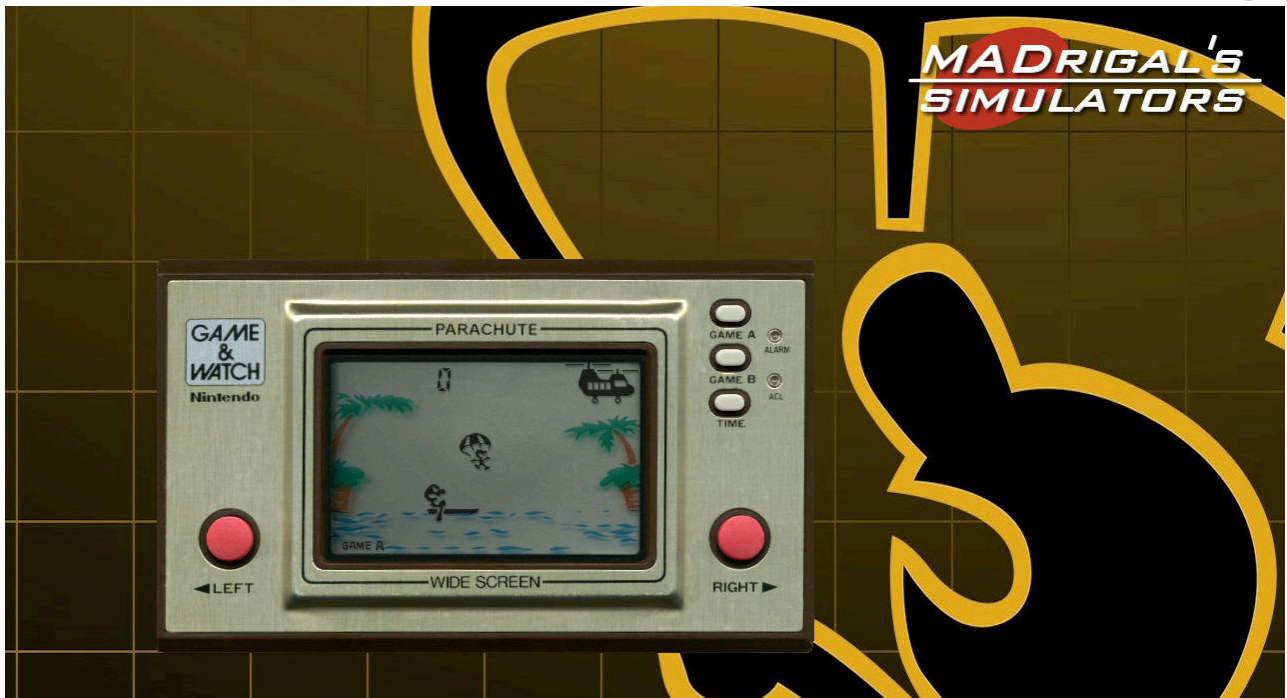


Figura 5

dal vivo la tecnica produttiva che porto' Nintendo a produrre i primi giochi LCD tascabili. Servono ingegneri del software, grafici, designer del case plastico ed uno o piu' game designer.

Nel caso dei simulatori, mi sono dovuto 'arrangiare' imparando le cose da me, combinando la programmazione (nel mio caso Borland Delphi) con l'editing grafico (con Paint Shop Pro solitamente) e audio (Audiowave o simili). Seguo piu' o meno sempre la stessa procedura: provo il gioco, se mi piace lo scannerizzo, preparo le immagini e le posiziono sulla finestra di programmazione, e poi inizio a programmare le routine di input cioe' come funzionano i pulsanti. Infine gioco ancora, registro l'audio e inserisco i vari effetti sonori nell'ambiente di programmazione. Infine programmo le routine del gioco e testo il prodotto finale finche' non mi rendo

conto di aver inserito tutte le possibili combinazioni e procedure presenti nel gioco originale.

Il mio stile e' quello del realismo piu' assoluto. Realizzo la grafica in modo che sia identica al gioco originale, con tanto di effetti di ombra nello schermo LCD, i pulsanti animati, e cosi' via. Idem per gli effetti sonori, sono sempre campionamenti di quelli originali. Mi piace molto l'idea di avere non solo lo schermo LCD giocabile, ma anche il gioco plastico intorno, coi pulsanti animati."

RMW: "Siamo nati e cresciuti con gli (al tempo) onnipotenti C64, il potere era nelle nostre mani. Ci troviamo nel 2020, periodo videoludico animato da imponenti livelli grafici, credi che oggidi vi siano prodotti dotati di dubbia anima?"

LA: "Ma guarda, penso che il mercato videoludico si



Figura 6





sia evoluto esattamente come tutti gli altri mercati nei secoli scorsi. Si comincia con una nicchia di mercato, con tanto spirito pionieristico, e poi piano piano diventa un mercato grosso in cui viene difficile inventare prodotti completamente nuovi, ma si 'potenzia' quel che già esiste. Il primo periodo della storia del videogiochi è costellata di idee, ma anche fallimenti. Le idee migliori sono sopravvissute e oggi sono la base dei nuovi videogame. Troviamo un po' di Pacman, Donkey Kong, Tetris e Space Invaders un po' in ogni videogame, anche se in diverse forme.

È difficile oggi criticare il mercato dei videogame, perché alla fine i prodotti veramente validi vendono, quelli che hanno poca qualità alla fine soccombono e non ce li ricorderemo più. Per me ben vengano i vari 'C64 mini' o 'NES mini', alla fine sono un po' frutto di operazione nostalgia, un po' un modo per tenere vivo il nostro passato e farlo conoscere alle nuove generazioni. Certamente sono operazioni commerciali, ma non dimentichiamo che quello dei videogame è un mercato che muove miliardi ogni anno, non tanto diverso dai venti film della Marvel o della Disney. Sì, alla fine possiamo stufarci, ma possiamo anche ignorare i nuovi prodotti e rifugiarci in quelli precedenti, come appunto film o videogame di qualche anno fa, sempre divertenti."

RMW: "Quanti progetti relativi ai G&W e Handheld vari hai completato, su quanti stai lavorando e su quanti ancora vorresti lavorare?"

LA: "Ho completato 60 simulatori di giochi elettronici, in realtà 59 più uno che è una versione speciale di Donkey Kong, basato sulla logica di gioco originale, ma con ben 4 vesti grafiche e sonore diverse, personalizzabile

dall'utente. Un autentico omaggio ed atto d'amore verso quello che secondo me è il gioco tascabile più emblematico di sempre.

Non sto lavorando su alcun progetto, ho smesso da qualche anno, facendo seguito al mio trasferimento in Australia, il nuovo lavoro, la nuova vita e le tante sfide che mi ritrovo ogni giorno – non ultime la situazione COVID, l'impossibilità di tornare in Italia per vedere la mia famiglia, il lavoro che cambia forma ed altre cose personali. Insomma come puoi capire, non è per me il momento di programmare videogame. Non nego che mi piacerebbe essere in una condizione più serena che mi dia stimoli a programmare nuovi giochi. In tal caso, la mia prima scelta sarebbe Mario's Cement Factory di Nintendo – un gioco elettronico che adoro e che sarebbe una sfida da programmare."

RMW: "Immagino che nella vetrina possiedi tanti G&W e handheld di tante colorazioni, marche, cloni, etc.. Sei alla ricerca di qualche prodotto in particolare? Magari qualche lettore potrebbe aiutarti nella tua caccia!"

LA: "Fino a qualche anno fa avevo parecchi giochini, ma non mi sono mai definito un collezionista. Ho comprato o scambiato giochi prevalentemente col proposito di programmare i simulatori, raramente li ho comprati solo per sfizio. Ho dato via o venduto quasi tutto, una volta che i giochi sono stati programmati, oppure ho deciso che non li avrei mai programmati (per varie ragioni), non ho avuto alcun problema a darli via.

Non sono in cerca di alcun gioco al momento. Talvolta mi vengono offerti giochi in regalo a condizione che ne programmi il simulatore – ma oramai rifiuto sempre. Il mio tempo al momento vale più di ricevere un giochino gratis."

The screenshot shows the website for CREATI/EMU, with the tagline "creatiVision emulation central". On the left, there is a "MAIN RESOURCES" menu with links to HOME/NEWS, News archive, FORUM, INTERVIEWS, LINKS, CREDITS, and CONTACT ME. The main content area is titled "NEWS" and features a post dated "12th May, 2019". The post text reads: "First post this year. Today's update is all about two topics mainly: the third part of my interview on Youtube dated 2018 and new amazing findings of Salora Manager items. Let's go straight to the first piece of news then. Again, after weeks of hard working, I was able to add Italian and English subtitles to the 3rd part of the interview that my long time friend Massimo Serpillo and I did in July 2018." There are social media share icons for Facebook, Twitter, and YouTube.

Figura 7





RMW: "Nel web si trovano tanti simpatici meme a tema G&W. Ci puoi raccontare un aneddoto divertente che ti ha meravigliato riguardo i tuoi lavori passati?"

LA: "L'aneddoto piu' particolare e' quando sono stato contattato da una delle piu' grosse aziende produttrici di giocattoli americane, e mi e' stato chiesto di lavorare per loro per ideare e produrre dei demo di giochi elettronici - che poi sono stati effettivamente prodotti e sono stati immessi nel mercato americano. Ho programmato 4 giochi di cui 2 su licenza Namco. Di questi, 2 sono stati prodotti. E' successo intorno al 2007 (se non erro). E' stato interessante e stimolante."

RMW: "CreatiVision, IntelliVision, ColecoVision, solo alcune delle console geniali degli anni 80. Come nasce il tuo impegno soprattutto con CreatiVision?" (cfr. figura 7).

LA: "Ero un avido lettore di Video Giochi negli anni '80, prima di ricevere il mio C64. Per cui conoscevo bene le varie console di quegli anni, tranne il CreatiVision, che non avevo mai potuto provare ma sembrava davvero fico.

Quando nel 1999 mi resi conto che c'erano emulatori disponibili per praticamente qualsiasi console classica, tranne il CreatiVision, ne parlai con un paio di amici. Uno aveva un CreatiVision originale, e l'altro era un abilissimo programmatore di emulatori. Unimmo i nostri sforzi e in poco tempo nacque il progetto CreatiVEmu. Dapprima come tentativo di emulare la console, e poi come progetto di conservazione che raccoglie scansioni, ROM, giochi, libri, ed infine nel 2007 ho prodotto, con un amico, la prima multi-cartuccia per CreatiVision. E nel 2009 abbiamo prodotto una cartuccia diagnostica. Da allora continuo a mantenere il progetto attivo, aggiungendo di volta in volta le informazioni che reperisco in rete. Il progetto e' enorme ed e' il punto di riferimento mondiale per quella console."

RMW: "Tanta passione, tanto amore per questi fantastici progetti, oltre alle donazioni che è possibile effettuare attraverso il tuo sito:

<http://www.madrigaldesign.it/sim/>

hai mai pensato di collaborare con una ipotetica Sony o Microsoft o altro, per lanciare sul mercato la tua genialità?"

LA: "Ma in realta' non ci ho mai voluto pensare seriamente. Mi piace l'idea che il mio sia un progetto gratuito, un hobby. Ho lavorato nel settore

informatico, ingegneristico, educativo, grafico e pubblicitario per tanti anni. Sono giunto alla conclusione che la programmazione per me deve restare un hobby, ho fatto la mia scelta professionale in favore dell'ingegneria edile. Portare avanti progetti in ambito videoludico richiede tempo, concentrazione, studio e continua evoluzione, penso che alla mia eta' e dopo i vari cambiamenti nella mia vita, sia meglio rimanere ancorato alla mia carriera attuale.

Ma non ti nego che mi fa piacere vedere che diversi programmatori si sono interessati al mio lavoro e hanno dedicato anima e cuore in progetti che hanno reso possibile 'trapiantare' i miei giochi su moderne piattaforme come Andriod, iPhone, Sony, Microsoft, Raspberry, Switch, NES Mini e cosi' via. Ora tutti i miei giochi sono giocabili su praticamente tutte le piattaforme presenti sul mercato. Questo mi rende davvero felice."

RMW: Quali potrebbero essere le future evoluzioni di questo progetto? Ne hai anche altri nel cassetto?"

LA: "Al momento non ho altri progetti, tranne quello di mantenere attivo quel che gia' ho fatto e supportare i nuovi programmatori che mi contattano per supporto o per ampliare quel che ho gia' fatto. Stessa cosa per quanto riguarda il CreatiVEmu. In futuro non si puo' mai sapere..."

Bene cari lettori, vi ringraziamo per l'attenzione e soprattutto ringraziamo MADrigal per aver risposto alle domande dell'intervista.

Se vorrete approfondire il discorso troverete tantissimo materiale adorabile in:

www.madrigaldesign.it

Oltretutto nel sito sono contenuti tre voluminosi progetti. N!Zone : archivio del sito, chiuso nel 2006 ma tutto visitabile e in italiano (cfr. figura 6). Game & Watch Mania : dedicato ai G&W interamente in italiano. CreatiVEmu : attivo, in inglese, con immenso database, emulatori, ROM , forum ed altro sul CreatiVision. Mi raccomando infine di rimanere sintonizzati sul discorso G&W, infatti se tutto andrà realmente bene e la produzione globale ripartirà, a breve ci saranno grandi sorprese di cui vorrei parlare nel prossimo numero o quello seguente. A presto!





Decuplicare facilmente i giochi disponibili nel THE64? Si può fare!

di Marco Pistorio



Il progetto "THE 64", apprezzato da tanti e bistrattato da molti altri, continua ad andare avanti a gonfie vele.

Oltretutto, e' notizia di pochi giorni fa il lancio ormai prossimo di una versione del dispositivo che ricalca le caratteristiche e le dimensioni reali del Commodore VIC-20 e che si chiamerà "The VIC20". Ciò a riprova del fatto che la società Retrogames Ltd punti ancora sul progetto e sforna quindi una ulteriore variante del "THE 64", già presente sul mercato nelle versioni "Maxi" e "Mini".

Questo articolo non verterà sulla bontà del progetto in sé, sul fatto che il "THE 64" debba essere necessariamente un oggetto gradito ai collezionisti del "commie" reale oppure no, nè insisterò sul fatto che, a mio avviso, qualsiasi cosa riesca a far rivivere l'emozione di giocare con un Commodore 64, che porti ancora alla ribalta il Commodore 64 presentandolo a vecchi e soprattutto a nuovi utenti sia qualcosa che merita, a prescindere da tutto, di essere sempre e comunque accettata e promossa.

In questo articolo invece mi soffermerò sui titoli a disposizione all'interno del "THE 64" ed, in particolare, su come far si che diventino quasi dieci volte di più dei 64 normalmente a disposizione.

In passato mi è capitato di leggere qualcosa in merito, ma tutte le soluzioni proposte in tal senso andavano a modificare in maniera permanente il dispositivo e costringevano ad un accesso fisico al suo hardware interno, invalidando ovviamente qualsiasi tipo di garanzia.

La soluzione invece di cui vi parlo oggi centra l'obiettivo di mettere a disposizione ben 623 titoli e non prevede alcun intervento invasivo nè alcuna modifica permanente al "THE 64" e funziona con entrambi i modelli del "THE 64" in commercio, cioè sia sul "THE 64 Maxi" che su quello Mini. Mica male, vero?

L'idea è grossomodo questa:

Si predispongono una pendrive USB con i giochi, snapshots dei giochi etc. insieme ad un aggiornamento del firmware "fake" del "The 64". Si accende il dispositivo, si applica il falso aggiornamento e, alla sua riaccensione, troveremo i nuovi giochi in un nuovo carosello.

I nuovi caroselli predisposti sono in realtà 4.

In ciascun carosello possono essere ospitati al massimo 226 giochi.

C'è un primo carosello di giochi preferiti.

Quindi un carosello di giochi i cui titoli iniziano con un carattere "Da O a G", con 222 giochi, un carosello di giochi i cui titoli iniziano con un carattere "Da H a R", con 219 giochi ed infine un carosello di giochi i cui titoli iniziano con un carattere "Da S a Z" con ulteriori 182 giochi, per un totale complessivo di 623 giochi.

E' possibile navigare da un carosello a quello successivo spostandosi sulla icona della "chiave inglese" e poi da lì puntando sul nome del carosello sul quale spostarsi e scegliendo di caricarlo.

Dopo aver giocato con soddisfazione, spegneremo il dispositivo e potremo rimuovere in sicurezza la pendrive USB. Riaccendendo il dispositivo tornerà tutto come prima, senza aver apportato alcuna modifica permanente!

E' possibile scaricare la cartella contenente tutto il necessario già predisposto seguendo il link all'interno della pagina del progetto "Project Carousel USB by Spannernick", a questo indirizzo:

<https://thec64community.online/thread/501/project-carousel-usb>

previa però una semplice registrazione all'interno del forum "The C64 Community". Attenzione! Il progetto è FREE per uso non commerciale. Si estrae quindi il contenuto del file .zip all'interno di una pendrive USB a vostra disposizione e... Il gioco è fatto :)

Personalmente trovo il tutto davvero comodo ed intuitivo.

Esiste anche la possibilità di cambiare il contenuto dei giochi messi a disposizione grazie ad un tool che si trova all'interno delle cartelle che contengono ciascun carosello.

Il nome dell'eseguibile da lanciare è "TheC64MaxiGameTool.exe"

Per ulteriori dettagli relativi anche a come personalizzare i vari caroselli, aggiungerne di nuovi e molto altro ancora vi suggerisco di leggere con attenzione il contenuto della pagina ufficiale che potrete raggiungere seguendo il link già fornito.

Buon divertimento e... Un saluto a tutti!

PROJECT CAROUSEL USB





RITORNO AL... PASSATO - Tappa nr. 2: Windows 2000

di Marco Fiaschi

Ben ritrovati in questo entusiasmante viaggio virtuale tra i sistemi operativi che ci hanno fatto compagnia per mezzo decennio della nostra vita. Sicuramente vi starete chiedendo come mai questa seconda tappa fa capolinea in Windows 2000, anziché in Windows ME (la famosa Millennium Edition). Questo nostro tour in realtà, non fa fede alle versioni del kernel di Windows, ma alla data di release dei vari software. Prima di introdurvi Windows 2000, voglio rispondere ad alcune domande che ho ricevuto in seguito alla pubblicazione dell'articolo riguardante la virtualizzazione di Windows 98 su RetroMagazine 22 (per chi se lo fosse perso, vi invito a recarvi sul sito ufficiale per scaricarne una copia). Procediamo quindi ad elencarvi alcuni dei quesiti maggiormente ricorrenti:

D – Come mai Windows 98 ha una brutta grafica su Virtualbox?

R – Purtroppo i driver che ci offre Virtualbox (le GuestAdditions) non sono disponibili per la serie di sistemi operativi Windows 9x, pertanto è necessario installare driver di terze parti direttamente sul Windows emulato.

D – Come mai l'emulazione di Windows 98 è lentissima?

R – La risposta è nella domanda. Un'emulazione è pur sempre un'emulazione. E' impensabile ottenere le stesse prestazioni che possiamo ottenere da un'installazione su disco fisico di un sistema operativo. Nelle impostazioni del sistema virtualizzato, proviamo ad abilitare almeno l'accelerazione grafica 3D, portando la memoria dedicata al massimo consentito (128MB o 256MB).

D – Quale driver video è meglio selezionare nelle impostazioni?

R – Virtualbox mette a disposizione ben 4 driver che consentiranno al sistema operativo di riconoscere schede grafiche fittizie. Essi sono:

VboxSVGA, che rappresenta la scelta ottimale per sistemi operativi Windows 7 e successivi. VMSVGA viene scelta di default quando si ha a che fare con sistemi operativi Linux, e rappresenta un'emulazione del driver Vmware SVGA (incorporato proprio nel kernel Linux). Infine troviamo il driver VboxVGA, che rappresenta l'ultima spiaggia su cui scegliere di approdare, questo perché è un driver molto obsoleto supportato però da Windows Vista e precedenti. Questo vuol dire che per i sistemi operativi presi in esame in questi articoli (Windows 98, Windows 2000 e Windows ME che troverete nel prossimo numero di RetroMagazine), è il driver più adatto che si

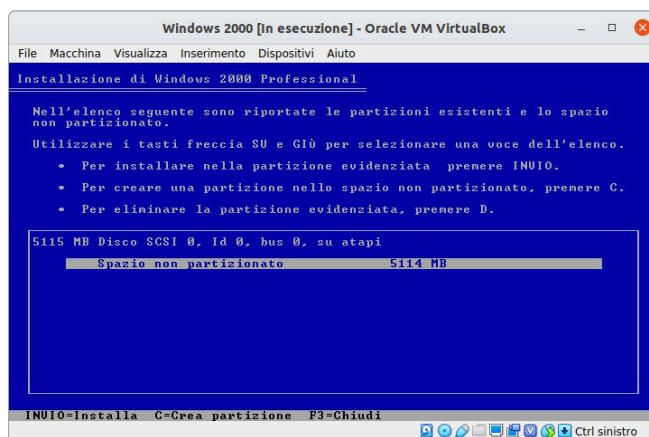


Figura 2 - Installazione S.O.

possa scegliere. Attenzione però...Non è che scegliendo questo driver, la grafica dei sistemi aumenterà a 32bit. Bisogna comunque installare driver esterni.

Chiusa la parentesi delle domande, partiamo per la seconda tappa del nostro viaggio. Rilasciato il 17 febbraio 2000 sotto i nomi di Odyssey e poi Neptune (o meglio NT 5.0), Windows 2000 è stato un sistema operativo appartenente alla famiglia Windows NT (New Technology), basato interamente su kernel ibrido a 32 bit, il cui supporto è terminato il 13 luglio 2010. Windows 2000, a differenza delle edizioni 95, 98 e ME, non fu concepito come sistema operativo "domestico", ovvero destinato ai clienti, ma fu ideato per venire incontro alle esigenze dei professionisti e delle aziende. Windows 2000 infatti non proviene dalla famiglia Windows 9x, ma dalla famiglia NT, alla quale appartengono tutte le edizioni Windows per i server e per le workstation, tra cui NT 3.1, NT 3.5, NT 4.0, fino ad arrivare all'attuale Windows 10. Questo sistema operativo introdusse per la prima volta in assoluto un nuovo tipo di filesystem, ovvero l'NTFS (standard tuttora largamente utilizzato), il supporto all'EFS (Encrypting File System), un nuovo tipo di kernel completamente riscritto sulla base del predecessore Windows NT 4.0 per quanto riguarda l'interfaccia grafica e la gestione dei protocolli di rete, l'Active Directory e il BITS (Background Intelligent Transfer System). Windows 2000 ha avuto un largo impiego in azienda, ma una scarsissima diffusione tra i consumatori, a causa della sua più totale incompatibilità con i videogiochi. In poche parole, questo sistema era ideale per poter eseguire programmi di tipo professionale per la grafica,

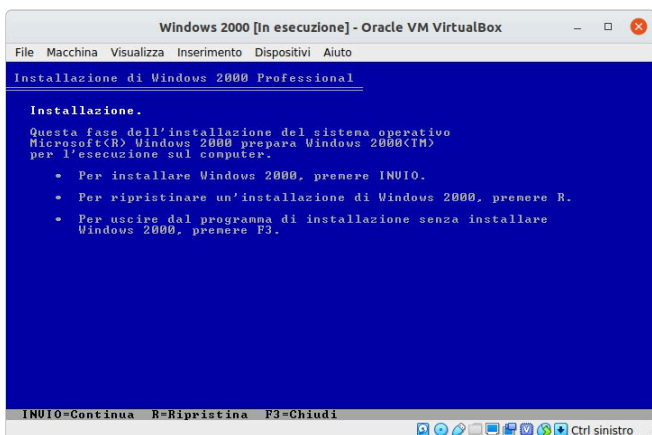


Figura 1 - Partiamo con l'installazione



Figura 3 - Formattazione della partizione



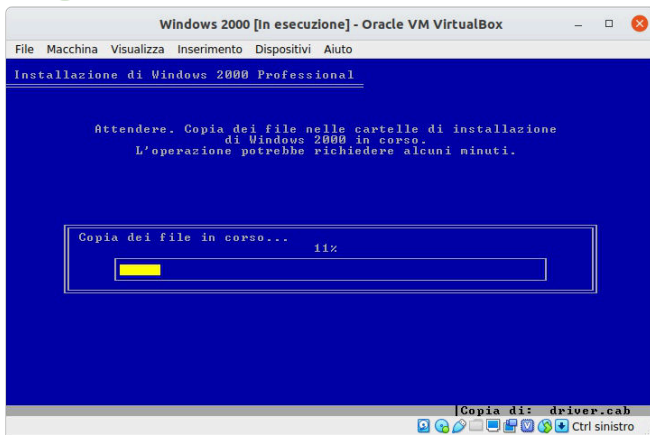


Figura 4 - Copia dei file di Win2k in corso

per il calcolo matematico, per CAD e per workstation (computer ad alte prestazioni con una bassissima latenza della CPU per spingere al massimo la potenza del processore fino all'ultimo Hz). Windows 2000 venne rilasciato in ben 4 varianti che comprendono anche Windows 2000 Server, Advanced Server e Datacenter Server. Quasi sconosciute sono invece le versioni a 64 bit di Windows 2000 sviluppate per processori Itanium e Itanium 2 di Intel. Windows 2000 è tuttora considerato uno dei sistemi operativi più sicuri mai sviluppati da Microsoft...Non a caso Windows 10 oggi ne è considerato ugualmente e, l'edizione Pro for Workstation ricorda alla lontana proprio Windows 2000 a causa di alcune funzionalità implementate nel sistema operativo. Un'ulteriore curiosità di questo sistema operativo è la sua versione del kernel che ricordo essere la quinta, gli altri sistemi consumer come Windows 95, 98 e ME sono sviluppati con versione del kernel 4. Infatti è possibile aggiornare Windows ME a 2000, ma non il contrario, anche se 2000 è uscito prima di ME.

Terminata questa ampia introduzione al sistema, corriamo subito a vedere come installare Windows 2000 sul nostro Virtualbox. Clicchiamo sulla voce "Nuova" nel menù del software, per poi inserire il nome da dare alla nostra macchina virtuale, scegliendo "Microsoft Windows" alla voce "Tipo" e "Windows 2000" alla voce "Versione". A questo punto VirtualBox ci consiglia 168MB di memoria RAM da destinare alla macchina virtuale, ma noi ne assegneremo 512 MB per un'emulazione più performante. Scegliamo poi il tipo del file da utilizzare per il disco fisso virtuale che sarà per nostra comodità VDI (Virtualbox Disk Image) allocato dinamicamente. Quando il software ci chiederà lo spazio da destinare al disco, noi inseriremo



Figura 5 - Avvio di Windows 2000 Professional

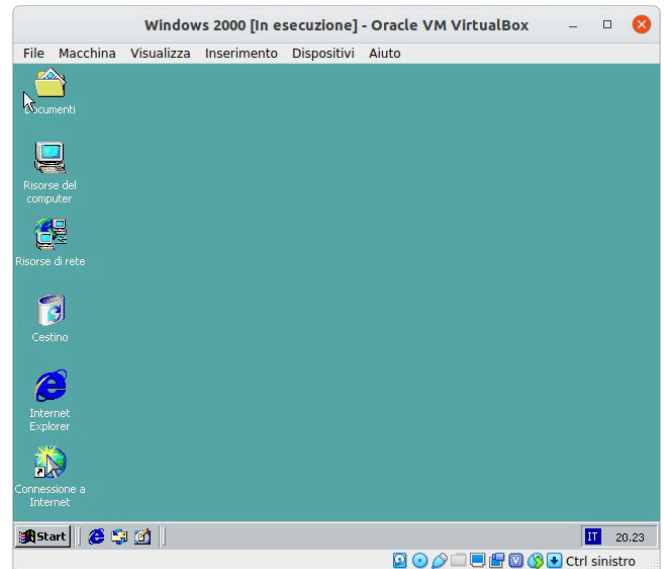


Figura 6 - Windows 2000 in esecuzione

il valore di 5GB (nonostante Windows 2000 ne richieda meno di 700MB), ma il tutto ci permetterà di beneficiare di una migliore emulazione. Fatto ciò, abbiamo creato la macchina virtuale che potremo avviare con un doppio click sul suo nome. Selezioniamo il lettore host dal quale avviare il sistema operativo e procediamo con l'installazione (figura 1) che comincerà con il caricare tutti i driver e i file necessari per il sistema. Clicchiamo su Invio quando Windows ce lo chiederà, accettiamo il contratto di licenza Microsoft e di nuovo Invio per selezionare la partizione che ci suggerisce l'installer stesso (figura 2). Piccola curiosità...Avete notato che il nome dell'installer è "Installazione di Windows 2000 Professional"? Tranquilli, non avete sbagliato edizione e/o versione! Non esiste un'edizione Home di Windows 2000, ma è una unica...ed è quella Pro. Nella schermata successiva (figura 3) formattiamo la partizione con il nuovo filesystem che ha introdotto Microsoft, ovvero l'NTFS e attendiamo la copia dei file di Windows 2000 sul nostro disco rigido virtuale (figura 4). Una volta riavviato il sistema, potremo subito notare una rinnovatissima grafica dell'installazione del sistema operativo che tuttora viene utilizzata per i pacchetti d'installazione .MSI. Successivamente ci toccherà impostare la data e l'ora, il layout di tastiera, e poi dovremo inserire il nome utente, il product key (disponibile sul manuale di Windows 2000), il nome del computer, la password e infine attendere il completamento dell'installazione. Quando meno ce l'aspetteremo, comparirà una finestra che ci comunica che l'installazione di Windows 2000 è andata a buon fine ed è stata completata. Al riavvio del sistema, potremo goderci la nuova schermata d'avvio di Windows 2000 che contiene la dicitura "Sviluppato con tecnologia NT" (figura 5), e poi il sistema sarà pronto all'uso (figura 6). Immediatamente noteremo un rinnovamento nelle icone del desktop, anche se non potremo beneficiare di una grafica accattivante poiché Virtualbox, come detto in precedenza, non supporta questi sistemi operativi (purtroppo il supporto comincia da XP), a meno che non installate driver di terze parti.

Purtroppo il viaggio verso la seconda meta termina qui, ma vi invito a cercare alcuni siti web dai quali potete scaricare un'infinità di software e di giochi abandonware... E' vero che Windows 2000 non è stato progettato per funzionare con i videogiochi, ma provare non fa mai male no?

Alla prossima tappa ;)





Caratteri ridefiniti per Amstrad CPC

di Francesco Fiorentini

Antefatto

Qualche giorno fa mi imbatto in un post di **Arturo Dente** su **Retro Programming Italia (RPI) by The Nerds**, contenente una rivista in formato pdf a beneficio di tutto il gruppo. Arturo aveva voluto condividere quella particolare rivista al fine di sollecitare altri membri del gruppo a recuperare un pezzo di programma basic per Commodore 64, atto a caricare in memoria un codice LM per splittare lo schermo in due parti, una hires e una testuale... Una funzionalita' molto utile se si volesse, per esempio, scrivere un'avventura testuale con un po' di grafica a corredo.

Spinto dalla curiosita' apro il post e vedo che gia' qualcun altro della redazione, **David La Monaca**, aveva preso a cuore questa operazione.

Apro comunque la rivista per curiosare sul codice, ma subito vengo attratto da un altro programmino che si trova un paio di pagine dopo quello indicato da Arturo.

Future Set on Amstrad

Il nome mi colpisce immediatamente, lasciando poco spazio all'immaginazione. Si tratta quasi sicuramente di una serie di istruzioni per ridefinire il set di caratteri dell'Amstrad CPC per modificarne il look & feel. Un rapido sguardo al codice, fugge immediatamente ogni dubbio. E' un vero e proprio set di caratteri ridefiniti da utilizzare nei nostri programmi in Locomotive Basic.

Ho subito un'idea! Nel numero 23 abbiamo portato l'avventura testuale *Astronave Farmer* in Locomotive Basic. Si tratta di un gioco che ci vede a bordo di un'astronave alla deriva nello spazio... Quale migliore occasione per sfruttare un set di caratteri futuristici? Decido quindi di mettermi all'opera per recuperare questo pezzo di storia a firma di **Pete White** per dargli una seconda vita.

Recupero del codice

Ovviamente se il codice fosse stato selezionabile e copiabile dal pdf allegato, Arturo non avrebbe mai chiesto il supporto di altri utenti del gruppo... Quindi, anche in questo caso, l'unica soluzione era quella di riscrivere il codice a mano. Ma io sono estremamente pigro e l'idea di mettermi a scrivere una per una un'ottantina di righe di codice piene zeppe di numeri, non mi ha mai sfiorato nemmeno lontanamente nell'anticamera del cervello. Anche perche' molti dei numeri sono quasi illeggibili. Siamo nel 2020, possibile che non ci sia un modo piu' pratico?

Ma certo che c'e'! Un **OCR**, un riconoscitore di caratteri! Decido quindi di provare questa strada.

Cerco online un riconoscitore di caratteri gratuito e tra i tanti scelgo <https://ocr.space/>.

Con lo snipping tool creo 3 immagini, una per ogni colonna del codice e provo a dare la prima in pasto all'OCR.

Il primo risultato e' deludente oltre ogni aspettativa. Pochissimi numeri riconosciuti e testo illeggibile.

Armeggio un po' con i settaggi ed utilizzo il motore OCR 2 che, secondo il sito, e' piu' efficiente nel riconoscimento dei numeri.

Questa volta il risultato e' decisamente accettabile. Circa un 60% del testo e' stato riconosciuto. Devo soltanto fare delle sostituzioni nel testo per trasformare le virgole che erano state interpretate come punti, correggere a mano i numeri errati e correggere a sua volta tutte le occorrenza errate del comando SYMBOL. Un lavoraccio, ma sicuramente piu' veloce che scrivere tutto da zero.

Abbiamo finito?

Ho corretto tutto il codice, sono soddisfatto! Eseguo il

Future Set on Amstrad

by Pete White

The following routine produces a futuristic character set which can easily be incorporated into your own programs. The characters are based on the Data 70 set which frequently crops up in films using 'computer print'.

30 SYMBOL 49,16,49,16,24,24,126,0	180 SYMBOL 79,126,66,66,98,98,126,0	420 SYMBOL 109,0,0,126,90,90,66,66,0
40 SYMBOL 65,126,66,66,126,98,98,98,0	190 SYMBOL 80,126,66,66,126,96,96,0	430 SYMBOL 110,0,0,108,114,98,98,98,0
50 SYMBOL 66,126,66,66,126,99,98,126,0	200 SYMBOL 81,126,66,66,98,98,106,126,4	440 SYMBOL 111,0,0,126,102,102,102,126,0
60 SYMBOL 67,126,64,64,96,96,96,126,0	210 SYMBOL 82,126,66,66,126,106,100,98,0	450 SYMBOL 112,0,0,126,98,98,126,96,96
70 SYMBOL 69,254,66,66,98,98,98,254,0	220 SYMBOL 83,126,64,64,126,6,6,126,0	460 SYMBOL 113,0,0,126,70,70,126,6,6
80 SYMBOL 69,126,64,64,120,96,96,126,0	230 SYMBOL 84,126,16,16,24,24,24,24,0	470 SYMBOL 114,0,0,108,114,96,96,96,0
90 SYMBOL 73,126,64,64,120,96,96,96,0	240 SYMBOL 85,66,66,66,98,98,98,126,0	480 SYMBOL 115,0,0,126,96,126,6,126,0
100 SYMBOL 71,126,64,64,102,98,98,126,0	250 SYMBOL 86,66,66,66,66,66,36,24,0	490 SYMBOL 116,24,62,24,24,24,24,30,0
110 SYMBOL 72,66,66,66,126,98,98,98,0	260 SYMBOL 87,66,66,66,98,106,106,126,0	500 SYMBOL 117,0,0,102,102,102,102,126,0
120 SYMBOL 73,60,16,16,24,24,24,60,0	270 SYMBOL 88,102,102,36,24,36,102,102,0	510 SYMBOL 118,0,0,102,102,102,60,24,0
130 SYMBOL 74,126,0,0,24,24,24,120,0	280 SYMBOL 89,66,66,126,16,24,24,24,0	520 SYMBOL 119,0,0,66,66,90,90,126,0
140 SYMBOL 75,68,68,68,120,100,100,100,0	290 SYMBOL 90,126,4,8,16,32,64,126,0	530 SYMBOL 120,0,0,198,104,16,104,198,0
150 SYMBOL 76,64,64,64,96,96,96,126,0	300 SYMBOL 97,0,0,126,6,126,70,126,0	540 SYMBOL 121,0,0,102,102,102,126,6,126,0
160 SYMBOL 77,126,74,74,98,98,98,98,0	310 SYMBOL 98,96,96,96,126,98,98,126,0	550 SYMBOL 122,0,0,126,12,24,48,126,0
170 SYMBOL 78,98,82,74,102,98,98,98,0	320 SYMBOL 99,0,0,126,96,96,96,126,0	560 SYMBOL 50,126,2,2,126,96,96,126,0
	330 SYMBOL 100,6,6,6,126,70,70,126,0	590 SYMBOL 51,126,2,2,30,6,6,126,0
	340 SYMBOL 101,0,0,126,98,126,96,126,0	600 SYMBOL 52,96,96,96,96,104,126,98,0
	350 SYMBOL 102,60,48,48,120,48,48,48,0	610 SYMBOL 53,126,64,126,6,6,6,126,0
	360 SYMBOL 103,0,0,126,70,70,126,6,126,0	620 SYMBOL 54,126,64,64,126,98,98,126,0
	370 SYMBOL 104,96,96,96,126,98,98,98,0	630 SYMBOL 55,126,2,4,62,16,32,64,0
	380 SYMBOL 105,24,0,24,24,24,24,0	640 SYMBOL 56,126,66,66,126,66,66,126,0
	390 SYMBOL 106,6,0,6,6,6,6,126,0	650 SYMBOL 57,126,66,66,126,6,6,6,0
	400 SYMBOL 107,96,96,102,108,120,108,102,0	660 SYMBOL 48,126,102,110,110,102,102,126,0
	410 SYMBOL 108,24,24,24,24,24,24,0	670 SYMBOL 95,0,255,0,0,0,0,0,0





programma e mi accorgo che alcuni caratteri sono illeggibili... Eppure mi sembrava di aver interpretato correttamente i numeri, anche quelli quasi illeggibili. Evidentemente mi sbagliavo, oppure anche nel codice iniziale c'erano degli errori.

Devo quindi correggere i caratteri errati. Punto il mio browser all'indirizzo <http://xlr8.at/8x8hexbin/> e controllo i caratteri che non mi convincono. Dopo un po' di prove, le correzioni mi soddisfano e dichiaro concluso il mio lavoro di recupero.

Astronave Farmer Enhanced

Rinumero le righe del codice che ho recuperato e le aggiungo in coda al codice di Astronave Farmer. Utilizzando un GOSUB (vai a subroutine) faccio eseguire la parte del codice che ridefinisce i caratteri, prima che il programma stampi qualcosa a schermo e... Ci siamo! L'avventura testuale ha il suo carattere 'Future Set'.

Il gioco **Astronave Farmer Enhanced** potete scaricarlo dal sito di RetroMagazine World a questo indirizzo: www.retromagazine.net/download/Farmer_Enhanced.dsk

Il gioco e' listabile, quindi potete tranquillamente controllarne il codice, ma se lo vorrete in formato txt, contattatemi.

Ah dimenticavo. Non e' finita qui... La versione enhanced e' corredata anche di uno splash screen in formato SCR. Volete sapere come realizzarne uno? Non perdetevi il prossimo articolo.

Ecco il codice dei caratteri ridefiniti.

```

10 REM *****
11 REM * Future Set on Amstrad CPC
12 REM * original code by Pete White
13 REM * Popular Computing Weekly 7-13 August 1983
14 REM *
15 REM * Typed and corrected by
16 REM * Francesco Fiorentini on June 2020
17 REM * RetroMagazine World July 2020
18 REM *****
20 SYMBOL AFTER 32
30 REM Upper case chars
40 SYMBOL 65,126,66,66,126,98,98,98,0
50 SYMBOL 66,126,66,66,126,98,98,126,0
60 SYMBOL 67,126,64,64,96,96,96,126,0
70 SYMBOL 68,254,66,66,98,98,98,254,0
80 SYMBOL 69,126,64,64,120,96,96,126,0
90 SYMBOL 70,126,64,64,120,96,96,96,0
100 SYMBOL 71,126,64,64,102,98,98,126,0
110 SYMBOL 72,66,66,66,126,98,98,98,0
120 SYMBOL 73,60,16,16,24,24,24,60,0
130 SYMBOL 74,126,8,8,24,24,24,120,0
140 SYMBOL 75,68,68,68,120,100,100,100,0
150 SYMBOL 76,64,64,64,96,96,96,126,0
160 SYMBOL 77,126,74,74,98,98,98,98,0
170 SYMBOL 78,98,82,74,102,98,98,98,0
180 SYMBOL 79,126,66,66,98,98,98,126,0
190 SYMBOL 80,126,66,66,126,96,96,96,0
200 SYMBOL 81,126,66,66,98,98,106,126,4
210 SYMBOL 82,126,66,66,126,106,100,98,0
220 SYMBOL 83,126,64,64,126,6,6,126,0
230 SYMBOL 84,126,16,16,24,24,24,24,0

```

```

240 SYMBOL 85,66,66,66,98,98,98,126,0
250 SYMBOL 86,66,66,66,66,66,36,24,0
260 SYMBOL 87,66,66,66,98,106,106,126,0
270 SYMBOL 88,102,102,36,24,36,102,102,0
280 SYMBOL 89,66,66,126,16,24,24,24,0
290 SYMBOL 90,126,4,8,16,32,64,126,0
295 REM Lower case chars
300 SYMBOL 97,0,0,126,6,126,70,126,0
310 SYMBOL 98,96,96,96,126,98,98,126,0
320 SYMBOL 99,0,0,126,96,96,96,126,0
330 SYMBOL 100,6,6,6,126,70,70,126,0
340 SYMBOL 101,0,0,126,98,126,96,126,0
350 SYMBOL 102,60,48,48,120,48,48,48,0
360 SYMBOL 103,0,0,126,70,70,126,6,126
370 SYMBOL 104,96,96,96,126,98,98,98,0
380 SYMBOL 105,24,0,24,24,24,24,24,0
390 SYMBOL 106,6,0,6,6,6,6,126
400 SYMBOL 107,96,96,102,108,120,108,102,0
410 SYMBOL 108,24,24,24,24,24,24,24,0
420 SYMBOL 109,0,0,126,90,90,66,66,0
430 SYMBOL 110,0,0,108,114,98,98,98,0
440 SYMBOL 111,0,0,126,102,102,102,126,0
450 SYMBOL 112,0,0,126,98,98,126,96,96
460 SYMBOL 113,8,0,126,70,70,126,6,6
470 SYMBOL 114,0,0,108,114,96,96,96,0
480 SYMBOL 115,0,0,126,96,126,6,126,0
490 SYMBOL 116,24,62,24,24,24,24,30,0
500 SYMBOL 117,0,0,102,102,102,126,0
510 SYMBOL 118,0,0,102,102,102,60,24,0
520 SYMBOL 119,0,0,66,66,90,90,126,0
530 SYMBOL 120,0,0,198,104,16,104,198,0
540 SYMBOL 121,0,0,102,102,102,126,6,126
550 SYMBOL 122,0,0,126,12,24,48,126,0
555 REM Numbers
560 SYMBOL 48,126,102,110,118,102,102,126,0
570 SYMBOL 49,24,56,24,24,24,24,126,0
580 SYMBOL 50,126,2,2,126,96,96,126,0
590 SYMBOL 51,126,2,2,30,6,6,126,0
600 SYMBOL 52,96,96,96,96,104,126,8,8
610 SYMBOL 53,126,64,126,6,6,6,126,0
620 SYMBOL 54,126,64,64,126,98,98,126,0
630 SYMBOL 55,126,2,4,62,16,32,64,0
640 SYMBOL 56,126,66,66,126,66,66,126,0
650 SYMBOL 57,126,66,66,126,6,6,6,0
680 SYMBOL 95,0,255,0,0,0,0,0,0
1000 CLS
1011 PRINT "Future Set on Amstrad CPC"
1012 PRINT "original code by Pete White"
1013 PRINT "Popular Computing Weekly 7-13 Aug 1983"
1014 PRINT ""
1015 PRINT "Typed and corrected by"
1016 PRINT "Francesco Fiorentini on June 2020"
1017 PRINT "RetroMagazine World July 2020"

```

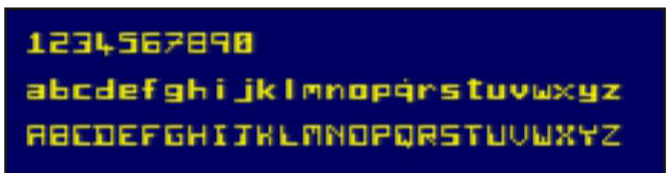


Figura 1 - I caratteri del 'Futuristic Set'





Uno splash screen per Amstrad CPC in formato SCR

di Francesco Fiorentini

Questo articolo e' la continuazione logica dell'articolo precedente. Ho preferito fare due articoli piuttosto che uno solo, per comodita' dei lettori, che potrebbero essere interessati, anche con il passare del tempo, ad un solo argomento.

Un'idea brillante...

Dopo aver modificato il set di caratteri sul gioco Astronave Farmer, mi sono accorto che mancava ancora qualcosa.

L'idea di un'avventura testuale ambientata nello spazio e' interessante, i caratteri ridefiniti aggiungono senza ombra di dubbio un pizzico di pepe al codice, ma secondo il mio modesto parere mancava ancora qualcosa per personalizzare il tutto in maniera efficace.

Sarebbe davvero interessante aggiungere, sopra la descrizione dei luoghi visitati, una piccola immagine che ne raffigurasse le fattezze e trasformare cosi' un'avventura completamente testuale in un'avventura testuale con grafica.

Ovviamente un progetto del genere e' piuttosto ambizioso e richiede del tempo per essere completato con successo, senza contare che, personalmente, sono negato per il disegno.

Aggiungiamo inoltre che le mie conoscenze di grafica in ambiente Amstrad CPC sono pari a zero ed avrete un'idea piuttosto fedele dell'ambizioso obiettivo che mi era passato per la testa.

Decido quindi di procedere per gradi. Tanto per cominciare proviamo ad aggiungere un'immagine statica come splash screen durante il caricamento del gioco.

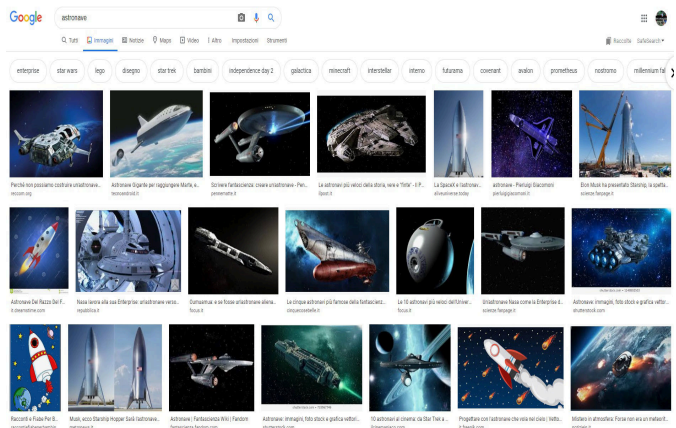
Astronave

Ok, adesso che avevo definito un approccio, il meno era fatto. Dovevo solo disegnare un'immagine che in qualche modo rappresentasse un'astronave nello spazio.

Facile a dirsi, ma piuttosto difficile a farsi in pratica se, come me, siete veramente negati in disegno ed il massimo a cui potete aspirare e' la rappresentazione stilizzata di qualsiasi oggetto reale.

Per fortuna che siamo nel 2020 e possiamo facilmente trovare milioni di immagini da cui prendere ispirazione

con una semplice ricerca su Google. Apro il browser e digito semplicemente la parola 'astronave'. Immediatamente appaiono ai miei occhi decine di immagini bellissime.



Sono tutte belle, ma la seconda immagine mi colpisce in maniera particolare. Decido che sara' lo splash screen del gioco Astronave Farmer Enhanced.



Primi problemi

Perfetto, abbiamo un candidato per la nostra immagine di introduzione, ma... L'immagine e' una JPG con una risoluzione di 660 x 371 pixel, una risoluzione di 96 dpi ed una profondita' di colore di 24 bit. Niente di che al giorno d'oggi, ma decisamente troppo per il nostro Amstrad CPC.

Devo trovare il modo di trasformare un'immagine JPG in un qualcosa che il nostro Arnold (cosi' era affettuosamente conosciuto negli anni 80) possa digerire e possibilmente visualizzare a video.

Faccio un po' di ricerche e sembra che il formato per antonomasia sia il SCR. Il formato SCR non e' altro che la rappresentazione fisica della memoria del nostro CPC





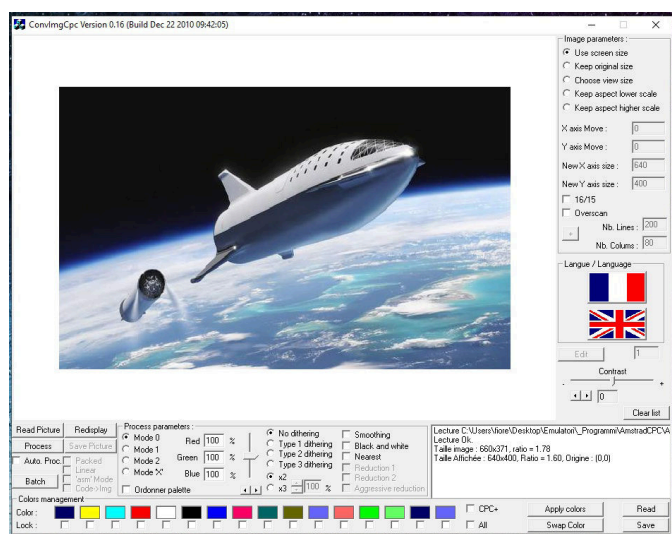
che potremo poi ricaricare usando il comando **LOAD** del Locomotive Basic.

Le mie ricerche di un convertitore mi portano a provare tra gli altri, un programma chiamato **ConvImgCPC**.

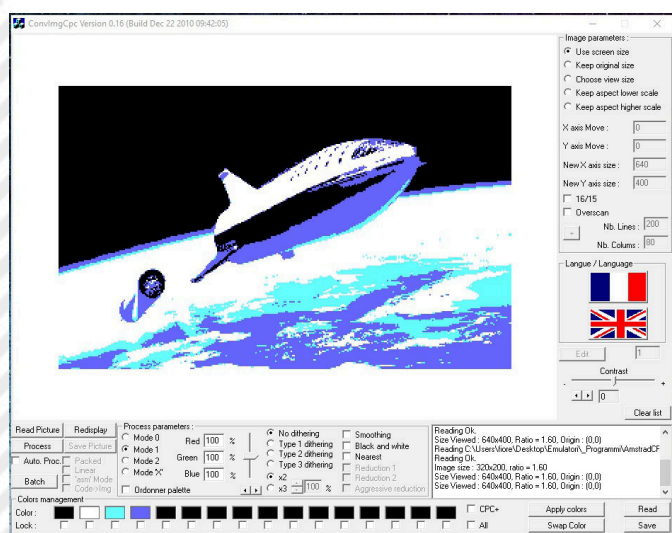
ConvImgCPC e' un programma freeware creato da **Demoniak** che permette la conversione di immagini da PC a CPC.

A differenza di tanti altri prodotti simili che avevo trovato, tutti a linea di comando, questo e' corredato da un'ottima interfaccia grafica, piuttosto intuitiva e con la possibilita' di scegliere un numero impressionante di opzioni per generare le nostre immagini SCR.

Dopo aver scelto lingua inglese ed aver caricato l'immagine in formato JPG, comincio a smanettare con le tante opzioni del programma.



Voglio generare un'immagine da visualizzare in **MODE 0**, quindi formato 320x200 ma con soltanto 4 colori a disposizione. Il software devo ammettere che fa veramente quello che promette, con pochi click sono in grado di trasformare una complessa immagine JPG in un file SCR pronto da dare in pasto al caro CPC.



Il tutto e' veramente semplice, basta selezionare il formato

MODE 1 nel menu' **Process Parameters** e poi cliccare il bottone **Process** per elaborare l'immagine. Il software fara' tutto il resto da solo, adattando l'immagine alla dimensione corretta e preoccupandosi di convertire i colori in un modo piuttosto fedele. Anche in questo caso ci sono decine di possibilita' di scegliere i colori, ma per quello che volevo fare io il processo automatico e' piu' che sufficiente visto e considerato che parliamo di soli 4 colori.

Per salvare poi l'immagine e' sufficiente premere il bottone **Save Picture** e una volta scelto il formato SCR, dare un nome al file.

E' possibile anche esportare anche la palette dei colori, tramite il bottone **Save** nel menu' **Colors Management**. Devo ammettere che devo ancora capire cosa fare con la palette... Probabilmente e' possibile utilizzarla in programmi grafici evoluti come GIMP, oppure, con un po' di smanettamento anche sul nostro Amstrad.

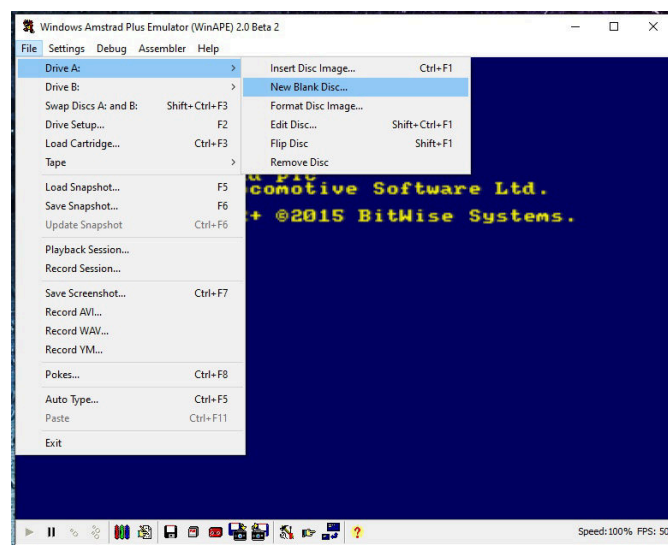
L'immagine SCR sull'Amstrad CPC

Bene, adesso abbiamo un'immagine in formato SCR, ma come facciamo a caricarla su un disco virtuale in formato .disk, del nostro Amstrad CPC?

Le successive istruzioni prenderanno in esame l'emulatore WinAPE, un ottimo emulatore di Amstrad CPC.

Una volta lanciato WinAPE procediamo a creare un'immagine di un disco con l'opzione -> **File -> Drive A: -> New Blank Disc** avendo cura di scegliere la directory dove creare l'immagine del disco e assegnando un nome valido.

Dopo la creazione, il disco deve essere formattato tramite il comando -> **File -> Drive A: -> Format Disc Image** rispondendo OK alla successiva domanda.



Adesso che il nostro disco e' formattato proviamo ad accedere al suo contenuto tramite il comando **CAT**.

Se tutto e' stato eseguito correttamente dovremmo ricevere come risposta dal nostro emulatore:

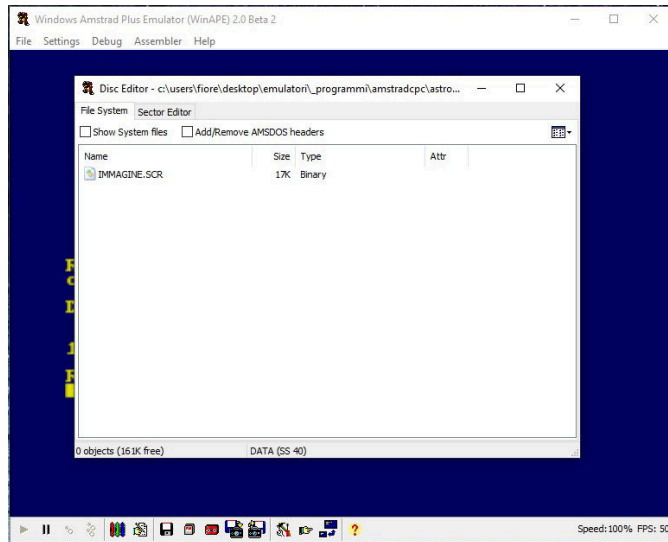




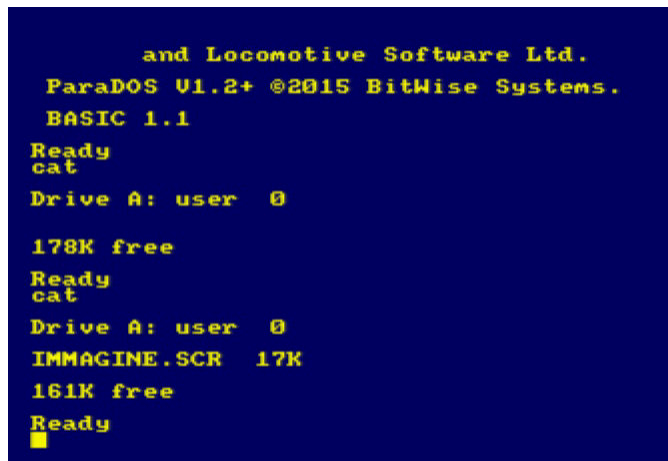
DRIVE A: user 0
178K Free

Lo vedete anche voi? Perfetto, passiamo a copiare la nostra immagine nel nuovo disco. Ancora una volta WinAPE ci facilita enormemente il compito perche' possiamo editare il contenuto del disco direttamente da Windows tramite i comodi comandi dell'emulatore.

Sempre dal menu' -> **File** -> **Drive A:** -> **Edit Disc**, bastera' trascinare nel disco il file che vogliamo copiare, in questo caso la nostra immagine ed il gioco e' fatto!



Eseguiamo di nuovo il comando CAT. Questa volta dovremmo trovare all'interno del disco il nostro file immagine.



Visualizziamo l'immagine SCR

Adesso che tutto e' pronto, non ci resta che visualizzare l'immagine che abbiamo preparato sul nostro Amstrad CPC. Scriviamo questo programmino in Locomotive Basic:

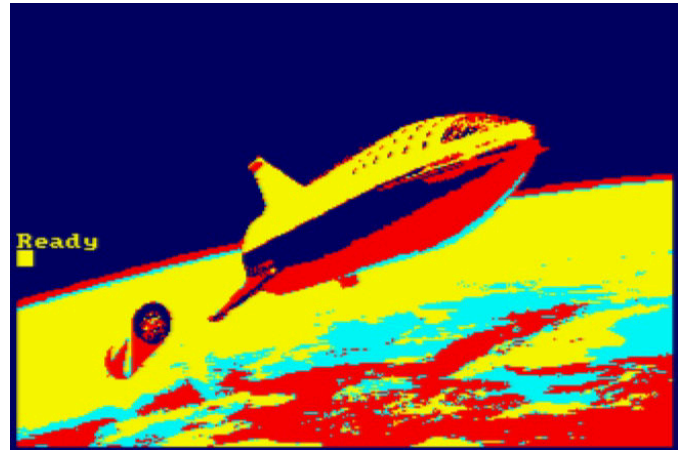
```
10 LOAD "IMMAGINE.SCR", &C000
```

e diamo il RUN...

Ehi, ma che succede? Cosa sono questi colori?

L'immagine che avevamo generato con ConvIMGPC era

leggermente diversa. Come mai adesso vediamo questi 4 colori al posto di quelli visualizzati dal convertitore?



Per capire cosa e' successo dobbiamo dare un'occhiata al formato SCR da vicino...

Il file SCR non e' altro che una mappa dei pixel in memoria con l'indicazione di quale dei quattro colori del MODE 1 utilizzare.

Per capire come questi pixel sono valorizzati dobbiamo studiare la 'scheda grafica' dell'Amstrad CPC, il **Motorola 6845 Cathode Ray Tube Controller**, per gli amici **CRTC**.

Il CRTC

I modi grafici standard supportati dal CRTC sono:

MODE 0 - 160x200 - 16 colori

MODE 1 - 320x200 - 4 colori

MODE 2 - 640x200 - 2 colori

Se controlliamo la rappresentazione dei pixel in queste 3 modalita' noteremo che:

- in MODE 0 ogni byte contiene 2 pixel (4-bit) 16 colori
- in MODE 1 ogni byte contiene 4 pixel (2-bit) 4 colori
- in MODE 2 ogni byte contiene 8 pixel (1-bit) 1 colore

La particolarita' e' che la distribuzione dei pixel non e' lineare, ma segue la seguente struttura:

Byte/Pixel structure									
VM (Mode)	7	6	5	4	3	2	1	0	
%00 (0)	A0	B0	A2	B2	A1	B1	A3	B3	
%01 (1)	A0	B0	C0	D0	A1	B1	C1	D1	
%10 (2)	A	B	C	D	E	F	G	H	

Partendo dalla modalita' 2, la piu' semplice, ogni byte di memoria contiene 8 pixel di profondita' di colore di 1 bit. La rappresentazione dei pixel in questo caso e' semplice, 1 bit per ogni pixel partendo dal piu' significativo.

Nel caso della modalita' 1, quella che interessa a noi, ogni





byte contiene 4 pixel, con una profondita' di colore a 4 bit. Come si vede dalla tabella, la distribuzione in questo caso non e' lineare. Il colore del primo pixel e' composto dal bit 7 e dal bit 3.

Nella modalita' 0, ogni byte contiene solo 2 pixel ed il discorso e' ancora piu' complicato, in quanto il colore del primo pixel e' composto dal bit 7, seguito dal bit 3, dal bit 5 e dal bit 1...

Se andiamo a dare un'occhiata alla paletta standard dell'Amstrad CPC ci sara' immediatamenmte chiaro del perche' i colori visualizzati dall'emulatore siano diversi da quelli del convertitore...

0 – Dark blue (1)
1 – Yellow (24)
2 – Cyan (20)
3 – Red (6)
4 – White (26)
5 – Black (0)
6 – Blue (2)
7 – Magenta (8)
8 – Dark cyan (10)
9 – Dark yellow (12)
10 – Light blue (14)
11 – Pink (16)
12 – Green (18)
13 – Light green (22)
14 – Dark blue (1)
15 – Cyan-blue (11)

I primi 4 colori sono esattamente quelli che vediamo nell'immagine postata poco prima.

Cambiamo i colori

Ma noi siamo ostinati e vogliamo a tutti i costi visualizzare l'immagine con i 4 colori che meglio la caratterizzano.

Provate a digitare queste 4 linee di codice:

```
10 INK 0,0
20 INK 1,26
30 INK 3,14
40 LOAD "IMMAGINE.SCR", &C000
```

Salvate il programmino sul disco con il comando:

```
SAVE "LOADER.BAS"
```

e riavviate l'emulatore con un bel RESET.

Adesso lanciate questo comando:

```
RUN "LOADER.BAS"
```



Molto meglio, non trovate?

Ma c'e' ancora qualcosa che non ci convince del tutto. Quel Ready e' veramente antiestetico e poi la nostra intenzione era quella di creare uno splash screen per il gioco Farmer. Quindi usiamo ancora una volta il nostro Loader, questa volta realmente per lanciare il gioco FARMER.BAS.

Copiate il gioco che trovate nel repository di Antonino Porcino a questo indirizzo:

aggiungete (se volete) i caratteri del precedente articolo, salvate il tutto come FARMER.BAS e poi scrivete questo semplice programmino:

```
10 INK 0,0
20 INK 1,26
30 INK 3,14
40 LOAD "IMMAGINE.SCR", &C000
50 FOR I=1 TO 5000
60 NEXT I
70 RUN "FARMER.BAS"
```

Et voila! Il nostro splash screen introduttivo del gioco Farmer e' pronto!

Mi auguro che questa semplice guida possa essere utile a qualcuno di voi per realizzare giochi per Amstrad CPC con un minimo di grafica.

Come sempre, se avete domande, non esitate a contattarmi.





Abbreviazioni per appunti sull'uso di una Interfaccia Grafica

di Alberto Apostolo

Durante lo studio dell'Analisi Dati e della Business Intelligence con l'uso di un noto spreadsheet, ho avuto spesso l'esigenza di annotare in modo abbreviato alcune sequenze di comandi dati con mouse e tastiera. In Rete ho cercato trucchi per sostituire una frase come "Cliccare con il tasto sinistro del mouse e trascinare il quadratino di riempimento automatico dalla cella A2 alla cella A10" con una abbreviazione del tipo

`CnD([Fill handle].[A2:A10]).`

Non trovando qualcosa di soddisfacente, ho deciso di creare "motu proprio" un sistema di abbreviazioni che fosse di aiuto anche nell'uso di altri software con G.U.I. (Graphical User Interface).

NOTAZIONI L'USO DEL MOUSE E PER I TASTI PREMUTI DA TASTIERA

Le azioni effettuate con più frequenza usando il mouse sono le seguenti:

- 1) Click (pressione del tasto sinistro)
- 2) Right-Click (pressione del tasto destro)
- 3) Double-Click (doppio click con il tasto sinistro)
- 4) Drag&Drop (selezione, trascinamento e rilascio)
- 5) Click&Drag (selezione e trascinamento)
- 6) Mouse Hover (spostare il mouse sopra elementi grafici).

N.B.: alcuni autori considerano sinonimi le azioni 4) e 5). In realtà esiste una distinzione [MA20]. L'azione Drag&Drop richiede due elementi grafici: il primo è l'elemento da trascinare, il secondo è il "bersaglio" sul quale effettuare il rilascio. L'azione Click&Drag richiede un elemento grafico e un offset costituito da uno o più parametri. Alcuni esempi di Click&Drag sono: modificare l'ampiezza di una finestra, il riempimento automatico di celle, regolare l'audio dell'altoparlante del computer muovendo uno slider come quello riportato in Figura 1.

Le azioni sopra menzionate si possono abbreviare con acronimi (alcuni di essi trovati in Rete):

- 1) LMB (left mouse button)
- 2) RMB (right mouse button)
- 3) DCLK (double click)
- 4) D&D (drag and drop)
- 5) C&D (click and drag)
- 6) MH (mouse hover).

Per indicare la pressione di uno o più tasti esistono diverse notazioni. Una di queste è la nuova notazione



Figura 1

Microsoft (es. Ctrl+X) che si trova citata su Wikipedia ma ognuno può adottare liberamente le notazioni più congeniali.

I COMPONENTI GRAFICI O WIDGET

In Informatica si definisce **widget** (window gadget) un qualsiasi componente grafico di una Interfaccia Grafica Utente (fonte Wikipedia).

Alcune tipologie di widget sono elencate di seguito (la spiegazione si trova anch'essa su Wikipedia):

- 1) Contenitori: window, modal window, ecc.
- 2) Navigazione: toolbar, link, tab, ecc.
- 3) Comandi: menu, ribbon, dock, icon, button, ecc.
- 4) Dialogo: dialog box, file dialog box, alert box, about box, ecc.
- 5) Input: check box, text box, combo box, list box, drop down list, slider, spinner, ecc.
- 6) Output: status bar, label, tooltip, progress bar, grid o data grid, ecc..

Le tipologie sopra elencate si possono arricchire aggiungendo celle, matrici, vettori, grafici, sheet, ecc..

"**Everything is a Widget**" è la filosofia adottata da alcuni software open-source che creano interfacce native per sistemi iOS e Android.

N.B.: molti dei termini elencati sono in Inglese. E' vero che si potrebbe parlare di "finestra di dialogo", "riquadro", "casella" e addirittura tradurre la parola "default" con "impostazione predefinita" ma è anche vero che l'Inglese è una delle lingue più parlate nel mondo e molti termini informatici sono più conosciuti in quella lingua.

INIZIARE A COSTRUIRE LA NOTAZIONE

Un **widget** ha come identificativo una stringa alfanumerica **W**. Se si desidera indicare anche il suo **tipo T** allora si adotta la notazione **T.W**. Siccome l'attenzione è posta esclusivamente sulla navigazione attraverso i vari componenti di una interfaccia grafica, sono trascurate le altre caratteristiche come colori, forma, dimensioni, posizione sullo schermo, ecc.

N.B.: Va bene anche la notazione **W.T** simile alla rappresentazione "**FileName.Ext**". **Le stringhe (salvo diversa indicazione) non sono case-sensitive.**

Come in una specie di **screen scraping**, per esprimere l'**azione A** su un gruppo di widget W_1, \dots, W_n si usa una notazione basata sulle funzioni (anche A è una stringa alfanumerica)

$A(W_1, \dots, W_n)$.

Nel caso venga prodotto un widget W in output si scriverà

$W=A(W_1, \dots, W_n)$.





Se le stringhe (usate per identificare widget, tipi, azioni) contengono caratteri diversi da lettere e numeri allora si racchiudono tra parentesi quadre come in SQL. Un altro "prestito" da SQL sono i commenti preceduti da doppio "trattino".

Per indicare un **widget Wk contenuto nel widget W** si scrive **W\Wk**, usando il carattere "\" come nello specificare il path di un file.

L'asterisco * si usa per evidenziare un widget sul quale è obbligatoria una azione (es. un campo da editare in un questionario). Se il contesto lo richiede si può usare anche per le azioni. Tuttavia si consiglia di aggiungere l'asterisco solo dove veramente necessario per non appesantire la scrittura.

Esempi:

- 1) Click(icon.Save)
- 2) DnD(W1,[Recycle Bin]) -- Drag&Drop verso il "bersaglio"
- 3) Edit([E-mail Address]*) -- azione su un campo obbligatorio
- 4) Edit(dialogbox.Price\textbox.Amount)
- 5) [Ctrl+Shift+Enter], [Ctrl+Click]
- 6) Press*(anykey) -- azione obbligatoria

N.B.: Le combinazioni di tasti premuti sono un caso particolare di azioni e possono esistere anche senza parametri.

OPERATORI PER CONCATENARE LE AZIONI

Non ho studiato il paradigma della programmazione funzionale. La scelta di una notazione basata sulle funzioni è dovuta ai miei studi di Matematica e alla possibilità di usare in modo intuitivo la composizione di funzioni per descrivere azioni più strutturate.

Inoltre sono messi a disposizione alcuni "operatori" per scrivere espressioni "algebriche" con parentesi tonde: "|" (Alt+124) indica azioni da eseguire nella sequenza in cui sono state scritte, "/" (shift+7) alternativa esclusiva, "+" azioni che si possono eseguire in qualunque sequenza.

L'uso del carattere "|" per concatenare operazioni in sequenza è simile al sistema operativo UNIX ed è usato con lo stesso significato in alcuni testi per la certificazione ECDL (European Computer Driving Licence).

Esempio: creazione di un vettore in uno spreadsheet specificando un cell-range.

Select(cell.[A2:A5]) | [Ctrl+Shit+Enter]

COS'È UN CELL-RANGE

In uno spreadsheet, un **cell-range regolare** è una matrice o un vettore definito a partire dalla cella più in alto a sinistra (valore minimo) fino alla cella più in basso a destra (valore massimo) [PX20].

Esempi: C7 (una singola cella), E2:E7 (vettore verticale), A3:F3 (vettore orizzontale), A1:C6 (matrice di 6 righe per 3 colonne).

Un **cell-range irregolare** è l'unione di gruppi di cell-range regolari separati con il carattere ";" (per es. A1:C6;C9;E2:E7).

LO SCREEN SCRAPING

Si realizza quando una applicazione interagisce con un'altra inviando comandi (tasti premuti, ecc.) e "leggendo" le risposte come se fosse un essere umano [HT18].

CREARE UN DIZIONARIO DELLE AZIONI E DEI TIPI

Data l'ampia libertà nello scegliere nomi di azioni e tipi di widget, se si vogliono condividere gli appunti con altre persone, è opportuno creare un piccolo "dizionario" con brevi note esplicative.

Le liste proposte sono studiate per non ricorrere alle parentesi quadre per delimitare le stringhe (chi vuole può fare aggiunte).

Lista delle azioni

- 1) Click oppure LC (Left Click)
- 2) RC (Right Click)
- 3) CC (Click Click = Double Click)
- 4) DnD (Drag and Drop)
- 5) CnD (Click and Drag)
- 6) CnP (Copy and Paste)
- 7) XnP (Cut and Paste)
- 8) Select (selezione oggetti)
- 9) Edit (edit celle,textbox, ecc.)
- 10) Copy
- 11) Cut
- 12) Paste
- 13) Close (di una finestra)
- 14) Press (per i tasti della tastiera)
- 15) MH (mouse hover)

Azioni per spreadsheet

- 16) Delete (elimina celle con scorrimento di quelle vicine)
- 17) Clear (cancella solo il contenuto delle celle)

Lista dei tipi

window, modalwindow, toolbar, link, tab, menu, ribbon, dock, icon, button, dialogbox, filedialogbox, alertbox, aboutbox, checkbox, textbox, combobox, listbox, dropdownlist, slider, spinner, statusbar, label, tooltip, progressbar, grid, datagrid, matrice, vettore, cella, fillhandle, grafico, sheet, key.

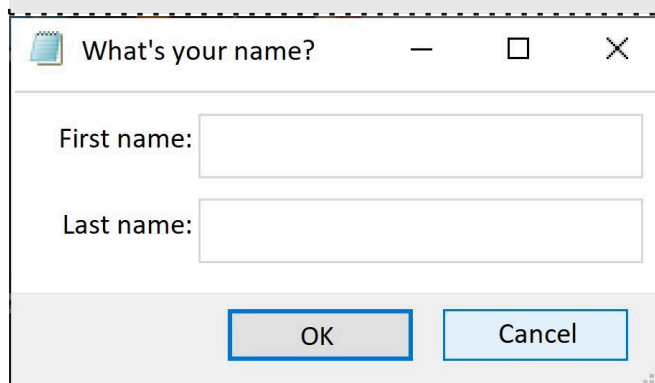


Figura 2





Se A, B, C sono azioni valgono le proprietà:

- 1) | , / , + sono associativi ossia
 $(A \times B) \times C = A \times (B \times C)$,
 - 2) / e + sono commutativi ossia : $A \times B = B \times A$,
 - 3) | non è commutativo ossia $A | B$ diverso da $B | A$.
- L'operatore | è distributivo solo a sinistra rispetto a / :
 $A | (B / C) = (A|B) / (A|C)$.
- L'operatore + è distributivo rispetto a / :
 $A+(B/C)=(A+B)/(A+C)=(B+A)/(C+A)=(B/C)+A$.

Esempio (descrivere l'editing dell'oggetto in Figura 2):

```
Edit([What's your name?]) =
( Edit([First name]) + Edit([Last name]) ) |
( Click(OK) / Click(Cancel) )
```

N.B.: Qualcuno potrebbe obiettare che nella realtà si potrebbe "cliccare" su "OK" prima ancora di avere editato i due campi richiesti. L'obiezione è corretta ma il traguardo è descrivere almeno una sequenza di azioni sensate e non analizzare tutti i possibili eventi nell'uso di una interfaccia.

RAGGRUPPAMENTI

In alcuni casi particolari si possono utilizzare alcune convenzioni per ottenere maggiore sintesi.

Se i widget W_1, \dots, W_n sono soggetti alla stessa azione A e allo stesso operatore x, è consentito scrivere:

$$A(W_1) \times \dots \times A(W_n) = A (W_1 \times \dots \times W_n)$$

Se i widget W_1, \dots, W_n hanno lo stesso tipo T e sono soggetti alla stessa azione A e allo stesso operatore x, è consentito scrivere:

$$A(T.W_1) \times \dots \times A(T.W_n) = A (T.(W_1 \times \dots \times W_n))$$

Esempio (di nuovo l'editing dell'oggetto in Figura 2):

```
Edit([What's your name?]) =
Edit([First name] + [Last name]) |
Click(OK / Cancel)
```

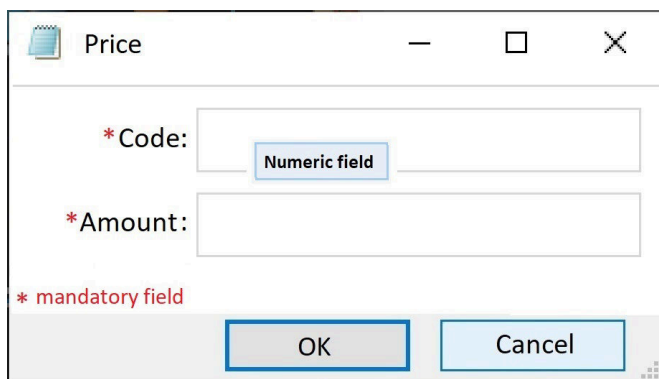


Figura 3

DOCUMENTARE LA STRUTTURA DI UN WIDGET

Qualche volta è utile fare una documentazione rudimentale della struttura di un widget.

La notazione $W(W_1, \dots, W_n)$ significa che W dipende dai widget W_1, \dots, W_n mentre la notazione $W\{W_1, \dots, W_n\}$ significa che W è composto dai widget W_1, \dots, W_n .

Esempio: nel dialog box Price sono presenti due box obbligatori. Un tooltip (costituito da una etichetta) avvisa che Code è numerico (Fig. 3).

```
Price{ textbox.Code*,
       textbox.Amount*,
       tooltip.Msg(Code){label.[Numeric field]},
       button.OK,
       button.Cancel
}
```

Se non è troppo complicato si possono raggruppare i widget per tipo come segue:

```
Price{ textbox.(Code*,Amount*),
       tooltip.Msg(Code){label.[Numeric field]},
       button.(OK , Cancel)
}
```

N.B.: In [HT18] al Capitolo 2, ho scoperto che questa rappresentazione ha una certa somiglianza con i "Data Language" e i "Resource File".

WIDGET EDITABILI

Nel caso di widget editabili (come le celle di uno spreadsheet) è comodo poter assegnare (tra parentesi quadre) un valore di default o una espressione.

Esempi:

- 1) Edit([First Name]=[Mario] + [Last name]=[Rossi])
- 2) Edit(cell.A7=[ROUND(A1+A4;2)])

N.B.: Nell'esempio 1, "+" è un operatore e vale come se fosse un separatore. Nell'esempio 2, "+" appartiene a una espressione all'interno delle parentesi quadre.

CONCLUSIONI

Sono state riutilizzate simbologie e User Experience esistenti senza pretendere di inventare l'ennesimo linguaggio informatico (secondo me, ne esistono già troppi).

L'obiettivo è avere subito a disposizione appunti in formato testo per descrivere in modo abbastanza ordinato la navigazione attraverso i vari menu e finestre di una interfaccia grafica.

Per chi desidera contribuire con miglioramenti, vale l'invito a presentarli sulle pagine di RetroMagazine.

Bibliografia

- [CZ05] M.Coggi, C.Zanon, "ECDL. Eserciziario e simulatore completi.", Edizioni FAG Srl, 2005.
- [Gas08] C.Gasparini, "ECDL CAD. Il manuale", Apogeo Editore, 2008.
- [HT18] A.Hunt, D.Thomas, "The Pragmatic Programmer" (Edizione Italiana stampata da Apogeo, 2018).
- [MA20] AA.VV., "Drag and drop steps", <https://help.mabl.com/docs/drag-and-drop-steps> , consultato il 27/4/2020.
- [MH05] J.Moran, V.Hull, "Brilliant ECDL", Pearson Education, 2005.
- [PX20] AA.VV., "What is a Cell Range", <https://www.perfectxl.com/about-spreadsheets/Excel-glossary/what-is-cell-range/> consultato il 28/4/2020.





Speciale Scacchi: 1° Torneo RMW per computer a 8-bit

di David La Monaca e Mathias Lorenz

Presentazione

In un precedente numero di RetroMagazine abbiamo recensito e fatto giocare uno contro l'altro due programmi scacchistici a 8-bit (Cyrus Chess I su ZX Spectrum 48K e Colossus Chess 3.0 su Atari 800XL) e con l'occasione abbiamo fatto il punto sulla forza di gioco dei motori scacchistici più forti degli ultimi anni. Per completezza abbiamo quindi presentato AlphaZero, il software di intelligenza artificiale firmato DeepMind/Google, che in modo abbastanza sorprendente è apparso sul mercato nel 2017 e ha dimostrato di poter sbaragliare senza appello tutti i tradizionali chess engine come Stockfish, Komodo, Deep Shredder, Houdini, Fritz, ecc. In che modo? Battendo nel dicembre 2017 in un match di 100 partite il software Stockfish 8 (uno dei più forti, con circa 3400 punti ELO al suo attivo) con 28 vittorie di AlphaZero, di cui 25 ottenute con il Bianco, e 72 partite patte.

Il secondo episodio di questo confronto, che può essere tranquillamente definito "generazionale", ha avuto luogo qualche mese dopo ed i risultati sono stati anche questa volta sorprendenti e hanno segnato un vero e proprio punto di non ritorno, aprendo definitivamente le porte ad una nuova epoca negli scacchi al computer, ma anche nell'informatica applicata all'AI. Per la cronaca una versione aggiornata di AlphaZero ha letteralmente distrutto Stockfish 8 in un match condotto su 1000 partite con lo straordinario punteggio di +155 -6 =839, ossia 155 vittorie, 6 sconfitte e 839 patte. E non è tutto, perché AlphaZero ha anche stravinto contro il motore tradizionale in altre partite organizzate con tempi di riflessione diversi fra i due motori (con differenze anche fino a 10 a 1). In altre partite, il nuovo AlphaZero ha battuto sonoramente l'ultima versione di sviluppo di Stockfish, precisamente la 9, con risultati praticamente identici a quelli di Stockfish 8, secondo quanto affermato da DeepMind. Il motore ad apprendimento automatico ha vinto tutte le partite anche contro una variante di Stockfish che utilizza un libro di aperture molto esteso. L'aggiunta del libro di aperture ha comunque portato qualche aiuto a Stockfish, che alla fine ha vinto un numero considerevole di partite quando AlphaZero era con il Nero, ma non abbastanza per vincere il match.

Nel numero 5 di RM (aprile 2018) un articolo/recensione su un paio di programmi scacchistici a 8-bit, per dare forma ad un mio vecchio pallino, si era trasformato in un match di 6 partite. La sfida fra Cyrus Chess (Spectrum 48K) e Colossus 3.0 (Atari 800XL) terminò 3.5 a 2.5 a favore di Colossus, che dimostrò di poter contare su di un algoritmo di maggiore forza di gioco ed un sostanziale controllo durante il mediogioco e certamente una tendenza a commettere meno errori rispetto all'avversario. Alcune

riviste degli anni 80 e 90 hanno tentato di stabilire quale programma scacchistico a 8-bit fosse il più forte. La serie Colossus, culminata nella versione 4.0, sembra essere il candidato più accreditato per questo primato, ma prove certe nessuno ce le ha mai fornite.

A distanza di tempo, l'articolo è stato ripubblicato in traduzione sul numero #00 di RetroMagazine World ed è stato notato da Mathias Lorenz, un appassionato di retrocomputer e fervente scacchista. Dopo un rapido scambio di email, con Mathias abbiamo deciso di provare a definire il programma scacchistico a 8-bit più efficiente di sempre. Ma, ci siamo chiesti, in che modo e con quali regole avremmo potuto arrivare ad un tale traguardo, ammesso che fosse possibile giungere ad un esito assoluto? Dopo alcune ulteriori considerazioni, abbiamo deciso di organizzare un vero e proprio torneo di scacchi fra microcomputer. I software a 8-bit sfidanti sono stati selezionati accuratamente fra tutti quelli pubblicati negli anni 80 e al termine 4 programmi fra i più venduti sono stati ammessi al torneo. Ad ogni programma è stato abbinato un microcomputer, anche questi fra i più diffusi dell'epoca. Così, ecco la lista completa dei partecipanti:

Amstrad CPC 464 / Colossus 4.0

(1986) ELO 1676 – 30 secs/move

ZX Spectrum 48K / Cyrus Chess II

(1986) ELO 1525 – Level 4

Commodore 64 / Chessmaster 2000

(1986) ELO 1621 – Level 3

Atari 800XL / Sargon III

(1985) ELO 1569 – Level 3

Il torneo si è svolto sulla base di un calendario di incontri con andata e ritorno, detto "all'italiana", in modo che ciascun software/sistema avesse lo stesso numero di partite con i pezzi Bianchi. Tutte le partite del torneo sono state raccolte e sono disponibili su richiesta in formato PGN. Nella classifica finale il vincitore risulta essere Cyrus II/ ZX Spectrum con 3.5 punti, a pari merito con Sargon III/ Atari 800XL che ha totalizzato lo stesso punteggio sulla scacchiera, ma Cyrus II ha prevalso per "spareggio tecnico" dato il numero maggiore di vittorie (ben tre) ottenute con i pezzi Neri. Tutti gli altri criteri per imporre un tie-break e quindi un vincitore erano risultati anch'essi in totale equilibrio.

Il premio virtuale come migliore partita del torneo va a quella disputata fra ChessMaster 2000 e Colossus 4.0 nel quarto turno, primo incontro, terminata con la vittoria dei Bianchi.

Lascio ora la parola a Mathias che vi descriverà come si è svolto il torneo e, soprattutto, come si è giunti al risultato finale, che per certi versi ha sorpreso entrambi.





Program	Sargon 3 Atari 800XL	Cyrus 2 ZX Spectrum	Chessmaster 2000 C64	Colossus 4.0 Amstrad CPC464	White Black	Overall	Rank
Sargon 3 Atari 800XL	X	0-1	1-0	1-0	2	3.5	1
Cyrus 2 ZX Spectrum	0-1	X	0-1	1/2-1/2	0.5	3.5	1
Chessmaster 2000 C64	1/2-1/2	0-1	X	1-0	1.5	2.5	2
Colossus 4.0 Amstrad CPC464	1-0	0-1	1-0	X	2	2.5	3
	1-0	1/2-1/2	1-0		0.5		

Figura 1 - Il tabellone del torneo con la classifica finale

Il Torneo

Per questo torneo di microcomputer avremmo potuto prendere in considerazione ogni sistema a 8-bit ed ogni programma scacchistico. E inoltre la scelta dei software scacchistici è in realtà ancora più ampia di quella degli home computer diffusi all'epoca. I sistemi come Commodore 64, Atari 800XL e Amstrad CPC464 potevano essere equipaggiati con un drive per dischetti, mentre lo ZX Spectrum utilizzava un registratore a nastro. A parte ciò, questi computer fornivano un framework sufficiente per far girare i primi programmi commerciali di scacchi multiplatforma. Il mercato offriva, per i giocatori di scacchi amatoriali, anche un'intera gamma di computer scacchistici da tavolo, che erano dotati di hardware progettato appositamente all'interno di scacchiere in legno e pezzi dalla forma piacevole. Fino a quel momento il computer doveva adattarsi alle abitudini di gioco con pezzi reali e scacchiere in legno. Al giorno d'oggi accade il contrario e gli scacchisti si sono adattati alle modalità di gioco su schermo, giocando online e muovendo i pezzi con un mouse. Da questo punto di vista i computer a 8-bit sono stati i pionieri nel portare a tutti questa nuova modalità di gioco che oggi consideriamo normale. Ancora negli anni 80 i programmi cercavano di fornire al giocatore umano la stessa esperienza di gioco a cui era da sempre abituato, come se il giocatore stesso fosse seduto davanti alla scacchiera. La grafica 3D in prospettiva elaborata dagli home computer erano spesso anche un importante argomento di marketing per promuovere l'acquisto di questo o quel programma. La rappresentazione 2D risultava meno impressionante era più familiare per via delle illustrazioni che apparivano sui libri di scacchi ma non era ancora giunta nell'immaginario collettivo.

Insieme agli home computer già citati, i programmi di scacchi partecipanti al nostro torneo dovrebbero essere tra i più forti secondo quanto pubblicato dalle riviste di videogame e secondo quanto visto nei tornei fra computer e quelli fra computer ed esseri umani. Inoltre, la data di pubblicazione dei programmi dovrebbe essere avvenuta nello stesso periodo, dal momento che uno sviluppo continuativo avrebbe certamente portato a programmi più forti e questo avrebbe portato ad una distorsione dell'esito finale. In base a questo, i programmi scelti Colossus 4.0, ChessMaster 2000 e Cyrus II sono stati

effettivamente pubblicati nello stesso anno 1986 e soltanto Sargon 3 è di un anno più tardi dei suoi avversari. La disposizione dei programmi sulle macchine ospiti è stata decisa in modo più pragmatico: a sorte. In alternativa, avevamo pensato ad un solo programma scacchistico che gira sui vari sistemi tutte le partite contro se stesso. Ciò ci avrebbe forse fornito un risultato su quale fosse la migliore combinazione hardware/software. Ma la nostra attenzione si è concentrata su come determinare una certa superiorità da parte di uno dei programmi testati senza tener conto della piattaforma utilizzata. Subito dopo aver effettivamente trovato una tale superiorità, una seconda serie di partite avrebbe potuto rilevare l'ambiente hardware ottimale.

I programmi prescelti sono tutti dotati di una libreria di aperture comparabile fra loro (circa 50-100 mila mosse). Pertanto, le aperture sono state gestite correttamente durante le partite, per quanto sia possibile dire in base allo stato attuale della teoria. Gli errori posizionali durante le prime mosse e persino le sviste clamorose sono mancate, anche se nessun programma avrebbe cercato di provocare errori di questo genere mediante trappole mirate in apertura. Il tempo limite (breve) è stato scelto per ottenere risultati rapidi delle partite. L'impostazione di base è stata di 30 secondi per mossa o un livello di difficoltà corrispondente allo stesso tempo per mossa. Questa regolazione ha condotto a partite dalla durata da 1 a 2 ore.

Dopo 12 partite complessive (tra andata e ritorno), la forza di gioco dimostrata dai vari programmi/computer è sostanzialmente simile e non è stato possibile determinare una vera e propria superiorità da parte di uno dei programmi, anche se questo primato spesso era stata un'arma di propaganda dei prodotti durante il loro periodo di vendita. Per poter stabilire un chiaro vincitore forse avremmo dovuto giocare molti altri match. Riguardo a questo, in pratica abbiamo ottenuto soltanto un'istantanea della reale forza di gioco, ma lo svolgimento delle singole partite ha fornito anche delle indicazioni per poter osservare i diversi stili di gioco in maniera comparativa. Le partite sono state spesso simili a quelle fra avversari umani, in cui entrambe le parti a volte fanno mosse dubbie che comportano grandi fluttuazione nel vantaggio attribuito nel corso degli incontri.





A mosse mediocri spesso veniva risposto con mosse altrettanto mediocri, segno che i motori hanno ignorato l'intrinseca debolezza di alcune decisioni. Solo quando ci si è trovati in una posizione realmente cattiva, allora il contro-attacco si è dimostrato decisivo. Anche se non sono stati commessi errori di rilievo, lo stile di gioco sostanzialmente solido dei motori è stato parzialmente annullato sotto la pressione del tempo. Cyrus II ha costituito un'eccezione. Come vincitore a pari merito del torneo, la sua maniera rapida di giocare nel mediogioco è stato particolarmente impressionante. I 30 secondi per mossa sono stati spesso ignorati dal programma con mosse migliori degli avversari che hanno usato tutto il tempo a disposizione. Si deduce che questo programma è particolarmente consigliato per giocatori occasionali e impazienti in cerca di un avversario forte.

Ma i due leader del torneo hanno mostrato anche in varie situazioni che i principi fondamentali e ben noti degli scacchi non sono stati presi in considerazione. Per esempio, l'isolamento di un pedone è favorevole all'avversario, perché senza essere coperto da un altro pedone potrebbe essere attaccato facilmente. Nella prima partita del sesto round (21-esima mossa), ChessMaster 2000 non ha gestito lo scambio di pezzi nel modo appropriato. Allo stesso modo è incomprensibile che Cyrus II rifiuti la patta per ripetizione di mosse alla 74-esima mossa della stessa partita solo per poi andare a vincere in una posizione completamente perduta, grazie ad un errore clamoroso dell'avversario. E anche i novellini conoscono modi più efficaci per dare scacco matto al Re con la propria Torre nel finale di partita, come è capitato a ChessMaster 2000. In questo modo il programma si avvicina allo scacco matto con algoritmi apparentemente severi e rigidi.

Ciò che è speciale nei titoli dei migliori programmi di scacchi è l'essere multiplatforma. I sistemi più diffusi negli anni 80 erano molto simili in termini di larghezza del bus a 8-bit e per dimensione della RAM fisica. Come già detto, anche il supporto di memorizzazione dei dati ha giocato un ruolo importante per l'acquirente dei giochi. La limitazione della capacità di memorizzazione mediante l'uso di una cassetta magnetica ha avuto anche la triste conseguenza



Figura 2 - Cyrus II - CM 2000 mossa 21

di incorporare un motore diverso e più debole, come nel caso della versione su nastro di ChessMaster 2100. Allo stesso modo la frequenza di clock di un processore Z80 a 3.5 MHz dello ZX Spectrum o i 4 MHz dell'Amstrad CPC sono chiaramente superiori a quella di 1 MHz di un 6510 del C64. Almeno dal punto di vista teorico, questa differenza doveva significare una forza di gioco più potente su questi dispositivi. E infatti, i processori dei computer da tavolo dello stesso periodo hanno frequenze di clock intorno ai 5 MHz, il che ci porta ad una forza di gioco paragonabile a quella del programma più forte del torneo, Colossus 4.0 su Amstrad CPC in termini di punteggio ELO (2).

Tuttavia, è importante considerare in questo contesto che la profondità di ricerca di una valutazione effettuata da un programma a proposito di una determinata posizione è di solito fornita in passi di mezza mossa - un cosiddetto "ply" o metà mossa. Questa profondità indica in modo chiaro quante mosse il programma è in grado di calcolare in anticipo. Per cercare la mossa ottimale, il programma deve simulare il maggior numero possibile di posizioni regolari. Un algoritmo di ricerca veloce è utile per calcolare queste posizioni potenziali. La velocità è specificata in nodi al secondo - NPS in breve. Dal punto di vista visivo, un programma con NPS elevato ha un albero di ricerca molto più ramificato. Calcolando più velocemente, il processore fornisce una base più ampia per ulteriori valutazioni. Un processo che ovviamente acquista importanza con un numero crescente di "ply".

Sorprendentemente, la pratica della programmazione dei moderni motori scacchistici ha reso chiaro che l'influenza sulla reale forza di gioco è piuttosto marginale (3). Da un lato, c'è un programma che può esaminare molte posizioni con NPS elevati in modo piuttosto impreciso. Dall'altro lato, c'è un programma che analizza meno posizioni con NPS bassi in modo molto preciso. I programmi possono differire fino ad un fattore 10 nel numero di NPS e giocare comunque nello stesso campionato. Un'alta frequenza di clock ha un vantaggio per il numero NPS, perché l'algoritmo di ricerca ad alta intensità di calcolo può essere eseguito più velocemente. In retrospettiva, questo fatto da solo non è sufficiente a rendere il programma automaticamente migliore.

Al giorno d'oggi, questo piccolo torneo non sarebbe certamente preso molto seriamente dai programmatori dei motori scacchistici per quanto riguarda la determinazione di una concreta forza di gioco dei partecipanti. Questo perché ci sono da un bel po' di tempo strumenti specializzati per testare le particolari abilità di un motore. Essi aiutano a far competere i motori l'uno contro l'altro migliaia di volte e hanno fornito ottime previsioni fino a calcolare i punteggi ELO con 1 unità di scarto soltanto (4). Un altro metodo per testare la qualità di un motore è quello di specificare una certa posizione scacchistica e passarla al programma per determinare la mossa successiva. In questo modo si può determinare in modo abbastanza obiettivo se la mossa corretta è stata trovata e se è disponibile in un





Figura 3 - Cyrus II - CM 2000 mossa 74

periodo di tempo ragionevolmente breve.

Negli anni Ottanta, questo tipo di automazione era ancora piuttosto lontano. I migliori giocatori tendevano a sorridere nei confronti dei programmi di scacchi che giravano sugli home computer. Per questo motivo, i programmi a prezzi accessibili di allora erano dotati di manuali spessi e completi che spiegavano le regole degli scacchi al volenteroso principiante, ma anche il giocatore esperto poteva imparare con le partite famose fornite come esempi.

Inoltre, interi saggi storici sugli scacchi venivano in questo modo trasmessi al grande pubblico. Non appena l'interesse per il gioco era risvegliato, i programmi hanno comunque presentato al giocatore occasionale un serio avversario. Speriamo che leggere di questo torneo possa far rivivere ai lettori un po' di questo spirito pionieristico. Di seguito alcuni brevi commenti sulle singole partite alla maniera di un cronista sportivo per fornire un indizio sullo svolgimento delle partite.

1° Turno

Game 1: Colossus 4.0-Chessmaster 2000 | 1-0

Colossus non lascia il mediogioco indenne ma trova un modo per ritornare in partita e vince con una bella combinazione di matto.

Game 2: Sargon III-Cyrus II | 0-1

A Cyrus viene regalato un pedone in apertura per finire col conquistare un pezzo nel finale che lo porta alla vittoria.

2° Turno

Game 1: Chessmaster 2000-Sargon III | 1/2-1/2

Chessmaster trova un pareggio che equivale ad una vittoria. Sargon mostra una gestione del finale molto debole nonostante una posizione superiore ed un pedone in più.

Game 2: Cyrus II-Colossus 4.0 | 1/2-1/2

Colossus raggiunge una posizione migliore ma dissipa tutto il suo vantaggio con una manovra di Regina inutile. Alla fine si salva con un pareggio.

3° Turno

Game 1: Chessmaster 2000-Cyrus II | 0-1

Dopo un grave errore posizionale in apertura, Cyrus vince un pezzo e poi l'intera posta.

Game 2: Colossus 4.0-Sargon III | 1-0

L'apertura, giocata in modo non ottimale da Colossus, offre delle chance di vittoria all'avversario. Nel finale Sargon mostra tendenze suicide!

4° Turno

Game 1: Chessmaster 2000-Colossus 4.0 | 1-0

Una partita molto divertente con opportunità da entrambi i lati. Nel finale un pedone in più decide in favore di Chessmaster. La migliore del torneo.

Game 2: Cyrus II-Sargon III | 0-1

La sorpresa del torneo. Sargon vince la sua prima partita contro l'imbattuto Cyrus. Dal canto suo Cyrus butta via un certo vantaggio e viene superato dopo alcune imprecisioni.

5° Turno

Game 1: Sargon III-Chessmaster 2000 | 1-0

Dopo un attacco rischioso, Sargon impiega quasi 30 mosse per trovare il matto nel finale con due Alfieri ed una Torre contro una sola Torre.

Game 2: Colossus 4.0-Cyrus II | 0-1

Colossus riesce a guadagnare lentamente una qualità per poi lanciarsi inutilmente in un contrattacco che alla fine lo spazzerà via.

6° Turno

Game 1: Cyrus II-Chessmaster 2000 | 0-1

Una sola mossa prima della triplice ripetizione, Cyrus getta via la chance di una patta sicura e perde un'importante partita.

Game 2: Sargon III-Colossus 4.0 | 1-0

Un'emozionante lotta che si conclude solo nel finale di partita, inizialmente in perfetto equilibrio, dopo che Colossus commette un grave errore posizionale.

Riferimenti

- (1) http://www.spacious-mind.com/html/commodore_c64_fidelity_chessma.html
- (2) <https://www.schach-computer.info/wiki/index.php/Wiki-Elo-Liste>
- (3) <https://www.stmintz.com/ccc/index.php?id=198960>
- (4) <http://rebel13.nl/misc/nice.html>

(R1) Alphazero Vs Stockfish – 100 games match
<https://www.chess.com/news/view/google-s-alphazero-destroys-stockfish-in-100-game-match>

(R2) AlphaZero Vs Stockfish – 1000 games match
<https://www.chess.com/news/view/updated-alphazero-crushes-stockfish-in-new-1-000-game-match>





Introduzione ad ARExx – quarta parte

di Gianluca Girelli

GAME CODING CON AREXX - PARTE 1

Dopo aver analizzato nei numeri precedenti le caratteristiche basiche del linguaggio, vediamo ora come implementare una avventura testuale con ARExx. Per meglio capire gli argomenti che verranno trattati, consiglio di riprendere le parti precedenti pubblicate su RetroMagazine nei numeri 20, 22 e 23, nonché l'articolo "Cenni di Game Coding" pubblicato sul numero 17.

E' importante notare che l'implementazione proposta è solo una delle innumerevoli implementazioni possibili di un "game engine" di questo tipo. Pur non essendo necessariamente la più efficiente, è stata scelta per la sua facilità di comprensione anche da parte dell'utente meno esperto, senza preoccuparsi troppo di questioni relative all'occupazione di memoria sia per quanto riguarda il mondo di gioco che per quanto concerne la gestione delle variabili.

1. IMPLEMENTAZIONE DEL MONDO DI GIOCO

Nei libri di William Gibson, noto scrittore "cyberpunk", si è sempre fatto largo uso del termine "matrice" per definire la struttura del cyberspazio dentro il quale si svolgeva parte della vita dei protagonisti. Nella trilogia "The Matrix", chiaramente ispirata a tali romanzi, la matrice era il costruito dentro il quale i protagonisti credevano di vivere. In ossequio a tali illustri predecessori ecco quindi il nostro mondo di gioco, definito come una matrice bidimensionale:

```
/* prepara matrice. 1=esiste stanza */
matrice.=1
matrice.1.1=0
matrice.1.5=0
...
...
```

Come si è visto nei numeri precedenti, la definizione "matrice.=1" è da sola sufficiente a dichiarare un costruito multidimensionale ed ad inizializzarlo per indicare al codice successivo quali locazioni esistono veramente e quali no. Poichè non tutte le "stanze" sono locazioni accessibili, come si vede dalla mappa in figura 1, le istruzioni tipo "matrice.x.y=0" servono a "marcare" alcuni

elementi per altri scopi. Ad esempio, la prima locazione (indici 1,1), verrà utilizzata per immagazzinarvi all'interno la descrizione del "game over" (vedremo più avanti in che modo). La mappa descrive un mondo di gioco cyberpunk che si snoda su 7 righe ed 8 colonne. Dal punto di vista del codice, usare un tale costruito ne rende estremamente semplice la navigazione. Se per esempio assumiamo di essere nella locazione (x,y), andare verso "nord" vorrà semplicemente dire "spostati alla riga superiore", mentre andare verso "est" vorrà dire "avanza di una colonna". In altre parole:

```
when nome='NORD' then x=x-1
when nome='EST' then y=y+1
```

A questo punto è però necessario fare alcune assunzioni su come vogliamo che il mondo di gioco venga rappresentato all'utente e su come vogliamo che egli/ella interagisca con esso. Anche in questo caso, ho scelto di fare ricorso ad una matrice multidimensionale, composta da tante "colonne" quante sono le locazioni del gioco (contando anche quelle inaccessibili per l'utente poichè usate per altri scopi), e sette righe. Di seguito un esempio con relativa spiegazione.

```
Situazione.1.2='Panthers cove.'
Situazione.2.2='Le pantere sono un gruppo di crackers
che vivono ai margini della societa'.'
Situazione.3.2='Sono fortemente politicizzati e il loro
scopo e', come spesso accade,'
Situazione.4.2='mettere in crisi il sistema. Nonostante
non condividiate gli stessi ideali,'
Situazione.5.2='rispettano la tua etica. Non avrai aiuti
diretti se non sotto forma di software.'
Situazione.6.2='Al momento e'' disponibile un war DIALER.'
Situazione.7.2='NORD EST OVEST UP DOWN'
```

Come si potrà vedere confrontando la mappa, quella appena riportata è la descrizione della locazione Numero 2. Nel gioco, ogni stanza è descritta da un testo di 7 righe. Quando necessario, la penultima riga descrive una creatura o un oggetto. Uccisa la creatura, o rimosso l'oggetto, è necessario che tale riga venga "azzerata" (con un semplice Situazione.6.2='') per non dare informazioni false. L'ultima riga, invece, elenca le direzioni non consentite per la navigazione delle stanze. Definite poi le variabili "x" ed "y" (posizione corrente) e le variabili "maxrow" e





	1	2	3	4	5	6	7	8	
1	game over description	Panthers: no help ma software	Bassifondi	Body Bank: corpo, botta in testa. Targhetta dice Edison Carter	x	Residential 1: cadavere esploso, TV accesa	Residential 2: gente ipnotizzata, TV accesa	game won description	1
2	vicolo: giornale	strada	strada	strada	strada	strada	strada	train station to spaceport - voli sospesi	2
3	vicolo	Arcade	strada	First Orbital Bank	x	Bank of Zurich: Flatline.		senso/net	3
4	Chatsubo: 50c, pawn ticket		strada	dunking donuts, police jack- se ti connetti ti arrestano	tactical police, info su casi in residential 1 e 2	strada	strada	Net 23 - 4° piano - attacco finale	4
5	x	Shin Pawn Shop: cyberdeck UXB	downtown ovest	downtown est	strada	strada	Cray computing, 2° attacco- visione 3D del cyberspace	Net 23 - 3° piano -3° attacco	5
6	x	Cheap Hotel: hacking software lev1	strada	Amiga R&D, 1° attacco, blipvert project	Maas Biolabs, ripristino fisico per cyberspace	strada	Hosaka computing, nuove interfacce neurali	Net 23 - 2° piano	6
7	x	Musabari Grill	strada	strada	strada	piazza	strada	Net 23 - 1° piano	7

Figura 1

"maxcol" (limiti della matrice del mondo di gioco), la descrizione della locazione corrente si troverà nella nostra matrice delle locazioni all'indirizzo "Pos=(x-1)*maxcol+y". Di conseguenza, "Situazione.1.Pos" conterrà, nell'esempio sopra descritto, il "titolo" della nostra locazione (Panthers Cove). Abbiamo quindi risolto il primo problema (rappresentazione del mondo di gioco) e la sua implementazione in codice ARexx è la seguente:

```
/*-----*/
/* Stampa posizione e situazione */
/*-----*/
Scenario:
say
do i=1 to 6
  if Situazione.i.Pos~='' then say Situazione.i.Pos
end
return
```

La routine appena descritta non fa altro che visualizzare a video le 6 righe di testo che descrivono la locazione (a meno che una di esse non sia vuota, ad esempio perchè abbiamo già prelevato l'oggetto). La settima riga invece, che indica al codice quali sono le direzioni non consentite alla navigazione, verrà spiegata nel paragrafo seguente.

2. NAVIGAZIONE DEL MONDO DI GIOCO

Capito come possiamo creare un mondo di gioco e come renderlo fruibile al giocatore, vediamo ora come muoverci all'interno di esso. Nell'articolo precedente, pubblicato sul numero 23, abbiamo visto come implementare un semplice "parser" (analizzatore sintattico). Tale parser fu creato proprio per questo gioco, e si fonda sul fatto di lavorare con una semplice grammatica basata sulla storica forma "verbo+oggetto" tipica delle primissime avventure testuali. Inserita una frase del tipo "vai nord" il parser "spezza" quindi la frase nei suoi componenti e, dopo aver controllato che "vai" sia effettivamente una parola contenuta nel suo vocabolario, chiama la routine corrispondente.

Tale routine potrebbe avere la seguente forma:

```
/*-----*/
/*      VAI      */
/*-----*/
Vai:
if find(Situazione.7.Pos,nome)>0 then say'Non
puoi andare a ' || nome
else do
  select
```





```

when nome='NORD' then x=x-1
when nome='SUD' then x=x+1
when nome='EST' then y=y+1
when nome='OVEST' then y=y-1
when nome='UP' then x=x-1
when nome='DOWN' then x=x+1
    otherwise say "non capisco. riprova"; say;
return
end
Pos=(x-1)*maxcol+y
do i=1 to 6
    if Situazione.i.Pos~='' then say Situazione.i.Pos
end
end
return

```

A questo punto i lettori più smaliziati avranno già capito cosa fa il codice e come funzioni la famigerata riga 7:

- innanzitutto si controlla se la direzione scelta sia effettivamente consentita: ad esempio, se vogliamo andare verso est mentre siamo nell'ultima colonna più a destra, ciò non sarà possibile (say'Non puoi andare a '). L'istruzione "find" cerca una sottostringa all'interno di una stringa: se la sottostringa viene trovata nella riga 7 della locazione attuale significa che quella direzione è interdetta alla navigazione;
- se la direzione voluta è invece tra quelle consentite, verranno modificate le variabili "x" o "y" a seconda della direzione, di fatto spostando il giocatore nella nuova locazione;
- infine, dopo avere trasformato le coordinate matriciali in una posizione scalare, verrà visualizzata a video la descrizione della nuova posizione.

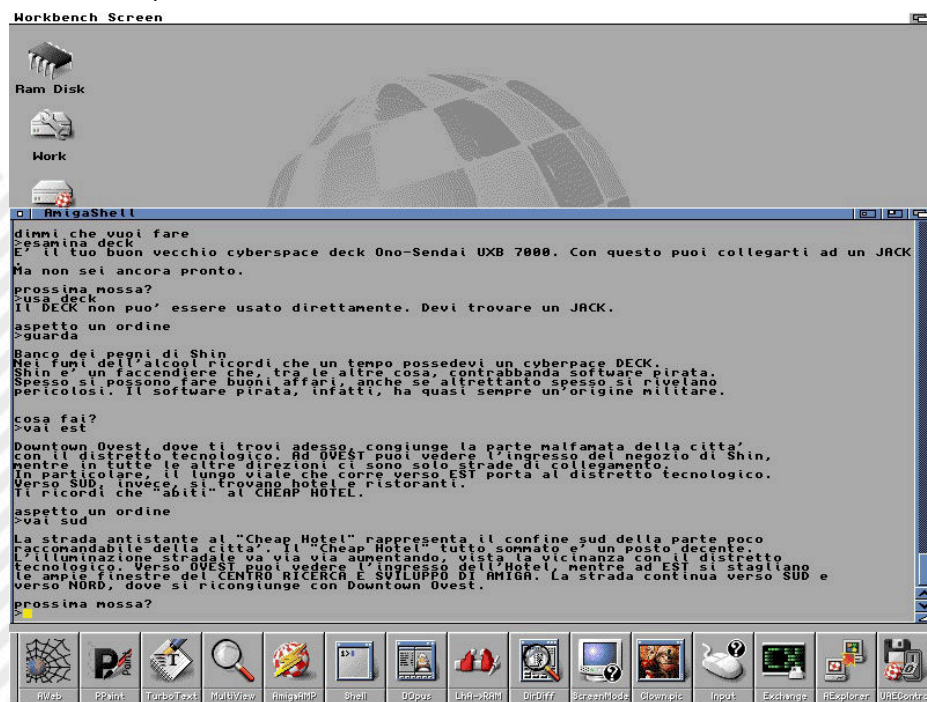


Figura 2

3. INTERAZIONE CON IL MONDO DI GIOCO

Come ultima parte di questo articolo, vediamo ora come è possibile far interagire l'utente con il mondo di gioco. La routine di "navigazione" è chiaramente una parte importante, ma senza qualcosa di più complesso non ci sarà mai un gioco, nemmeno se fosse relegato al rango di semplice "walking simulator". Quante e quali azioni il nostro parser dovrà essere in grado di capire dipende da come vogliamo portare avanti la nostra storia, ma come minimo dovremo essere in grado di elaborare le seguenti richieste: "vai", "prendi", "usa", "esamina" (per ottenere maggiori informazioni su un oggetto), "guarda" (per richiedere che venga rivisualizzata la descrizione della locazione), "lista" (o "inventario", per controllare il nostro equipaggiamento) ed ovviamente "vocabolario" per dare all'utente un'idea di ciò che può fare. Inoltre, il nostro codice potrebbe contenere altre istruzioni non documentate, di solito usate dai programmatori e dai tester durante lo sviluppo del software.

E' impossibile, infatti, pretendere che tali figure debbano ogni volta ricominciare il gioco dall'inizio solo per testare nuove funzionalità o per eliminare eventuali bug presentatisi in fase avanzata. Nel caso tali funzionalità vengano rilasciate al pubblico una volta completato lo sviluppo siamo di fronte ai cosiddetti "cheat codes". Nel caso del nostro gioco, "debug" richiama una routine usata per spostarsi velocemente da qualsiasi locazione a qualsiasi altra immediatamente e senza dover attraversare tutta la mappa, mentre "var" visualizza a schermo lo stato delle variabili di gioco per controllare che non ci siano assegnazioni errate.

Non tutte le azioni hanno bisogno di argomenti (es: "lista"

o "guarda") ed ecco quindi spiegato perchè il parser seguente è diviso in due sottoparti: se la "frase" inserita contiene solo una parola essa viene direttamente assegnata alla variabile "verbo" (sempre che sia una comando valido) e la variabile "nome" (il nostro oggetto) viene azzerata, altrimenti la frase viene scomposta nei suoi elementi.

E' importante notare che in un gioco come questo non sarebbe forse necessario "azzerare"

sistematicamente la variabile "nome". In fin dei conti, se dopo il comando "GUARDA" (che non la usa) venisse invocato il comando "VAI" (per il quale la variabile "nome" conterrà la direzione desiderata), non ci sarebbe





nessun conflitto all'interno del parser. In linea generale, però, ricordatevi di mantenere sempre e comunque un controllo attivo su tutto il vostro sistema di variabili: all'aumentare della complessità del codice potrebbe essere estremamente difficile capire dove si genera un eventuale "bug" e, di conseguenza, riuscire a correggerlo.

```

/*-----*/
/* Parser sintattico      */
/*-----*/
Parser:
if words(frase)=1 then
  select
    when frase='GUARDA' then do
      verbo='GUARDA'; nome=''
      end
    when frase='LISTA' then do
      verbo='LISTA'; nome=''
      end
    when frase='QUIT' then do
      verbo='QUIT'; nome=''
      end
    when frase='VOC' then do
      verbo='VOC'; nome=''
      end
    when frase='DEBUG' then do
      verbo='DEBUG'; nome=''
      end
    otherwise say"non capisco. riprova"
  end
if words(frase)=2 then do
  verbo=left(frase,index(frase,' ')-1)
  nome=right(frase,length(frase)-index(frase,' '))
end
return

```

A questo punto il "gioco" è quasi fatto. Il passo seguente è quello di invocare il "main code" e richiamare, a seconda della situazione, la routine corrispondente:

```

select
  when verbo='VAI' then call Vai(nome, Pos)
  when verbo='GUARDA' then call Guarda()
  when verbo='PRENDI' then call Prendi(nome)
  when verbo='ESAMINA' then call Esamina(nome)
  when verbo='USA' then call Usa(nome)
  when verbo='LISTA' then call Lista()
  when verbo='VOC' then call Vocabolario()
  when verbo='DEBUG' then call Debug()
  when verbo='VAR' then call Var(nome)
  when verbo='QUIT' then nop

```

```

otherwise say 'non so che significa ' || verbo
end

```

Dal punto di vista concettuale il nostro lavoro è praticamente già completato. Quello che manca è definire come la gestione degli eventi e degli oggetti all'interno del gioco faccia avanzare la nostra storia. Nel prossimo numero analizzeremo ad un livello di dettaglio maggiore ciò che ancora ci serve per poter finalmente goderci la nostra avventura testuale.

4. CONCLUSIONI

Al giorno d'oggi sono molti i framework già pronti per confezionare avventure testuali e l'uso di essi è imprescindibile per lo sviluppatore che abbia necessità di concentrarsi sulla trama della storia, più che "perdere tempo" nello scrivere il codice che la gestisce, specialmente quando l'intenzione è quella di pubblicare su molte piattaforme diverse, siano esse "retro" o "next gen". Per il programmatore che invece predilige cimentarsi con la scrittura del proprio codice, avere un'idea di come implementare un mondo di gioco e riuscire finalmente a realizzarla è però una soddisfazione impagabile.

Nel prossimo numero continueremo ad esplorare AREXX alla scoperta di come scrivere le routine che effettivamente ci permetteranno di interagire in modo complesso per avanzare nella nostra storia e scoprirne ogni segreto.

BIBLIOGRAFIA

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

Utilizzo degli esempi

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

```
>rx nome_script.rexx"
o, piu' semplicemente:
">rx nome_script".
```



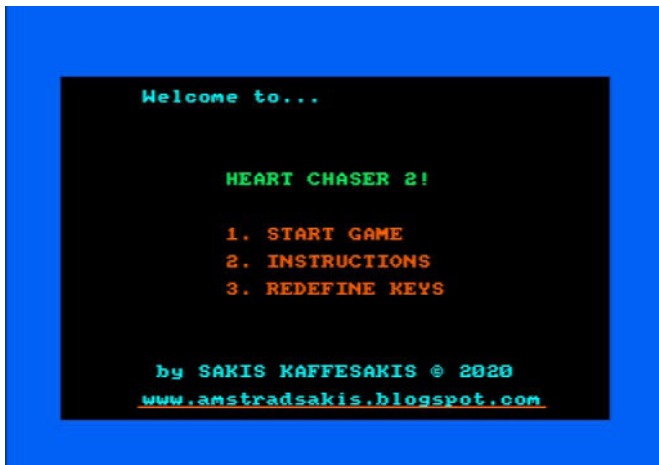


HEART CHASER 2 - un gioco in Locomotive Basic

di Sakis Kaffesakis - traduzione ed introduzione di Francesco Fiorentini

Non ho fatto in tempo a finire di scrivere l'articolo di chiusura del numero precedente, nel quale invitavo i lettori a partecipare con i loro lavori, che nella pagina di RetroMagazine veniamo contattati da Sakis Kafesakis, gestore del blog: <https://amstradsakis.blogspot.com/>. Sakis ci invitava a visitare il suo sito ed a dare un'occhiata ai suoi lavori in Locomotive Basic. L'occasione era troppo ghiotta per non approfittarne e quindi abbiamo provato a chiedergli se fosse stato disposto a condividere il codice di un suo gioco con noi, a vantaggio di tutti i lettori di RetroMagazine World.

Beh, la sua risposta non si e' fatta attendere e qui di seguito potete trovare il codice con tanto di spiegazioni di uno dei suoi ultimi giochi: **Heart Chaser 2**.



Heart Chaser 2 e' il sequel di Heart Chaser 1 (non lo avrei mai detto.. NdR) con nuove caratteristiche e livelli avanzati! In questo gioco dovete raccogliere i cuori, senza pero' essere catturati dai nemici sempre piu' agguerriti.

In ognuno dei 7 livelli devono essere raccolti 6 cuori, facendo attenzione ai nemici che diventano sempre piu' agguerriti di livello in livello.

Attenzione, il livello puo' essere completato se non si viene catturati, ma per passare al successivo dovete raccogliere obbligatoriamente i 6 cuori.

Hai solo 3 vite a disposizione per completare l'impresa. (Shh, c'e' un passaggio segreto nel menu' per ottenere 10 vite...).

Ecco il link diretto per scaricare il gioco:

<https://drive.google.com/drive/folders/1hZcn2TQ0GRmKUW4oY4IGR-eiDIePcan8>

Ed un longplay dello stesso:

https://www.youtube.com/watch?v=vxcUiwp18hM&feature=emb_title

https://www.youtube.com/watch?v=vxcUiwp18hM&feature=emb_title

Struttura del programma:

10-150 INTRO-MENU
150-900 MAIN GAME
1600-1690 DEFINING OPPONENTS MOVE
1900-1999 DEFINING INITIAL OPPONENT POSITION IN EVERY STAGE
2000-3080 INSTRUCTIONS- REDEFINE KEYS
4000-4610 STAGES DESIGN
4700-4830 LIMITING HEART APPEARANCE POINT DEPENDING ON STAGE
5000-5720 LIMITING PLAYER MOVE DEPENDING ON STAGE
6000-6720 LIMITING OPPONENT 1 MOVE DEPENDING ON STAGE
7000-7720 LIMITING OPPONENT 2 MOVE DEPENDING ON STAGE

X,Y = gestiscono la posizione del giocatore

OX,OY = posizione precedente del giocatore (se il giocatore prova a dirigersi verso un blocco o all'esterno della mappa, queste variabili sono utilizzate per forzare a tornare al punto precedente)

```
320 a$=INKEY$
330 IF A$=PLA$ AND X>1 AND X<40 THEN
OX=X:OY=Y:X=X-1:GOTO 390
340 IF A$=PLD$ AND X>1 AND X<40 THEN
OX=X:OY=Y:X=X+1:GOTO 390
350 IF A$=PLK$ AND Y>2 AND Y<25 THEN
OX=X:OY=Y:Y=Y+1:GOTO 390
360 IF A$=PLP$ AND Y>2 AND Y<25 THEN
OX=X:OY=Y:Y=Y-1:GOTO 390
365 IF DIFF=10 AND (A$="Y" OR A$="y" OR
A$="X" OR A$="Z") THEN GOTO 8000
370 GOTO 400
```





```

390 IF X=1 THEN X=2
391 IF X=40 THEN X=39
392 IF Y=2 THEN Y=3
393 IF Y=25 THEN Y=24
395 GOSUB 5000
399 LOCATE OX,OY:PRINT FIGK$
400 LOCATE X,Y:PRINT FIG1$

```

390-393 limiti di movimento standard per tutti i livelli.
 399 pulisce la posizione precedente del giocatore.
 400 visualizza il giocatore nella nuova posizione.

Lo stesso sistema e' utilizzato per lo spostamento dei nemici. XZ1 e YZ1 sono le variabili del primo nemico. La variabile ANT mostra la posizione successiva del nemico numero 1, in pratica dove il nemico 1 si sposterà'.

```

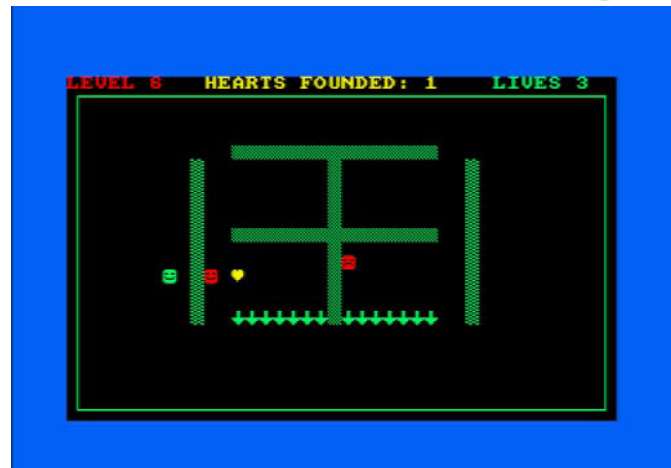
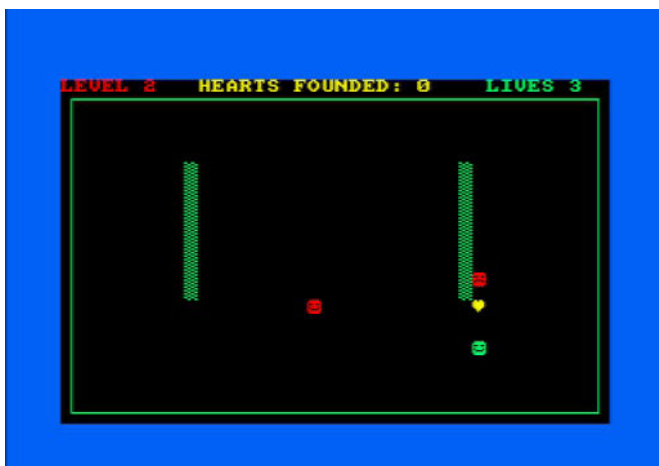
510 IF ANT=1 THEN OXZ1=XZ1:OYZ1=YZ1:XZ1=XZ1-1:GOTO 550
515 IF ANT=2 THEN
OXZ1=XZ1:OYZ1=YZ1:XZ1=XZ1+1:GOTO 550
520 IF ANT=3 THEN OXZ1=XZ1:OYZ1=YZ1:YZ1=YZ1-1:GOTO 550
525 IF ANT=4 THEN
OXZ1=XZ1:OYZ1=YZ1:YZ1=YZ1+1:GOTO 550
550 IF XZ1=1 THEN XZ1=2
560 IF XZ1=40 THEN XZ1=39
561 IF YZ1=2 THEN YZ1=3
562 IF YZ1=24 THEN YZ1=23
563 GOSUB 6000
580 LOCATE OXZ1,OYZ1:PRINT " "
590 PEN 2:LOCATE XZ1,YZ1:PRINT FIG8$:PEN 1

```

Intelligenza dei nemici:

La variabile DIFF setta la difficoltà' del gioco. E' settata a 4 all'inizio di ogni partita e viene incrementata di 1 ad ogni livello.

Incrementa l'abilità' dei nemici con il passare dei livelli.



La variabile ANT gestisce il successivo movimento del nemico 1. E' settata casualmente nella riga 500, ma poi viene gestita in una sotto-procedura: GOSUB 1600.

```
500 ANT=INT(RND*4)+1:GOSUB 1600
```

Nelle righe da 1610 a 1615, si evince come, con una DIFF settata a 4, c'e' un'elevata possibilita' di un RETURN alla riga 1615 e mantenere di conseguenza il valore casuale.

Nei livelli successivi, dove la variabile DIFF e' incrementata, la possibilita' di un movimento casuale decresce di conseguenza. Con il valore a 10, lo spostamento casuale e' del tutto scongiurato.

1620-1623 decidono il movimento del nemico quando si trova nelle stesse coordinate del giocatore (x o y).

Dalla riga 1630 viene deciso il valore di correzione da applicare se il giocatore ed il nemico hanno coordinate differenti.

Se il giocatore ed il nemico hanno x e y differenti, c'e' un 50% di probabilita' per il nemico di correggere la coordinata X ed un 50% di correggere la coordinata Y.

```

1610 ANT1=INT(RND*10)+1
1615 IF ANT1>DIFF THEN RETURN
1620 IF X=XZ1 AND Y<YZ1 THEN ANT=3:RETURN
1621 IF X=XZ1 AND Y>YZ1 THEN ANT=4:RETURN
1622 IF Y=YZ1 AND X<XZ1 THEN ANT=1:RETURN
1623 IF Y=YZ1 AND X>XZ1 THEN ANT=2:RETURN
1630 KAT=INT(RND*2)+1
1635 IF KAT=1 THEN GOTO 1650
1637 IF KAT=2 THEN GOTO 1654
1650 IF X<XZ1 THEN ANT=1
1652 IF X>XZ1 THEN ANT=2
1653 GOTO 1669
1654 IF Y<YZ1 THEN ANT=3

```





```
1656 IF Y>YZ1 THEN ANT=4
```

Per assicurare che, nell'ultimo livello i due oppponenti non facciano esattamente le stesse mosse, abbiamo reso impossibile per il primo oppponente (rappresentato dalla faccina triste) lo spostamento nella riga finale, mentre il

secondo oppponente (rappresentato dalla faccina sorridente) ha un piccola correzione nelle possibilita' delle mosse casuali. Come risultato, a volte, puo' capitare che il giocatore sia circondato dai nemici.

Di seguito il listato completo in Locomotive Basic.

```
10 REM HEART CHASER 2! COPYRIGHT BY SAKIS KAFFESAKIS 2020.
15 GOSUB 3100
16 GOSUB 3600
20 CLS:MODE 1:BORDER 11:PEN 3:INK 3,20:INK 0,0:RANDOMIZE TIME:SCORE=0:DIFF=4:LIVES=3
25 IF KRYFO=1 THEN LIVES=10
30 LOCATE 7,2:PRINT "Welcome to..."
40 LOCATE 13,8:PEN 2:SPEED INK 40,40:INK 2,19,24:PRINT "HEART CHASER 2!"
50 PEN 1:INK 1,15:LOCATE 13,12:PRINT"1. START GAME":LOCATE 13,14:PRINT"2.
INSTRUCTIONS":LOCATE 13,16:PRINT"3. REDEFINE KEYS"
60 LOCATE 8,22:PEN 3:PRINT"by SAKIS KAFFESAKIS":LOCATE 28,22:PRINT CHR$(164):LOCATE
30,22:PRINT"2020":LOCATE 7,24:PRINT"www.amstradsakis.blogspot.com":MOVE 90,15:DRAW 565,15
70 A$=INKEY$
80 IF A$="2" THEN GOSUB 2000:GOTO 20
90 IF A$="3" THEN GOSUB 3000:GOTO 20
100 IF A$="1" THEN GOTO 150
110 IF A$="L" OR A$="1" THEN GOSUB 8100:GOTO 30
140 GOTO 70
150 REM START GAME
160 REM START GAME
180 MODE 1:PEN 1:INK 1,19:
190 GOSUB 1000:GOSUB 1900
200 MOVE 13,13:DRAW 626,13:DRAW 626,376:DRAW 13,376:DRAW 13,13
250 LOCATE X,Y:PRINT FIG1$
260 PEN 2:INK 2,6:LOCATE XZ1,YZ1:PRINT FIG8$:LOCATE XZ2,YZ2:PRINT FIG9$:PEN 1
263 REM FOR DIFFK=1 TO 5:DIFF=2*DIFF
265 REM FOR ST=1 TO 2
280 PISTA=DIFF-3
285 ON PISTA GOSUB 4000,4100,4200,4300,4400,4500,4600
290 ON PISTA GOSUB 4700,4720,4740,4760,4780,4800,4820:REM
OBJX=INT(RND*37)+2:OBJY=INT(RND*21)+3
300 WHILE 1
301 LO=0:PISTA=DIFF-3
302 LOCATE 1,1:PEN 2:PRINT "LEVEL";DIFF-3:PEN 3:LOCATE 11,1:PRINT"HEARTS
FOUNDED:";SCORE:LOCATE 32,1:PEN 1:PRINT"LIVES";LIVES;
305 PEN 3:INK 3,24:LOCATE OBJX,OBJY:PRINT OBJ$:PEN 1
320 a$=INKEY$
330 IF A$=PLA$ AND X>1 AND X<40 THEN OX=X:OY=Y:X=X-1:GOTO 390
340 IF A$=PLD$ AND X>1 AND X<40 THEN OX=X:OY=Y:X=X+1:GOTO 390
350 IF A$=PLK$ AND Y>2 AND Y<25 THEN OX=X:OY=Y:Y=Y+1:GOTO 390
360 IF A$=PLP$ AND Y>2 AND Y<25 THEN OX=X:OY=Y:Y=Y-1:GOTO 390
365 IF DIFF=10 AND (A$="Y" OR A$="y" OR A$="X" OR A$="Z") THEN GOTO 8000
370 GOTO 400
390 IF X=1 THEN X=2
391 IF X=40 THEN X=39
392 IF Y=2 THEN Y=3
393 IF Y=25 THEN Y=24
395 GOSUB 5000
399 LOCATE OX,OY:PRINT FIGK$
400 LOCATE X,Y:PRINT FIG1$
430 IF X=OBJX AND Y=OBJY THEN score=SCORE+1:OBJX=INT(RND*37)+2:OBJY=INT(RND*21)+3:GOSUB
3500:ON PISTA GOSUB 4700,4720,4740,4760,4780,4800,4820
450 IF X=XZ1 AND Y=YZ1 THEN GOTO 1500
451 IF X=XZ2 AND Y=YZ2 THEN GOTO 1592
500 ANT=INT(RND*4)+1:GOSUB 1600
505 OXZ1=XZ1:OYZ1=YZ1
510 IF ANT=1 THEN OXZ1=XZ1:OYZ1=YZ1:XZ1=XZ1-1:GOTO 550
515 IF ANT=2 THEN OXZ1=XZ1:OYZ1=YZ1:XZ1=XZ1+1:GOTO 550
520 IF ANT=3 THEN OXZ1=XZ1:OYZ1=YZ1:YZ1=YZ1-1:GOTO 550
525 IF ANT=4 THEN OXZ1=XZ1:OYZ1=YZ1:YZ1=YZ1+1:GOTO 550
550 IF XZ1=1 THEN XZ1=2
560 IF XZ1=40 THEN XZ1=39
561 IF YZ1=2 THEN YZ1=3
562 IF YZ1=24 THEN YZ1=23
563 GOSUB 6000
580 LOCATE OXZ1,OYZ1:PRINT " "
```





```

590 PEN 2:LOCATE XZ1,YZ1:PRINT FIG8$:PEN 1
600 IF X=XZ1 AND Y=YZ1 THEN GOTO 1500
601 IF X=XZ2 AND Y=YZ2 THEN GOTO 1592
605 ANTIP=INT(RND*4)+1:GOSUB 1670
606 OXZ2=XZ2:OYZ2=YZ2
607 IF ANTIP=1 THEN OXZ2=XZ2:OYZ2=YZ2:XZ2=XZ2-1:GOTO 612
608 IF ANTIP=2 THEN OXZ2=XZ2:OYZ2=YZ2:XZ2=XZ2+1:GOTO 612
609 IF ANTIP=3 THEN OXZ2=XZ2:OYZ2=YZ2:YZ2=YZ2-1:GOTO 612
610 IF ANTIP=4 THEN OXZ2=XZ2:OYZ2=YZ2:YZ2=YZ2+1:GOTO 612
612 IF XZ2=1 THEN XZ2=2
613 IF XZ2=40 THEN XZ2=39
614 IF YZ2=2 THEN YZ2=3
615 IF YZ2=25 THEN YZ2=24
616 GOSUB 7000
617 LOCATE OXZ2,OYZ2:PRINT " "
618 PEN 2:LOCATE XZ2,YZ2:PRINT FIG9$:PEN 1
619 IF X=XZ1 AND Y=YZ1 THEN GOTO 1500
620 IF X=XZ2 AND Y=YZ2 THEN GOTO 1592
622 IF SCORE=6 THEN SCORE=0:DIFF=DIFF+1:LO=1
623 IF DIFF=11 THEN GOTO 950
625 IF LO=1 THEN CLS:LOCATE 17,13:PRINT "LEVEL";DIFF-3:GOSUB 1496:CLS:GOTO 150
900 WEND
950 MODE 1:LOCATE 17,10:PRINT"YOU WIN!":LOCATE 10,18:PRINT"PRESS SPACE TO RESTART":GOSUB
8200:GOSUB 1495:SCORE=0:DIFF=1
952 R$=INKEY$
954 IF R$=" " THEN GOTO 20
956 GOTO 952
998 END
999 MODE 2:PEN 1:INK 1,19:END
1000 FIG1$=CHR$(224):X=20:Y=20:FIGK$=" "
1005 INK 1,19:INK 2,6:INK 3,24
1010 FIG8$=CHR$(225):XZ1=5:YZ1=5:ekr$=CHR$(252)
1012 IF DIFF<4 THEN XZ1=5:YZ1=5
1013 IF DIFF>4 AND DIFF<7 THEN XZ1=30:YZ1=10
1014 IF DIFF>7 AND DIFF<10 THEN XZ1=20:YZ1=10
1015 IF DIFF=10 THEN XZ1=20:YZ1=18
1020 REM !!!!!!!!!!!!!!!SETTING DIFFICULTY!!!!!!!!!!!!!!!!!!!!!!
1030 OBJ$=CHR$(228)
1040 FIG9$=CHR$(224)
1050 TU$=CHR$(207):TUK$=CHR$(241):TUX$=CHR$(206):TUD$=CHR$(127):TUM$=CHR$(237):TUU$=CHR$(229)
1490 RETURN
1495
W1$=INKEY$:W2$=INKEY$:W3$=INKEY$:W4$=INKEY$:W5$=INKEY$:W6$=INKEY$:W1$=INKEY$:W2$=INKEY$:W3$=I
NKEY$:W4$=INKEY$:W5$=INKEY$:W6$=INKEY$:RETURN
1496 R$=INKEY$
1498 IF R$=" " THEN RETURN
1499 GOTO 1496
1500 SOUND 1,800,20,15,,,15:LOCATE XZ1,YZ1:PEN 2:PRINT EKR$:PEN 1:LIVES=LIVES-1:IF LIVES=0
THEN GOTO 1700
1548 GOSUB 1495
1550 D$=INKEY$
1560 IF D$="" THEN GOTO 1550
1570 MODE 1:LOCATE 17,20:PRINT"LIFE LOST":GOSUB 1495
1580 D$=INKEY$
1590 IF D$=" " THEN GOTO 150
1591 GOTO 1580
1592 SOUND 1,800,20,15,,,15:LOCATE XZ2,YZ2:PEN 2:PRINT EKR$:PEN 1:LIVES=LIVES-1:IF LIVES=0
THEN GOTO 1700
1593 GOSUB 1495
1594 D$=INKEY$
1595 IF D$="" THEN GOTO 1594
1596 MODE 1:LOCATE 17,20:PRINT"LIFE LOST":GOSUB 1495
1597 D$=INKEY$
1598 IF D$=" " THEN GOTO 150
1599 GOTO 1597
1600 REM
1610 ANT1=INT(RND*10)+1
1615 IF ANT1>DIFF THEN RETURN
1620 IF X=XZ1 AND Y<YZ1 THEN ANT=3:RETURN
1621 IF X=XZ1 AND Y>YZ1 THEN ANT=4:RETURN
1622 IF Y=YZ1 AND X<XZ1 THEN ANT=1:RETURN
1623 IF Y=YZ1 AND X>XZ1 THEN ANT=2:RETURN
1630 KAT=INT(RND*2)+1
1635 IF KAT=1 THEN GOTO 1650
1637 IF KAT=2 THEN GOTO 1654

```





```
1650 IF X<XZ1 THEN ANT=1
1652 IF X>XZ1 THEN ANT=2
1653 GOTO 1669
1654 IF Y<YZ1 THEN ANT=3
1656 IF Y>YZ1 THEN ANT=4
1669 RETURN
1670 ANTIPI=INT(RND*10)+1:IF DIFF=10 THEN ANTIPI=ANTIPI+1
1671 IF ANTIPI>DIFF THEN RETURN
1672 IF X=XZ2 AND Y<YZ2 THEN ANTIPI=3:RETURN
1673 IF X=XZ2 AND Y>YZ2 THEN ANTIPI=4:RETURN
1674 IF Y=YZ2 AND X<XZ2 THEN ANTIPI=1:RETURN
1675 IF Y=YZ2 AND X>XZ2 THEN ANTIPI=2:RETURN
1676 KAT2=INT(RND*2)+1
1677 IF KAT2=1 THEN GOTO 1680
1678 IF KAT2=2 THEN GOTO 1683
1680 IF X<XZ2 THEN ANTIPI=1
1681 IF X>XZ2 THEN ANTIPI=2
1682 GOTO 1690
1683 IF Y<YZ2 THEN ANTIPI=3
1684 IF Y>YZ2 THEN ANTIPI=4
1690 RETURN
1700 GOSUB 1496:MODE 1:LOCATE 17,20:PRINT"GAME OVER":LOCATE 10,18:PRINT"PRESS SPACE TO
RESTART":GOSUB 8300:GOSUB 1495
1710 D$=INKEY$
1720 IF D$=" " THEN GOTO 20
1730 GOTO 1710
1826 IF OBJX=26 AND OBJY=22 THEN GOTO 4820
1900 IF DIFF=4 THEN XZ1=15:YZ1=8
1902 IF DIFF=5 THEN XZ1=31:YZ1=10
1904 IF DIFF=6 THEN XZ1=32:YZ1=10
1906 IF DIFF=7 THEN XZ1=24:YZ1=10
1908 IF DIFF=8 THEN XZ1=20:YZ1=10
1910 IF DIFF=9 THEN XZ1=29:YZ1=12
1912 IF DIFF=10 THEN XZ1=29:YZ1=15
1920 IF DIFF=4 THEN XZ2=30:YZ2=10
1922 IF DIFF=5 THEN XZ2=11:YZ2=10
1924 IF DIFF=6 THEN XZ2=8:YZ2=10
1926 IF DIFF=7 THEN XZ2=14:YZ2=10
1928 IF DIFF=8 THEN XZ2=25:YZ2=10
1930 IF DIFF=9 THEN XZ2=11:YZ2=12
1932 IF DIFF=10 THEN XZ2=12:YZ2=15
1999 RETURN
2000 MODE 2:PEN 1:INK 1,19:LOCATE 1,3:PRINT"In this game, you have to collect the hearts, and
avoid the angry enemies.":LOCATE 1,5:PRINT"You have to collect six hearts for each level."
2010 LOCATE 1,7:PRINT"There are 7 levels, and the intelligence of the enemies increases in
each level.":LOCATE 1,9:PRINT"You have only 3 lives!"
2015 LOCATE 1,13:PRINT"In the final stage (level 7) you can enable <save yourself>
feature,":LOCATE 1,15:PRINT"by pressing <Y> or joystick fire button quickly, if you are in a
great danger."
2016 LOCATE 1,17:PRINT"Then, you will start from the previous level again but with 1 extra
life!"
2020 LOCATE 36,22:PRINT"HAVE FUN!"
2030 GOSUB 1495
2040 T$=INKEY$
2050 IF T$<>" " THEN GOTO 20
2060 GOTO 2040
2999 RETURN
3000 PEN 1:INK 1,19:MODE 2:LOCATE 4,6:PRINT"1. JOYSTICK KEYS":LOCATE 4,10:PRINT"2. a w s d
KEYS":LOCATE 4,14:PRINT"3. q a o p KEYS"
3005 LOCATE 3,18:PRINT"PLS DO NOT USE CAPITAL KEYS"
3010 K$=INKEY$
3020 IF K$="1" THEN PLK$="\n":PLP$=" ":PLA$=" ":PLD$="\t":GOTO 3060
3030 IF K$="2" THEN PLK$="s":PLP$="w":PLA$="a":PLD$="d":GOTO 3060
3040 IF K$="3" THEN PLK$="a":PLP$="q":PLA$="o":PLD$="p":GOTO 3060
3050 GOTO 3010
3060 LOCATE 16,22:PRINT "SELECTION OK"
3065 L$=INKEY$
3070 IF L$="" THEN GOTO 3065
3080 RETURN
3100 PLK$="\n":PLP$=" ":PLA$=" ":PLD$="\t":RETURN
3500 REM
3510 SOUND 1,90,10,15:SOUND 1,85,7,15:SOUND 1,80,7,15:SOUND 1,75,7,15:
3599 RETURN
3600 FOR SOU=1 TO 2:SOUND 1,80,10,15:SOUND 1,67,10,15:SOUND 1,60,15,15:SOUND 1,0,5:NEXT SOU
3649 RETURN
```





```

4000 FOR BL=1 TO 10:LOCATE 15+BL,10:PRINT TU$:LOCATE 15+BL,17:PRINT TU$:NEXT BL:RETURN
4100 FOR BL=1 TO 10:LOCATE 10,6+BL:PRINT TUD$:LOCATE 30,6+BL:PRINT TUD$:NEXT BL:RETURN
4200 FOR BL=1 TO 15:LOCATE 12+BL,6:PRINT TU$:LOCATE 12+BL,12:PRINT TUD$:LOCATE 12+BL,18:PRINT
TUX$:NEXT BL:RETURN
4300 FOR BL=1 TO 12:LOCATE 10,6+BL:PRINT TU$:LOCATE 20,6+BL:PRINT TUX$:LOCATE 30,6+BL:PRINT
TU$:NEXT BL:RETURN
4400 FOR BL=1 TO 12:LOCATE 10,5+BL:PRINT TUU$:LOCATE 30,5+BL:PRINT TUU$:NEXT BL
4401 FOR BL=1 TO 19:LOCATE 10+BL,6:PRINT TUU$:NEXT BL
4402 RETURN
4500 FOR BL=1 TO 15:LOCATE 12+BL,6:PRINT TU$:LOCATE 12+BL,12:PRINT TU$:LOCATE 12+BL,18:PRINT
TUK$:NEXT BL
4501 FOR BL=1 TO 12:LOCATE 10,6+BL:PRINT TUD$:LOCATE 20,6+BL:PRINT TU$:LOCATE 30,6+BL:PRINT
TUD$:NEXT BL
4502 RETURN
4600 FOR BL=1 TO 12:FOR BL1=1 TO 38:LOCATE 1+BL1,2+BL:PRINT TU$:NEXT BL1:NEXT BL
4601 FOR BL=1 TO 11:FOR BL1=1 TO 10:LOCATE 1+BL1,13+BL:PRINT TU$:NEXT BL1:NEXT BL:LOCATE 1,1
4602 FOR BL=1 TO 11:FOR BL1=1 TO 10:LOCATE 40-BL1,13+BL:PRINT TU$:NEXT BL1:NEXT BL:LOCATE 1,1
4603 LOCATE 15,18:PRINT TU$:LOCATE 15,22:PRINT TU$:LOCATE 26,18:PRINT TU$:LOCATE 26,22:PRINT
TU$
4604 LOCATE 6,4:PRINT"www.amstradsakis.blogspot.com"
4605 LOCATE 10,12:PRINT"YOU WILL NOT SURVIVE!!"
4606 LOCATE 4,20:PRINT"RUN":LOCATE 5,21:PRINT"OR":LOCATE 5,22:PRINT"DIE!":LOCATE
35,20:PRINT"I":LOCATE 33,21:PRINT"WILL":LOCATE 32,22:PRINT"CHASE":LOCATE 33,23:PRINT"YOU!"
4607 LOCATE 13,7:PRINT"Save <y>ourself!"
4610 RETURN
4700 REM OBJX=INT(RND*37)+2:OBJY=INT(RND*21)+3
4701 OBJX=INT(RND*37)+2:OBJY=INT(RND*21)+3
4702 IF OBJX>15 AND OBJX<26 AND (OBJY=10 OR OBJY=17) THEN GOTO 4701
4703 RETURN
4719 RETURN
4720 OBJX=INT(RND*37)+2:OBJY=INT(RND*21)+3
4721 IF (OBJX=10 OR OBJX=30) AND OBJY>6 AND OBJY<17 THEN GOTO 4720
4722 RETURN
4740 OBJX=INT(RND*27)+7:OBJY=INT(RND*13)+3
4741 IF OBJX>12 AND OBJX<28 AND (OBJY=6 OR OBJY=12 OR OBJY=18) THEN GOTO 4740
4742 RETURN
4760 OBJX=INT(RND*27)+7:OBJY=INT(RND*13)+3
4761 IF (OBJX=10 OR OBJX=20 OR OBJX=30) AND OBJY>6 AND OBJY<19 THEN GOTO 4760
4762 RETURN
4780 OBJX=INT(RND*27)+7:OBJY=INT(RND*13)+7
4781 IF (OBJX=10 OR OBJX=30) AND OBJY>5 AND OBJY<18 THEN GOTO 4780
4782 IF OBJX>9 AND OBJX<31 AND OBJY=6 THEN GOTO 4780
4783 RETURN
4800 OBJX=INT(RND*24)+8:OBJY=INT(RND*13)+7
4801 IF (OBJX=10 OR OBJX=20 OR OBJX=30) AND OBJY>6 AND OBJY<19 THEN GOTO 4800
4802 IF OBJX>12 AND OBJX<28 AND (OBJY=6 OR OBJY=12 OR OBJY=18) THEN GOTO 4800
4805 RETURN
4820 OBJX=INT(RND*37)+2:OBJY=INT(RND*21)+3
4821 IF OBJX<12 OR OBJX>29 THEN GOTO 4820
4822 IF OBJY<15 THEN GOTO 4820
4823 IF OBJX=15 AND OBJY=18 THEN GOTO 4820
4824 IF OBJX=15 AND OBJY=22 THEN GOTO 4820
4825 IF OBJX=26 AND OBJY=18 THEN GOTO 4820
4830 RETURN
5000 REM PAIXTIS
5010 ON PISTA GOSUB 5100,5200,5300,5400,5500,5600,5700
5020 RETURN
5100 IF X>15 AND X<26 AND (Y=10 OR Y=17) THEN X=OX:Y=OY
5199 RETURN
5200 IF (X=10 OR X=30) AND Y>6 AND Y<17 THEN X=OX:Y=OY
5201 RETURN
5300 IF X>12 AND X<28 AND (Y=6 OR Y=12 OR Y=18) THEN X=OX:Y=OY
5301 RETURN
5400 IF (X=10 OR X=20 OR X=30) AND Y>6 AND Y<19 THEN X=OX:Y=OY
5401 RETURN
5500 IF (X=10 OR X=30) AND Y>5 AND Y<18 THEN X=OX:Y=OY
5501 IF X>9 AND X<31 AND Y=6 THEN X=OX:Y=OY
5502 RETURN
5600 IF X>12 AND X<28 AND (Y=6 OR Y=12 OR Y=18) THEN X=OX:Y=OY
5601 IF (X=10 OR X=20 OR X=30) AND Y>6 AND Y<19 THEN X=OX:Y=OY
5605 RETURN
5700 IF X<12 OR X>29 OR Y<15 THEN X=OX:Y=OY
5701 IF X=15 AND Y=18 THEN X=OX:Y=OY
5702 IF X=15 AND Y=22 THEN X=OX:Y=OY
5703 IF X=26 AND Y=18 THEN X=OX:Y=OY

```





```
5704 IF X=26 AND Y=22 THEN X=OX:Y=OY
5720 RETURN
6000 REM ANTIPALOS 1
6010 ON PISTA GOSUB 6100,6200,6300,6400,6500,6600,6700
6020 RETURN
6100 IF XZ1>15 AND XZ1<26 AND (YZ1=10 OR YZ1=17) THEN XZ1=OXZ1:YZ1=OYZ1
6199 RETURN
6200 IF (XZ1=10 OR XZ1=30) AND YZ1>6 AND YZ1<17 THEN XZ1=OXZ1:YZ1=OYZ1
6201 RETURN
6300 IF XZ1>12 AND XZ1<28 AND (YZ1=6 OR YZ1=12 OR YZ1=18) THEN XZ1=OXZ1:YZ1=OYZ1
6301 RETURN
6400 IF (XZ1=10 OR XZ1=20 OR XZ1=30) AND YZ1>6 AND YZ1<19 THEN XZ1=OXZ1:YZ1=OYZ1
6401 RETURN
6500 IF (XZ1=10 OR XZ1=30) AND YZ1>5 AND YZ1<18 THEN XZ1=OXZ1:YZ1=OYZ1
6501 IF XZ1>9 AND XZ1<31 AND YZ1=6 THEN XZ1=OXZ1:YZ1=OYZ1
6502 RETURN
6600 IF XZ1>12 AND XZ1<28 AND (YZ1=6 OR YZ1=12 OR YZ1=18) THEN XZ1=OXZ1:YZ1=OYZ1
6601 IF (XZ1=10 OR XZ1=20 OR XZ1=30) AND YZ1>6 AND YZ1<19 THEN XZ1=OXZ1:YZ1=OYZ1
6605 RETURN
6700 IF XZ1<12 OR XZ1>29 OR YZ1<15 THEN XZ1=OXZ1:YZ1=OYZ1
6701 IF XZ1=15 AND YZ1=18 THEN XZ1=OXZ1:YZ1=OYZ1
6702 IF XZ1=15 AND YZ1=22 THEN XZ1=OXZ1:YZ1=OYZ1
6703 IF XZ1=26 AND YZ1=18 THEN XZ1=OXZ1:YZ1=OYZ1
6704 IF XZ1=26 AND YZ1=22 THEN XZ1=OXZ1:YZ1=OYZ1
6720 RETURN
7000 REM ANTIPALOS 2
7010 ON PISTA GOSUB 7100,7200,7300,7400,7500,7600,7700
7020 RETURN
7100 IF XZ2>15 AND XZ2<26 AND (YZ2=10 OR YZ2=17) THEN XZ2=OXZ2:YZ2=OYZ2
7199 RETURN
7200 IF (XZ2=10 OR XZ2=30) AND YZ2>6 AND YZ2<17 THEN XZ2=OXZ2:YZ2=OYZ2
7201 RETURN
7300 IF XZ2>12 AND XZ2<28 AND (YZ2=6 OR YZ2=12 OR YZ2=18) THEN XZ2=OXZ2:YZ2=OYZ2
7301 RETURN
7400 IF (XZ2=10 OR XZ2=20 OR XZ2=30) AND YZ2>6 AND YZ2<19 THEN XZ2=OXZ2:YZ2=OYZ2
7401 RETURN
7500 IF (XZ2=10 OR XZ2=30) AND YZ2>5 AND YZ2<18 THEN XZ2=OXZ2:YZ2=OYZ2
7501 IF XZ2>9 AND XZ2<31 AND YZ2=6 THEN XZ2=OXZ2:YZ2=OYZ2
7502 RETURN
7600 IF XZ2>12 AND XZ2<28 AND (YZ2=6 OR YZ2=12 OR YZ2=18) THEN XZ2=OXZ2:YZ2=OYZ2
7601 IF (XZ2=10 OR XZ2=20 OR XZ2=30) AND YZ2>6 AND YZ2<19 THEN XZ2=OXZ2:YZ2=OYZ2
7605 RETURN
7700 IF XZ2<12 OR XZ2>29 OR YZ2<15 THEN XZ2=OXZ2:YZ2=OYZ2
7701 IF XZ2=15 AND YZ2=18 THEN XZ2=OXZ2:YZ2=OYZ2
7702 IF XZ2=15 AND YZ2=22 THEN XZ2=OXZ2:YZ2=OYZ2
7703 IF XZ2=26 AND YZ2=18 THEN XZ2=OXZ2:YZ2=OYZ2
7704 IF XZ2=26 AND YZ2=22 THEN XZ2=OXZ2:YZ2=OYZ2
7720 RETURN
8000 CLS:MODE 2:PEN 1:LOCATE 3,19:PRINT"YOU SAVED YOURSELF BUT YOU DID NOT COLLECT THE
HEARTS": LOCATE 3,21:PRINT"PRESS SPACE TO TRY AGAIN FROM THE PREVIOUS LEVEL"
8005 LOCATE 3,23:PRINT"+1 EXTRA LIFE!"
8007 SOUND 1,60,8,14:SOUND 1,95,9,14:SOUND 1,119,8,14
8010 F$=INKEY$
8020 IF F$=" " THEN GOTO 8040
8030 GOTO 8010
8040 DIFF=9:score=0:LIVES=LIVES+1:GOTO 150
8100 LIVES=10:KRYFO=1:MODE 1:LOCATE 9,15:PRINT"YOU UNLOCKED 10 LIVES!!":LOCATE
9,17:PRINT"PRESS SPACE TO CONTINUE"
8105 SOUND 1,119,8,14:SOUND 1,95,8,14:SOUND 1,60,8,14
8110 F$=INKEY$
8115 IF F$=" " THEN MODE 1:RETURN
8120 GOTO 8110
8200 SOUND 1,119,10,14:SOUND 1,106,10,14:SOUND 1,95,10,14:SOUND 1,89,10,14:SOUND
1,80,10,14:SOUND 1,71,10,14:SOUND 1,67,10,14:SOUND 1,63,10,14:SOUND 1,60,20,14:RETURN
8300 SOUND 1,60,10,15:SOUND 1,63,10,15:SOUND 1,71,10,15:SOUND 1,80,10,15:SOUND
1,89,10,15:SOUND 1,95,10,15:SOUND 1,106,10,15:SOUND 1,119,10,15
8310 SOUND 1,179,10,15:SOUND 1,190,10,15:SOUND 1,213,10,15:SOUND 1,239,10,15:SOUND
1,253,10,15:SOUND 1,284,10,15:SOUND 1,319,10,15:SOUND 1,358,10,15
8320 SOUND 1,478,10,15:SOUND 1,506,10,15:SOUND 1,568,10,15:SOUND 1,638,10,15:SOUND
1,716,10,15:SOUND 1,758,10,15:SOUND 1,851,10,15:SOUND 1,956,50,15
8330 RETURN
```





YAMI NO RYŪŌ HADESU NO MONSHŌ

Era mai possibile che dopo l'uscita di Knightmare e del successo ottenuto, nessuno volesse tentare il colpaccio commerciale con un clone che ne fosse realmente all'altezza?

Ovviamente no!

Ma ci stava un problema non di poco conto. Knightmare per l'epoca era un'opera d'arte; colonna sonora incredibile, grafica fluidissima, giocabilità alle stelle, elementi di gioco disegnati con tale maestria da sembrare veri, incredibile uso della profondità ecc... Chiunque lo avesse sfidato difficilmente avrebbe potuto evitare una gran figuraccia, tanto che si fece avanti un vero e proprio colosso ludico. Qualcuno che ai tempi aveva la forza economica di spazzare via chiunque, sia in fatto di Hardware che di Software... La CASIO! Gli andò bene vi starete chiedendo? Le intenzioni erano buone, ma provare a combattere un gioco con un clone è già di suo un fallimento totale.

Il gioco in questione uscì in Giappone con il titolo di Yami no ryūō hadesu no monshō, ribattezzato Leonidas in alcuni paesi e Crest of the Dragon King Hades of Darkness in Inghilterra, The seal of Hades negli Usa e Crest of The Royal Family in Corea... Insomma, aveva un nome per ogni occasione; una cosa almeno era certa... il progetto era destinato al solo mercato MSX il cui scopo lo abbiamo capito benissimo.

Nonostante una colonna sonora molto articolata e ben fatta, una fluidità da manuale e una grafica decente, purtroppo il gioco non vendette secondo le aspettative; troppo simile a Knightmare e troppo difficile. La cosa peggiore inoltre, fu un errore commerciale senza precedenti... Knightmare uscì come distribuzione alla vendita tra il dicembre 1985 e gennaio 1986, giusto in tempo per i regali di Natale, mentre l'annuncio del gioco da parte della Casio e della sua futura vendita avvenne proprio a

ridosso di tale periodo!

I ragazzi che comprarono Knightmare persero totalmente l'interesse di giocare ad un suo clone. Leonidas, che fu realizzato in fretta e furia per uscire nell'estate del 1986, ebbe anche una pubblicità negativa da parte delle maggiori riviste specializzate che gridarono al plagio e tutto passò in sordina anche in occidente. Il gioco fu dimenticato in un attimo, e venne avvolto dall'oblio. Ecco spiegato il motivo per cui di questo gioco troviamo davvero poche informazioni e molti sicuramente lo sentono nominare per la prima volta, anche se, senza Knightmare di mezzo, sarebbe stato un bestseller senza tempo e acclamato da tutti i giocatori del pianeta.

Ora passiamo alle curiosità, per coprire la figuraccia fatta, la Casio lo presentò come il seguito di "The Stone of Wisdom" (il suo 25° gioco) che, manco a farlo apposta era uscito in contemporanea a Knightmare all'inizio del 1986. Ma The Stone of Wisdom era un gioco completamente diverso; si trattava di un RPG, molto bello, pieno di dungeon e con un sacco di livelli. Un gran bel gioco, ma che non aveva niente a che vedere né con il suo pseudo sequel Leonidas né con Knightmare...

Leonidas ha sei livelli, leggermente più lunghi rispetto al gioco Konami ed molto difficile da ultimare. Gioco frenetico e velocissimo ma calibrato in modo che a tratti sembra che puoi vincere facilmente, per poi morire miseramente. Ti costringe ad andare a prendere il calendario per ricordarti il nome di tutti i Santi... :-)

Se vi è piaciuto Knightmare, è un ottimo diversivo riservato ai veri game. Lo potete trovare e giocare al seguente sito:

<https://www.file-hunter.com/MSX/index.php?id=leonidas>

di **Ermanno Betori**

Anno: 1986

Sviluppatore: CASIO

Piattaforma: MSX1

Genere: RPG



GIUDIZIO FINALE

» Giocabilità 90%

Giocabilità elevata. Grafica curata sfrutta bene le caratteristiche del computer MSX1. Sonoro con un motivo musicale ben curato che accompagna bene le fasi del gioco. Giocando vi sono vari bonus da prendere tra i quali la riserva di ossigeno infatti il nostro eroe sembra un palombaro o simile.

» Longevità 70%

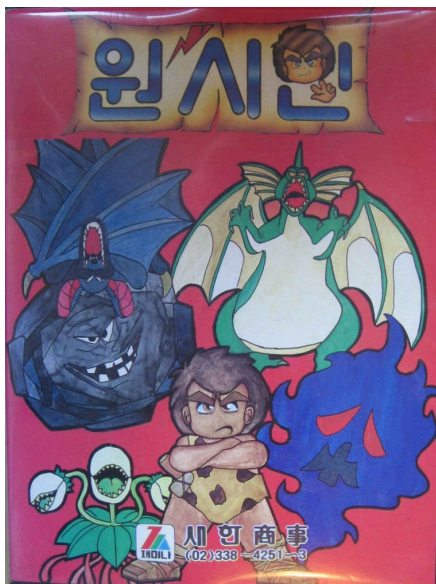
Un gioco difficile; un novizio di arcade non riesce nemmeno a finire il primo livello. Ma non è impossibile, anche se in certi punti è veramente tosto.





WON-SI-IN

L'altro clone di Knightmare, che ora esaminiamo, è un gioco rarissimo prodotto da Zemina, società della Corea del Sud nel 1991 che si chiama WON-SI-IN che tradotto significa uomo preistorico.



Wonsiin è un porting del gioco Famicom Shin Jinrui: The New Type (Adventure of Dino Riki negli Stati Uniti), che apparentemente ha dato a Zemina l'opportunità per sfruttare il loro codice di scrolling verticale nel porting con tanto di instabilità che è presente nell'originale Famicom.

A parte altri piccoli bug il porting è perfetto, persino i Dinosauri non sono stati cambiati di un solo pixel e l'impresa non era per niente



semplice. Tuttavia, Wonsiin termina dopo tre livelli completi, tralasciando le sfide finali e lo scontro finale con il boss supremo che assomiglia a Bowser di Mario Bros!

Il gioco vede impegnato un cavernicolo a sopravvivere lungo un percorso irto di difficoltà, aggiornando la sua arma da lancia ad asce di pietra, boomerang e persino lanciafiamme, seppur difficile Wonsiin non lo è mai quanto The Three Dragon Story.

All'inizio del 1992, Zemina aveva annunciato il sequel Wonsiin 2, che non è stato completato prima che ci fosse la fine della alleanza tra la Saehan Sangsa e la Zemina, Kim Eulsuk andò via e con il suo staff formò la ditta Open, e realizzarono il sequel su master system come Wonder Kid/Adventure Kid che sembra essere a tutti gli effetti un clone di Wonder Boy.

Per gli appassionati retro gamer collezionisti la cartuccia oggi ha un valore commerciale molto alto ed oltretutto è veramente introvabile cosa che ha contribuito all'oblio di tale gioco.

Grafica molto curata che sfrutta bene le caratteristiche del computer MSX1.

Sonoro: motivo musicale ripetitivo ma ben curato, forse non propriamente adatto allo storyboard del gioco... Magari nei livelli successivi cambia il motivetto, ma il sottoscritto viene annientato al primo livello!

di **Ermanno Betori**

Anno: 1991
Sviluppatore: Zemina

Piattaforma: MSX1
Genere: RPG



GIUDIZIO FINALE

» Giocabilità 88%

Veramente difficile, in confronto Knightmare sembra una passeggiata. Consigliato al vero Super Arcade Man.

» Longevità 70%

Tre soli livelli, ma la difficoltà ve li farà durare per molto tempo... Sempre se non ne sarete frustrati prima.

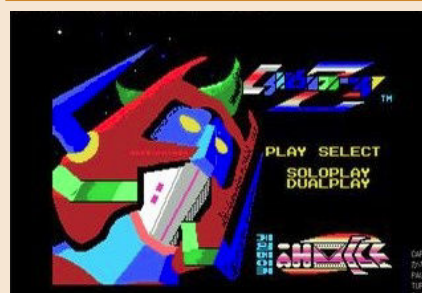




CYBORG-Z

Anno: 1991
Sviluppatore: Zemina

Piattaforma: MSX1
Genere: RPG



GIUDIZIO FINALE

» **Giocabilità 93%**

Elevata, equivalente come confronto a Nightmare!

» **Longevità 90%**

Diificolta' calibrata egregiamente. Lo rigiocherete!

Concludiamo i cloni di Nightmare parlando ancora della plagiatrice per eccellenza... La Zemina!

Uno dei suoi giochi più rari in assoluto si chiama Cyborg-z, uscito anche lui nel 1991 su MSX fu uno degli ultimi giochi sviluppati sulla piattaforma 8 bit da parte di Zemina che non destò molto interesse da parte del pubblico. I ragazzi all'epoca già smanettavano su macchine più potenti e la tiratura del gioco fu piuttosto contenuta.

L'unico filmato noto che provava la esistenza di questo gioco, ritenuto perso per sempre, proveniva da un proprietario giapponese. Un gruppo di appassionati iniziò quindi una vera e propria caccia al tesoro che si concluse con il ritrovamento di una cartuccia originale. La società aveva finalmente effettuato il passaggio alle rom Megabit (equivalenti a 128kB), quindi questo gioco contiene molta più grafica rispetto ai precedenti prodotti Zemina. Cyborg Z è uno sparattuto a scorrimento verticale molto simile a Nightmare, anche il gameplay è incredibilmente simile con poche differenze.

Il personaggio è una copia del mecha gigante Mazinger Z della serie anime del 1972, incluso il suo attacco Rocket Punch. Il gioco ha lo stesso sistema di armi di The Three Dragon Story, con le armi multiple che possono essere cambiate, ma e' molto più facile perche' al mecha sono concesse molte vite extra. Il mecha ha una barra di potenza e una barra di velocità; la barra di potenza si esaurisce quando il personaggio viene danneggiato dai nemici. La maggior parte delle armi, tranne il razzo e il raggio laser, hanno

munizioni limitate. Esistono due tipi di elementi di invincibilità. L'elemento "α" invincibile rosso può uccidere i nemici, ma l'elemento "β" invincibile blu non può. Ogni fase ha un boss centrale e un boss finale.

Dopo aver superato ogni fase, troveremo una roulette bonus. Se la freccia atterra su uno spazio giallo, è possibile selezionare un elemento e si può ripetere all'infinito finendo sempre sul giallo.

Esiste anche una modalità "dualplay" per due giocatori. Cosa particolare per questo tipo di giochi è che parte della musica di sottofondo del gioco è composta da brani di musica classica. Il motore audio di questo gioco fu "rubato" da Hi no Tori Hououhen: Gaou no Bouke.

IL gioco ha un comando nascosto per un "Test del suono". È possibile accedervi premendo Su, Su, Giù, Giù, Sinistra, Destra, Sinistra, Destra (la maggior parte del "Codice Konami") nella schermata Play Select.

E' uscita anche una versione per SG-1000 , porting diretto da MSX

Il gioco fu programmato da: Koo Eun Joong (scenario) curò lo scenario anche di Woonsin, l'altro clone di nightmare, Ji Eun Kyung (grafica), Kim Sung Keun (musica), Koo Eun Joong (programmazione) il papà di Brother Adventure, Kim Eul Suk (direzione) che diresse anche Super Boy 1&2.

Questa squadra di programmatori in Zemina veniva definita Magic Tiger.

di **Ermanno Betori**





ELLENICA

di Carlo N. Del Mar Pirazzini

La scena per le console degli anni 80/90 è davvero molto attiva.

Nel corso dei mesi ho assistito a diversi homebrew, giochi fisici e progetti in via di sviluppo per numerose console. Tempo fa su RMW abbiamo parlato anche di NESMaker e come questo stia aiutando gli sviluppatori a creare i propri giochi anche con poche capacità di programmazione.

Recentemente mi sono imbattuto su un progetto tutto italiano, conosciuto attraverso una pagina Facebook che mi ha subito incuriosito.

Sto parlando di Ellenica – Dusk of God, gioco in stile Ultima in “creazione” per SEGA MEGADRIVE.

Sono rimasto a bocca aperta guardando lo sviluppo del gioco e subito ho contattato il “papà” di Ellenica, Sefano Canali.

Dopo una breve presentazione via posta, di comune accordo abbiamo deciso di rendere visibile questo prodotto ai nostri lettori italiani e inglesi e di seguirlo in tutta la sua creazione, supportandolo.

Ma lascio la parola a Stefano in arte Mc Valdemar che ci parlerà dello sviluppo del gioco.

NTH – Parlati un po' di te e di come ti sei approcciato a questo mondo. Cosa fai nella vita e la tua passione dove nasce.

MC Valdemar - Salve a tutti voi, ho 47 anni, lavoro come dirigente del settore vendite in un'azienda che si occupa di servizi Cloud per le agenzie europee.

Videogioco dagli albori (partito con Videopac G7000, toccati tutti gli 8 bit Commodore e mosso al PC. Consoles ho assaggiato diverse offerte: MD, SNES, PSX, DC, XBOX/360, PS2/3/4, GC.

Diciamo che la console di riferimento per me rimane il MD per una questione di come ho vissuto quell'epoca.

Si sognava davanti ai cabinet, si sognava davanti alla tante cartucce disponibili nei negozi e si riusciva a comprare solo 3 o 4 giochi l'anno e finivi per conoscerli a memoria.

NTH - Hai altri progetti in "mente" oltre al bellissimo

Ellenica?

MC Valdemar - Ho sempre toccato lo sviluppo ma di rilevante posso solo ricordare:

- giochi per PC 8086 (CGA/Hercules) pubblicati su PC Games (una rivista dell'epoca)

- seguito in versione testuale della prima avventura grafica di Infocom: Return to Zork. L'ho scritta con autorizzazione di Activision. E' pubblicata su IFDB.

- l'anno scorso ho ripreso in mano il C64 e ho sfornato un gioco SEUCK con aggiunte extra ed ha vinto la SEUCK competition 2020: Pagoda Warriors 2.

- Negli ultimi 5 anni ho raccolto materiale, idee, lavorato su tiles, prototipi per questa mia idea di mix di Rpg/avventura che è Ellenica. Lo sviluppo solo di recente ho deciso di spostarlo su MD ma, con l'aiuto del creatore dell'SGDK, sembra che finora le cose procedano bene nonostante la sfida in termini di memoria.

Ovviamente, come tutti, ho un HD pieno di prototipi morti di giochi pensati, abbozzati e mai finiti.

NTH – Direi che è il momento di parlare di Ellenica. A ruota libera.

MC Valdemar - Ellenica nasce da un mio progetto, pensato per PC, di diversi anni fa su cui ho lavorato direttamente, e mediate freelancers pagati, ma sui cui non sono mai arrivato ad una milestone tangibile perchè tutto il lavoro è stato fatto principalmente nello scrivere storia, pensare al mondo da offrire, lavorare sui tileset, etc... e poco sul codice.

A Febbraio 2020 l'idea di provare a comprimere un RPG concettualmente pensato per PC in una cartuccia da 32Mbit per MegaDrive, questo grazie all'esistenza dell'eccezionale SGKDK di Stephane Dallongeville che semplifica notevolmente la gestione video e audio sul 16-bit SEGA.

LA STORIA

Il gioco è ambientato a Creta nell'anno antecedente l'eplosione di Thera (oggi Santorini), eruzione che porterà, assieme all'invasione dei Micenei, alla scomparsa dell'era





Minoica e la chiusura con i Micenei dell'epoca del Bronzo della storia Greca e l'inizio dei Dark Ages.

L'evento ineluttabile, il legame con la storia "vera", la vicinanza dei concetti delle ere con quelli usati da Ultima, un titolo che è dato da una parola latina a mio avviso era il giusto mix.

In questo anno il giocatore si trova vittima di uno dei terremoti che precedettero l'esplosione. Sopravvissuto al terremoto, scoprirà purtroppo la perdita di una persona amata e di lì inizia l'avventura che lo vedrà inseguire la flebile possibilità mostrata da un sogno, di poter riportare in vita la persona perduta.

Gli eventi che incontrerà nel suo cammino gli consentiranno però di operare azioni e scelte che lo avvicineranno di più agli dei del Partenone che sembrano promettergli questa possibilità, od il credo verso la dea Madre a cui apparteneva la persona perduta.

In una dualità che, nelle intenzioni, un pochino ricorda l'evoluzione degli eventi de Le Nebbie di Avalon.

Ci sono scontri, ci sono 4 colossi mitologici (di cui solo due obbligatori) ma l'accento è più su esplorazione ed investigazione che combattimento.

IL GAMEPLAY

Ellenica l'ho personalmente definito un ARPA, acronimo che indica un Action Role Playing Adventure.

Questo perchè ritengo che gli RPG siano un genere di gioco interessantissimo non avendo, più che altri generi, una ricetta "unica". Sono più come un cocktail dove anche il solo cambiare gli equilibri tra storia, personaggi, battle system, enigmi, libertà di esplorazione genera titoli apparentemente simili ma, giocati, sostanzialmente differenti. Ellenica parte dall'essere assimilabile ad un RPG 2D tile-based, dove l'esplorazione vede una mappa generale (overworld) e delle locazioni accessibili dall'overworld che funzionano da locazioni di dettaglio con una differente scala. Il concetto è molto simile ai primi Ultima (pre Ultima 6) ed abbiamo che l'isola di Creta è l'overworld e le locazioni di dettaglio, che vanno da città a grotte, da cimiteri a navi abbandonate, da dungeon a radure con monumenti particolari, sono in parte disponibili all'inizio ed in parte si sbloccano con la storia, od in seguito ad un dialogo o alla lettura di uno scritto.

Alcune sono "obbligatorie", altre opzionali.

Ogni mappa, sia generale che di dettaglio, è composta

da uno o più screen, questo è stato necessario per poter avere tileset molto differenti e riuscire a far funzionare tutto nelle limitazioni della console.

L'esplorazione avviene in tempo reale e l'interazione è possibile sia con alcuni elementi di fondale, come ad esempio porte o bauli, sia con oggetti che posso essere raccolti ed utilizzati o consumati.

Avvicinandoci ad un nemico, o per aggressione del nemico o per scelta nostra, possiamo iniziare un combattimento che trasforma il gioco in maniera più simile ai vecchi titoli SSI od un round di Advanced Hero Quest.

Appare una griglia ed il combattimento è a turni. Inizia il giocatore muovendosi di un numero di caselle in base alla STAT di velocità, poi può usare PUNTI AZIONE per attaccare, cambiare arma o consumare un oggetto, può decidere se utilizzarne per una postura di difesa e poi passa il turno ai vari nemici.

Ogni attacco considera chiaramente il risultato di un lancio di dadi (compresi tiri critici e fumble), la bravura con la classe di arma usata, ed il posizionamento dei due combattenti con bonus per attacchi laterali o alle spalle. Il danno è misurato tra danno da taglio, perforazione, contundente, da lancio e "magico" e si confronta con tutte queste categorie della difesa del nemico per vedere quanto ne passa.

Non c'è magia nel gioco se non in forma di armi degli dei (utilizzabili) e mostri che hanno capacità magiche.

La gestione di STAT e SKILL, uso degli oggetti, quest e mappa funzionano come in Skyrim. START apre un menu con quattro entrate principali dalle quali si controllano STAT/SKILL (oltre a livelli di esperienza, status del giocatore, fame/sete), OGGETTI per indossare armature/armi, consumare oggetti come cibo/bevande/pozioni, oggetti che possono essere letti, chiavi e oggetti vari (spesso quest item o item utili solo al commercio).

Livellando si guadagnano punti da distribuire sulle SKILL migliorando attacco, difesa, robustezza, fortuna (nei tiri col dado), etc...

Insomma, un gioco parecchio differente dagli oltre 1000 titoli disponibili per MegaDrive perchè "molto da PC".

Per ora le sfide principali sono:

- 1) E' il mio primo titolo per MegaDrive
- 2) 32Mbit (4MB) sono pochissimi considerando che il





titolo ha centinaia di tiles di gioco/animazione. Ed ogni tile di gioco corrisponde a 5x5 tiles del MD (per il fondale) e 8x8 tiles per sprite (standard).

3) Il lavoro procede nel tempo libero che ho, che è difficile da prevedere ed ha alti e bassi. Realisticamente dovrei però avere una DEMO pronta per fine anno dove cercherò di avere la maggior parte del gameplay pronto, così da lasciarmi il 2021 solo per bilanciamento/contenuti.

La DEMO indicativamente conterrà solo il 20% come contenuti ma l'85% in termini di possibilità di gameplay perchè sono quelle le cose che voglio testare.

Mi farebbe poi piacere ricordare la squadra di freelancers la cui professionalità sto utilizzando da tempo:

- Nathan Skaggs sviluppa la grafica su PC, grafica che poi io porto su MD per adattare risoluzione, colori, etc... secondo i limiti del VDP. Più poi mi curo di tutta la grafica aggiuntiva per items, interfaccia, digitalizzazioni, ...
- Al Riad per i disegni a mano che sono poi digitalizzati, compressi in immagini 320x184 perchè locazioni particolari hanno immagini e testo aggiuntivo come fosse una vecchia avventura/Libro Game.
- Walter Torbitto che ha preso la storia e l'ha arricchita delle situazioni che potranno il giocatore nel bivio "intimo" di seguire gli dei del patenone o rimanere legato alla dea madre
- Trevor Lentz per le composizioni musicali della versione PC. Che forse mai saranno rese pubbliche ma, MB permettendo, potrebbero far capolino in parte come digitalizzazioni.
- Paulo H. G. Ferreira che sta curando le musiche della versione MegaDrive.

Attualmente ci sarebbe anche un publisher che si è fatto vivo per la versione fisica ma è una cosa che preferisco valutare solo quando avrò la DEMO consolidata a fine anno. Sia perchè non posso fare commitment (essendo un "hobby") ma anche perchè l'investimento finora è stato tutto mio, specie con i freelancers, e sebbene fosse chiaro da subito che mai avrei potuto recuperare con un titolo MegaDrive, non voglio neanche buttare alle ortiche quanto fatto.

La mia idea, prima di discutere con questo publisher, era di rendere la ROM disponibile su Itch.io ed affiancare una limited di 25/30 copie come release fisica "vecchio stile",

ovvero non solo custodia, cartuccia e manuale, ma anche mappa di cotone, bestiario, player's card e coin di metallo come i vecchi Rpg.

Purtroppo la scena MegaDrive è sì piacevolmente viva ma anche legata soprattutto ad esperienze "mordi e fuggi", quindi titoli più arcade, puzzle games, dove con un paio di ore vedi buona parte di quello che il game può offrire e questo è un ulteriore ostacolo ad un gioco come Ellenica. Ma facendolo per passione e non per guadagno, non avevo alternative.

Una versione PC che offra l'intero gioco così come era pensato potrebbe riprendere ad essere sviluppata a seguito della demo per MegaDrive. Se il feedback mostrerà interesse, affiderò sviluppo ad un paio di persone che conosco per una versione Unity.

Ringraziamo Stefano Canali per la sua disponibilità e vi invitiamo a seguirci per ulteriori sviluppi e una prova su "strada".

Inoltre supportate e seguite i canali dove si parla del gioco ai seguenti indirizzi.

Dove seguire il progetto:

- 1) Twitter: <https://twitter.com/McValdemar>
- 2) Sito: <http://www.segamegadrive.it/ellenica/>
- 3) Itch.io: <https://mcvaldemar.itch.io/>

MC Valdemar – Grazie a voi e a presto!

Parere Personale

Purtroppo non ho potuto testare con mano il gioco, ma dalle foto presenti e dal video visionato sulle pagine di riferimento posso dire che la mia personale ASPETTATIVA è davvero molto alta.

Soprattutto dal video possiamo notare la cura dei dettagli e la fluidità del prodotto anche in questa fase iniziale.

Complimenti a Stefano e al suo gruppo per la passione e l'amore che impiegano nello sviluppo di un prodotto che come RMW seguiremo nei prossimi mesi con molta attenzione.





Aggiornamento di Ellenica...

Stavamo per chiudere il numero quando riceviamo una mail di **Stefano Canali** con un serie di screenshot aggiuntivi del suo gioco.

Tutte le immagini erano inoltre corredate da didascalie che completavano le informazioni che avevamo date nell'articolo dedicato ad Ellenica.

Ovviamente non potevamo esimerci dal pubblicare tutto questo materiale che, se possibile, aiuterà ancora di più a far venire l'acquolina in bocca a noi ed ai nostri lettori. :-)



Figura 1 - il menu' di apertura del gioco con le opzioni che consentiranno di cominciare una partita nuova, proseguire dall'ultimo save game, vedere i controlli (pad 3 o 6 tasti) ed una sintesi della storia.



Figura 3 e 4 - Questi sono due screen di esplorazione dell'overworld. L'esplorazione avviene "a stanze" ma le stanze sono spesso più larghe dello screen quindi scollano nelle 4 direzioni. In alcuni casi ci sono icone che mostrano locazioni di interesse nella stanza che possono essere: città, case single, dungeon, grotte, rovine, pozzi, relitti, etc... che sono esplorabili. Il mare così come le nuvole sono completamente animati.

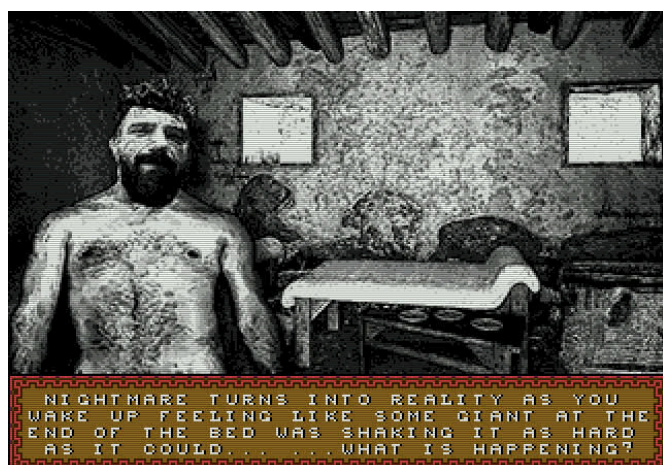


Figura 2 - il gioco ha diverse immagini in B/W che accompagnano alcuni momenti della storia o particolari locazioni scoperte. Questa viene dai primi istanti di gioco.



Figura 5 - Tranne quando si è in combattimento si può accedere ad un menu che, rispetto la mia preview che vi ho mandato, si è arricchito di una tab. In questo caso vediamo la MAP (mappa delle principali locazioni scoperte).





Figura 6 - Le altre aree del menu sono: STATS (vedi stats, skills, status, denaro, punti esperienza), ABILITIES (vedi punti eroe che puoi spendere in talenti attivi, che consumano action points, o passive), ITEMS (vedi, esami, usi, indossi items, leggi libri, etc..), QUESTS (tiene traccia delle quest e dei loro step).



Figura 7,8 - Grandissima parte degli elementi del fondale vegono davvero da version disegnate di elementi dell'era minoica o greca antica. In questo caso, il portale che funge in maniera simile ai portali di Ultima, esiste a Naxos e si chiama Porta del Sole.



Figura 9 - Il combattimento è in modalità a turni, simile ad un round di Advanced Hero Quest. Dalle opzioni di gioco si può decidere il livello di informazioni dei menu di combattimento così ognuno ha il suo livello di informazione; più invasiva o minimale.



Figura 10 - Ed infine uno screen da un primo boss (in un'area di test). Per chi ha memoria dei Libro Game non sarà difficile intuire il tributo alla saga di Sortilegio.



Figura 11 - Un prototipo della cartuccia, dove si intravede la grafica che sarà anche della copertina del gioco, e le principali ispirazioni: I giochi SSI, libri di storia della cultura Minoica, Ray Harryhausen (e film come Argonauti, Scontro di Titani, etc...) e la parte di avventura/disegni dei Libri Game.





STURMTRUPPEN

Editore: Leader distributore
Sviluppatore: Idea
Anno: 1992
Genere: Platform/Shoot'em up
Piattaforma: Amiga

Achtung!!! Achtung!!!

Nel 1992 l'Idea software sviluppo' un gioco basato sulle strisce dei soldati tedeschi di Bonvi (un grandissimo autore purtroppo scomparso) per i nostri Amiga.

L'Idea software a dire il vero aveva una vera passione per sviluppare su licenza ufficiale dalle strisce a fumetti e pochi anni prima aveva sfornato Lupo Alberto e nello stesso periodo stava per fare uscire Cattivik.

Il gioco basato sui soldati più pigri dell'universo a fumetti uscì all'inizio della stagione estiva collezionando pareri non sempre positivi sulle riviste del settore.

Ma prendiamo un pezzo alla volta e cominciamo spiegando cosa affronteremo.

Il giocatore controlla un soldato semplice delle Sturmtruppen, a piedi o alla guida di veicoli contro truppe e mezzi delle Sturmtruppen stesse (cosa piuttosto assurda non avere veri nemici, ndr).

Tutto si muove a scorrimento orizzontale da destra e l'obiettivo è il più classico, arrivare alla fine del percorso. Lo sviluppo del gioco è su sei livelli con diverse ambientazioni. Si va dalla città in guerra, alle rovine, alle caserme, al deserto e persino nei paesaggi alpini e glaciali.

Partiamo come dicevamo con un soldato appiedato nel più classico stile dei platform game. Inizialmente potremo solo dare calci e pugni agli avversari. Da alcuni avversari eliminati potremo raccogliere armi di vario tipo (pistola, fucile e mitra) ma con munizioni illimitate.

Quando ci impadroniamo di un veicolo lungo il percorso, la visuale si allontana ed il gioco diventa il classico sparatutto con munizioni illimitate.

Quando invece recupereremo un fuoristrada o la moto, il gioco ci ricorda il classico Moon Patrol. In ogni fase di gioco differente, anche gli avversari cambieranno e la sfida sarà equilibrata al livello che affronteremo.





Il nostro soldato ha una sola vita, ma può sopportare diversi colpi. La barra d'energia è rappresentata da una sagoma a forma del soldato stesso che si trasforma, ogni volta che verremo colpiti, in un scheletrino. Anche i veicoli hanno un'energia e se distrutti, torneremo appiedati.

Insomma, a guardarla in questo modo il gioco sembra stuzzicante e ricorda alla lontana un Metal Slug alla italiana. Il comparto grafico è molto gradevole e ben realizzato per il tempo, anche se l'Amiga non fu mai sfruttato completamente; alla vista finale il prodotto non è malaccio nemmeno ora.

Meno godibile il reparto audio con effetti sonori quasi inesistenti e una musica ripetitiva che risulterà fastidiosa durante il gioco.

Possiamo quindi paragonarlo a Metal Slug?

Nein! Si lascia giocare ed il gioco è simpatico grazie alla licenza del fumetto del compianto Bonvi, ma è piuttosto corto e in poco tempo vi ritroverete alla

schermata finale.

Il package Amiga del gioco presentava e presenta tuttora due floppy ed un bel manuale in italiano con cinque strisce a fumetti di Bonvi incomplete. Perché incomplete direte voi? Beh, lo scopo del gioco era farvi terminare i livelli per poter visionare il finale di ogni mini comics.

Tirando le somme, possiamo dire che si tratta di un prodotto di nicchia per appassionati e fan del fumetto italiano, realizzato forse troppo velocemente per uscire sul mercato.

Se siete fan di Sturmtruppen vi piacerà giocarci ogni tanto, tutti gli altri lo lasceranno presto sullo scaffale (o nella directory di qualche emulatore).

di Carlo N. Del Mar Pirazzini

GIUDIZIO FINALE

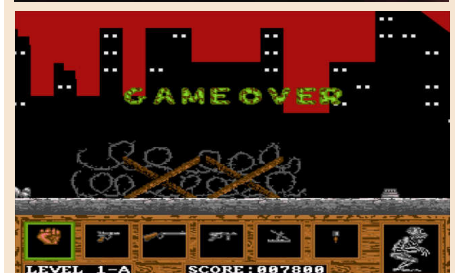


» Giocabilità 75%

E' sicuramente facile da apprendere e direi che il gameplay è anche ben sviluppato così come lo è la grafica. Manca di spessore.

» Longevità 60%

La mancanza di "spessore" che vi dicevo fa sì che questo prodotto si finisca in poco tempo e senza troppa fatica. Se siete fan di Bonvi e delle Sturmtruppen forse lo rigiocherete ogni tanto, in caso contrario cadrà nel terribile oblio videoludico.





WONDER BOY DRAGON'S TRAP

Sviluppatore: Lizardcube
Editore: Dotemu, Nicalis, Headup Games (e il benessere di Sega)
Genere: Platform game a scorrimento
Piattaforme: PC, Nintendo Switch, PS4, XBOX, IOS, Android
Piattaforme 1989: Sega Master System, Game Gear e Pc Engine

“Anche dopo molti anni, le perle brillano come appena raccolte”.

Nel 1989 Sega fa uscire sul mercato il seguito del giocabile e meraviglioso Wonder Boy in Monster Land, gioco da sala e in seguito convertito su qualsiasi sistema casalingo.

Uscì così sul mercato Wonder Boy III the Dragon Trap, ma solo su Sega Master System (e in seguito su PC ENGINE). Fu un successo incredibile tanto da essere riconosciuto uno dei migliori titoli della piccola console SEGA a 8 BIT.

Nel 2018 Lizardcube si tuffa in una nuova avventura quasi improbabile: creare un remake del gioco originale ma farlo con lo spirito dei tempi moderni.

Con il supporto di Ryuichi Nishizawa a garanzia che il lavoro originale venga rispettato, il gruppo Lizardcube ci permette oggi di giocare, anzi rigiocare, un capitolo già meraviglioso della storia videoludica.

Scopriamo assieme se questo nuovo “esperimento” è andato per il verso giusto.

Come dicevamo questo non è il primo episodio della saga di Wonder Boy, ma il terzo (sarebbe meglio dire il quarto, ma vorrei dimenticare il pessimo Wonderboy in Monster Lair). Tutto viene ripreso alla fine del secondo episodio (il già citato Wonderboy in Monster Land) con il nostro piccolo eroe Tom-Tom che si appresta a combattere il terribile Meka Dragon. Una volta sconfitto, il mostro prima di morire lancerà una maledizione che lo trasformerà in una piccola lucertola sputafuoco con l'obbligo di scappare dal castello in rovina e cercare in tutta Monster Land la cura per tornare umano.

Il viaggio si sviluppa in un mondo tutt'altro che lineare. Non avremo

livelli da raggiungere a destra o a sinistra, così che non ci saranno nemmeno suddivisioni in stage o livelli. In base allo scorrere della storia e alle abilità speciali delle trasformazioni e all'esplorazione dovremo scoprire la strada da intraprendere.

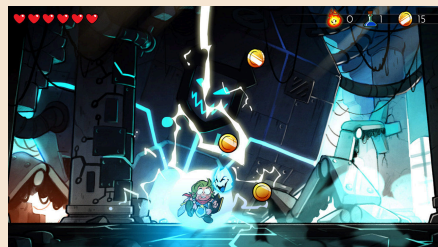
Il gioco si presenta come altri classici dei suoi tempi (Zelda II, Simon's Quest, Metroid) e oggi si potrebbe identificare come un classico Metroidvania.

La novità che colpì al tempo furono



le trasformazioni del protagonista. Ogni volta che sconfiggiamo un boss di fine zona questo ci maledice e ci trasforma in una creatura differente con differenti caratteristiche. E così da lucertola sputafuoco verremo trasformati in topo, in piranha, in leone e in falco. Ognuno di loro presenterà un gameplay differente e caratteristiche uniche che ci apriranno la strada per proseguire nel gioco.

Il topo ad esempio ci consentirà di arrampicarci e camminare sui muri ed intrufolarsi in posti impensabili, il





Piranha ci permetterà di nuotare nei livelli acquatici, il Leone con la sua forza ci permetterà di abbattere muri altrimenti incrollabili e infine il falco ci farà raggiungere alcune zone di Monster Land impossibili se non si vola.

Come dicevamo la non linearità del gioco ci consente di esplorare quasi tutto liberamente, ma se saremo male equipaggiati oppure non adatti a quella zona la nostra esistenza sarà ridotta ai minimi termini.

In questo caso oltre alle varie trasformazioni sarà necessario equipaggiarsi nei numerosi negozi visibili e segreti presenti nel gioco dove potremo acquistare armi, armature e scudi, pozioni di cura e altro.

Ma arriviamo parlando della realizzazione del gioco. La domanda è duplice.

La prima, è un gioco che resiste al tempo? Sono passati 30 anni dalla prima versione potrà reggere il confronto con il giorno d'oggi?

La risposta è SICURAMENTE SÌ! Il gameplay avvincente, l'esplorazione del mondo e il modo fresco di gestire l'azione di gioco rendono tutt'ora DRAGON'S TRAP una piccola perla.

La seconda domanda invece parla direttamente agli sviluppatori. Hanno fatto bene il loro lavoro?

Affermativo! Hanno reso il gioco qualcosa di davvero superlativo dal punto di vista tecnico. La grafica in alta definizione è meravigliosa, ridisegnata con uno stile fiabesco e a cartone animato che delizia gli occhi.

Per non parlare delle animazioni di ogni singolo elemento del gioco. Tutte le trasformazioni del protagonista sono eccezionali (la camminata/saltello del FALCO è bellissima) e tutto si muove perfettamente sia che si guardi su PC sia che lo si faccia su Switch.

L'elaborazione tecnica rimane a livelli altissimi anche sulle musiche che ti immergono nel gioco perfettamente e non sfigurano in nessun modo con le originali.

E proprio parlando degli originali non possiamo non citare la possibilità di switchare durante il gioco dalla versione moderna alla versione old del Master System, musica compresa. Il tutto cliccando un tasto del nostro joypad. Effetto grandioso per tutti noi nostalgici. Il livello di sfida rimane sempre equilibrato e mai troppo punitivo, ma

richiede uno sforzo di attenzione sul come affrontare le sfide e i boss finali. Ogni mostro finale si presenta con le caratteristiche ben delineate e potrà essere affrontato sempre con un uso perfetto di equipaggiamento corretto e colpi precisi.

Creare un remake di un'opera ha delle regole e la principale è quella di preservare lo spirito originale e questo è stato fatto in modo impeccabile.

Il primo impatto è sbalorditivo. Un lavoro che non parla solo del comparto grafico o sonoro, ma anche dalla caratterizzazione del personaggio. Un protagonista che ha subito un trattamento eccellente nelle sue trasformazioni e nelle già citate animazioni.

Concludendo, Wonder Boy Dragon's trap è un prodotto eccellente. Un prodotto che non sente nemmeno un secondo dei 30 anni passati grazie al miglioramento estetico e non tradisce di un grammo lo spirito.



Tre livelli di difficoltà, la possibilità di giocare l'avventura come Wonder Girl, l'aggiunta di un gameplay fresco, fa onore ai ragazzi di Lizardcube che ha dimostrato l'amore verso l'opera originale, riportandola sui nostri schermi per farci tornare bambini, ma anche per fare da anello di congiunzione tra noi sognatori di un tempo e le nuove generazioni.

...Ora vado... Devo tornare a Monster world per sognare ancora... E ancora... E ancora... Buon viaggio Wonder boy.

di Carlo N. Del Mar Pirazzini

GIUDIZIO FINALE

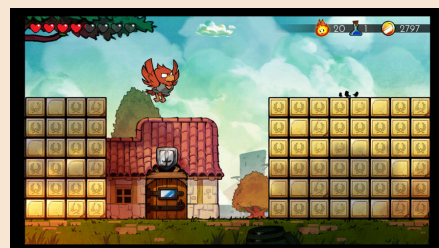


» Giocabilità 95%

Un gameplay superlativo nel 1989 rinfrescato da un supporto tecnico di prima categoria. Ogni personaggio vi fornirà ore di divertimento ed esplorazione. Per non parlare dei livelli di difficoltà selezionabili che alzano la posta in gioco e la possibilità di affrontare l'avventura al femminile. Mai punitivo, severo ma giusto nell'affrontare le sfide.

» Longevità 90%

Non è lunghissimo ma va esplorato con cura e vi piacerà esplorarlo anche finito al primo giro.





MOON CRESTA

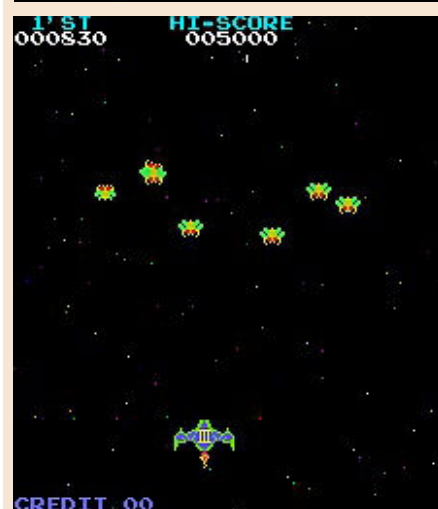
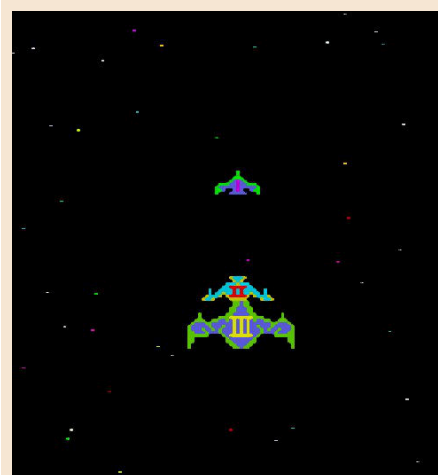
Anno: 1980
Pubblicazione: Nichibutsu
Piattaforme: Arcade, Commodore 64, ZX Spectrum, Amstrad, Wii, Playstation 4
Versione recensita: Arcade
Genere: Shoot'em up

Noi, ragazzini degli anni 70 abbiamo avuto non poche fortune a mio parere. Abbiamo vissuto l'epoca dell'invasione robotica giapponese quando un principe di un altro mondo dentro allo schermo della nostra TV ogni giorno si prodigava a difendere il nostro pianeta dagli invasori di Vega. Quando una sconosciuta e antichissima civiltà del passato sorgeva dalla terra per riconquistare un regno perduto e quando nella nostra giovane mente di adolescenti sognavamo di effettuare agganciamenti cybernetici per diventare possenti robot. Un periodo, quello tra la fine degli anni settanta e la metà degli anni '80 dove, grazie a delle fenomenali macchine che arrivavano dall'America, potevano concretizzare i nostri sogni in fantastiche realtà, sempre dentro lo schermo della TV ma agli ordini di un controller che potevamo prendere nelle nostre mani ! In quegli anni quando il Giappone ha saputo riprendersi la sua rivincita post nucleare conquistando il mondo in maniera pacifica grazie all'invenzione delle console da gioco e quando l'America dopo aver mandato gli uomini nello spazio cominciava a rivendere quelle fantastiche idee tecnologie facendole arrivare nelle nostre case sotto forma di personal computer.

Ancora oggi, e voi lettori lo sapete bene, nonostante la tecnologia dei videogames sia arrivata visivamente quasi a toccare la realtà, una partita ai classici del passato è diventata una piacevole consuetudine così da poter riassaporare ancora una volta l'intera storia videoludica in pochi secondi da quando sullo schermo si controllavano pochi pixel agli

straordinari mondi open-world tridimensionali di oggi. Ma lo stupore associato a quei cabinati di fine anni '70 è ancora vivo dentro la nostra mente quando ogni piccolo passo avanti nel gameplay era letteralmente un grande passo per il media videoludico. Così dopo infinite battaglie con il cabinato di Space Invaders dove le uniche nostre difese erano quelle barriere che secondo dopo secondo si sgretolavano sotto i nostri colpi e sotto i laser degli alieni ecco che Moon Cresta ci regalò un'astronave modulare composta da tre stadi (con il numero chiaramente impresso sulle navette stesse) così come una nuova invasione di navi aliene che pur arrivando in gruppo sapevano muoversi in ogni direzione utilizzando persino semplici e talvolta prevedibili pattern di movimento con l'unico scopo di ridurre tutti i nostri moduli in piccoli rottami dispersi nello spazio.

In Moon Cresta si iniziava con la navetta numero uno, piccole dimensioni e possibilità di sparare un colpo alla volta. Una nave con poca potenza di fuoco ma con la capacità, vista la grandezza, di sgusciare come un'anguilla tra i raggi alieni. Se si veniva colpiti si passava subito al controllo della nave due, la mia preferita, piccola più o meno come la prima ma dotata di un doppio sparo capace di farci credere di essere diventati immortali mentre distruggevano una dopo l'altra le navicelle nemiche. La terza parte della nostra nave, forse la più odiata dai giocatori, era dotata sempre di doppio sparo ma anche di una stazza enorme che purtroppo per noi diventava anche un facile bersaglio per i razzi e le meteoriti nemiche. Ma dove stava





Nichibutsu
Nihon Bussan Co., Ltd.

Main Office: 12-9, 1-chome, Tenjinbashi, Kita-Ku, Osaka
TEL. (06) 353-5211 TELEX 523-6891 NCBCOL J
Branch Office: Tokyo - Hiroshima - Nara - Sapporo - Sendai

GIUDIZIO FINALE

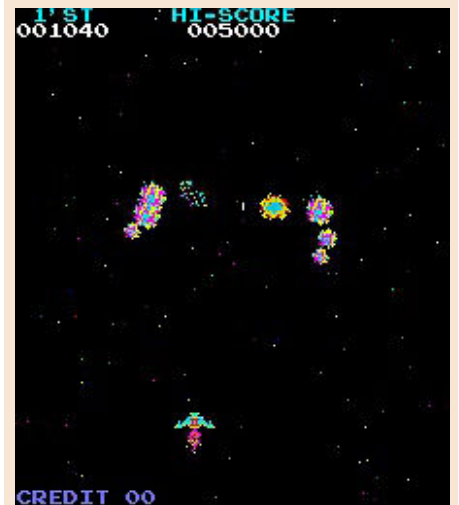


» **Giocabilità 89%**

Frenetico e divertente, combatteremo contro astronavi, meteoriti, razzi e formazioni aliene dalle forme strane e minacciose. Collisioni a volte approssimative specialmente ai livelli di difficoltà più elevata a favore della CPU ovviamente.

» **Longevità 80%**

Moon Cresta è sempre divertente da giocare anche se appartiene al periodo del big bang degli spatutto spaziale (e dell'intera industria videoludica). Le partite come in tutti gli Arcade di quell'epoca sono intense ma veloci ma a pio parere è consigliabile ancora oggi specialmente nella sua versione Arcade emulata dal MAME.



L'innovazione di Moon Cresta oltre a non essere uno spatutto statico (seppur favoloso) come il vecchio Space Invaders? Beh, se si superava un'orda aliena incolumi ci si poteva agganciare con il modulo immediatamente seguente a quello utilizzato con tanto di sequenza alla Jeeg Robot! Per farlo dovevamo far combaciare esattamente i punti di aggancio in modo da unire le due navicelle ottenendone una nuova con potenza di fuoco maggiorata!

Semplice ma geniale considerato che si stava parlando della prima volta in cui si poteva fare una cosa del genere sullo schermo di un cabinato a colori, al costo di 200 lire a partita, mentre gli altri giocavano a carte e fumavano sigarette dentro il bar dove fino a poco

tempo prima si poteva giocare solo a flipper e al biliardino!

Il massimo ovviamente era riuscire a ricomporre la nave con tutti i suoi componenti nell'ordine corretto ottenendo così una potenza di fuoco fuori parametro (gli spari delle tre navicelle si sommano). Ovviamente il senso di onnipotenza non durava molto giusto il tempo di rendersi conto che così la nave oltre che offensivamente efficace era comunque troppo grossa per resistere a lungo agli infidi e sempre più letali attacchi nemici. Il gioco infatti proponeva sempre gli stessi livelli in sequenza che una volta superati si dovevano riaffrontare con un maggiore livello di difficoltà.

di **Flavio Soldani**





RAGING BEAST

Anno: 1985
Sviluppatore: Firebird

Piattaforma: Commodore 64
Genere: Simulazione sportiva

Anche in queste nuove giornate caldissime, post pandemia, la mia attenzione non poteva rimanere ferma e mi è staltato all'occhio uno dei tanti giochi singolari e sottovalutati all'epoca, forse tipici delle cassettoni da edicola. Ed è proprio lì che lo trovai, in una delle tante che avevo, rispondeva al titolo "Olè". La presentazione del gioco aveva la schermata di un torero intento nel suo lavoro, ossia sfidare il toro sventolando il drappo rosso. Non appena partì, giocai per non più di cinque minuti, data la complessità dei comandi e la giocabilità per me quasi inesistente all'epoca.

In questi mesi ho scoperto che il suo nome vero era Raging beast, così ho deciso di rigiocarlo seriamente, imparando bene la meccanica di gioco.

Durante il caricamento mi chiedo se il gioco avrebbe incitato la violenza sugli animali, in questo caso sui tori nelle corride ma così non mi è sembrato perché nel gioco non si nota la minima violenza a parte quando si sbaglia di pochissimo un comando e il toro ti incorna a volontà fino a quando non giungono i barellieri per trasportare il povero torero all'ospedale, con un manifesto sulla piazza con il nome che abbiamo inserito all'inizio del gioco.

Il gioco si presenta con una schermata fissa che rappresenta l'intera arena con il pubblico, rumoroso ad ogni performance, il torero ed il toro in forma un po' piccolina a livello di sprite, ma comunque abbastanza comoda e scorrevole. Come dicevo, la difficoltà sta nel prendere confidenza con i comandi del joystick e ricordarsi le azioni da eseguire; per esempio, destra solleva il drappo, sinistra lo scuote per provocare il toro...

Lo scopo del gioco è quello di fare più punti possibili oltre a vincere la corrida non uccidendo il toro, bensì mettendogli una coroncina rossa lanciata dal pubblico per uscire entrambi dall'arena sani e salvi. Inoltre si può anche calvalcare il toro stile rodeo, molto divertente e non meno facile che nella realtà.

Essendo un gioco mono livello, si potrebbe pensare che possa stancare nel giro di pochi giorni e invece vi assicuro che una volta imparatolo, ci prenderete gusto e vi divertirte, non solo da soli, ma anche sfidando amici e parenti!

Devo ammettere che il biscottone aveva titoli per tutti i gusti ed era in grado di sfornare qualsiasi cosa passasse per la testa degli sviluppatori (...e sta avvenendo tuttora, anche se a ritmo più lento, ovviamente). Il bello delle cassette da edicola era proprio questo, oltre al fatto di trovare giochi a basso prezzo ed in italiano.

All'inizio dell'articolo avevo citato un particolare che mi ha catturato molto, ossia inserire il nome prima di cominciare la sfida... Qualcuno potrebbe obiettare: "Che sarà mai? Molti giochi lo facevano, in particolare quelli sportivi.", ma il nome serve a creare un manifesto come fossimo noi i toreri e vedere il titolo anche in spagnolo (lingua non molto popolare sui giochi del biscottone) ed a seconda del nostro operato, appariva come un titolo di giornale di elogio o provocatorio.

Per chi starà a casa in questi due mesi di vacanze, consiglio di giocarlo, anche con gli amici.

Per gli altri, ricordatevelo quando tornerete, magari sentirete meno la mancanza di spiagge e ombrelloni con plexiglass!

Ci vediamo nel prossimo numero.

di **Daniele Brahimi**



GIUDIZIO FINALE

» Giocabilità 60%

Comandi non facili, ma con la pratica...

» Longevità 70%

Da soli un po' ripetitivo, ma con gli amici...





LUCI DELLA FINANZA

Anno: 2020

Sviluppatore: Marco Vallarino

Piattaforma: Varie

Genere: Avventura Testuale

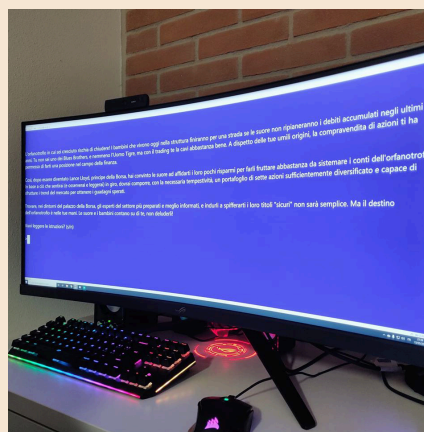
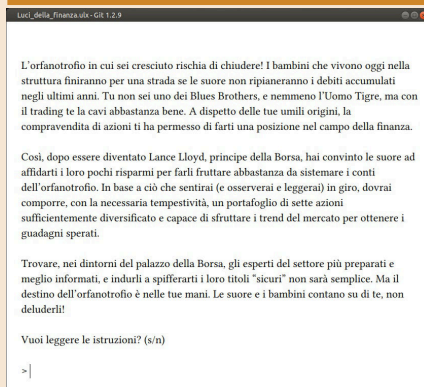


Immagine cortesia di Elena Mascolo

GIUDIZIO FINALE

» Giocabilità 90%

Un buon vocabolario, una trama umoristica ed un basso livello di difficoltà, permettono a chiunque di poter affrontare e divertirsi con questo titolo.

» Longevità ??%

È un'avventura testuale e sapete come funziona in questi casi, una volta finita, è fatta, non la si rigioca più, per cui parlare di longevità non ha senso.

L'estate è infine giunta e, nonostante le attenzioni più che doverose in questi tempi difficili, è tempo di godere di un po' di svago. Avete raggiunto la vostra meta balneare, finito di scavare fossati e stendere filo spinato per difendere i vostri 7 metri quadri di sicurezza anti virus e adesso cercate un po' di ristoro? Abbiamo il gioco che fa per voi.

Approfittando del forzato lockdown causa COVID19 (che Dio o chi per lui lo strafalmini) e cercando di trasformare in opportunità le restrizioni a cui siamo stati soggetti nei mesi passati, il buon Marco Vallarino ha ben pensato di rimettersi alla tastiera e tirare fuori una nuova avventura testuale da regalare a tutti noi appassionati del genere.

La trama è molto semplice: nei panni di Lance Lloyd, un uomo dalle umili origini, oggi diventato un trader di successo, dobbiamo trovare il modo di salvare l'orfanotrofio in cui siamo cresciuti e le suore che ci hanno amorevolmente tirato su. A differenza dei Blues Brothers però, non abbiamo il benché minimo talento nè per la musica nè per il furto ma possiamo sfruttare il nostro eccezionale fiuto per gli affari per giocare in Borsa e generare, con sole 7 azioni, un profitto sufficiente a salvare l'orfanotrofio. Convinte le sorelle ad affidarci i loro risparmi, iniziamo a girare in un mondo di gioco per nulla vasto ma più che sufficiente ad offrire un buon grado di "immersione" nella trama, alla ricerca di indizi che ci permettano di individuare gli affari più lucrosi e concluderli prima che sia troppo tardi. Il tempismo infatti qui è uno degli elementi chiave per vincere e, come nella finanza vera, un buon affare si chiude al momento ed al tempo giusto, pena la diminuzione del gruzzolo guadagnabile.

La meccanica è quella tipica delle cacce al tesoro, presente in molte avventure testuali fin dal loro esordio

(pensiamo ad esempio a Colossal Cave, Zork o Adventureland) anche se qui lo svolgimento è un po' atipico poiché, invece di cercare un oggetto e riportarlo nel luogo destinato ad accoglierlo (la bacheca dei trofei di Zork o l'inventario del giocatore di Adventureland), qui i "tesori" si trovano tutti in un unico posto (la Borsa), bisogna solo saperli riconoscere.

La necessità di accrescere la nostra capacità di selezione sarà la causa del nostro peregrinare per la città: individuare le azioni più lucrative richiede informazioni di prima mano, reperibili girovagando per la città ed interagendo con i personaggi che la popolano. Una volta scoperto qual è il titolo che vale la pena di avere a portafoglio, non ci resta che correre in Borsa prima che altri arrivino alla stessa conclusione e diano il via agli acquisti, facendolo diventare troppo costoso per essere conveniente.

Un gioco molto semplice, dalla durata piuttosto breve e dal livello di difficoltà decisamente basso per gli standard di Vallarino ma ciononostante divertente e perfetto come "gioco da ombrellone". Le locazioni da esplorare non sono molte e gli enigmi mai troppo difficili, con un po' di intuito se ne viene facilmente a capo e gli esperti di AT in qualche ora riusciranno a portare a termine l'avventura. Se è la prima volta che affrontate un gioco del genere, allora "Luci della finanza" è perfetto per introdurvi in questo magico ed affascinante mondo.

Che aspettate dunque? Scaricatelo e giocatelo, lo potete fare anche dal vostro cellulare, basta procurarsi un interprete per le AT Infocom. E giunti alla fine, non dimenticate di inviare una mail con la password che vi verrà rivelata a ovranilla@gmail.com: Vallarino vi inserirà a buon diritto nella Trading Room, la hall of fame di "Luci della finanza"!

di **Giorgio Balestrieri**





WARDNER

Anno: 1987
Sviluppatore: Taito

Piattaforma: Arcade
Genere: Platform

Verso la fine degli anni ottanta il genere platform imperversava nelle sale giochi e cercare di farsi spazio tra mostri sacri come Ghost' n Goblins, Pacland, Wonderboy o Vs. Super Mario Bros non era certo facile.

Ma nel 1987 la Taito pubblica Wardner, titolo prodotto dalla Taoplan che si presenta come un classico platform a scorrimento orizzontale che secondo me meritava maggior fortuna e fama.



Siamo nella sinistra foresta di Wardner quando i giovani e paffuti Pyros e Erika vengono sopresi da un mago inviato dal terribile Wardner (originale che sia il cattivo a dare il titolo al gioco) che con un incantesimo trasforma la giovane in una sfera di cristallo e la porta via con sè.

Ecco che siamo così costretti a compiere un lungo viaggio per cercare di liberarla e sconfiggere nello scontro finale il perfido Wardner.

Nel corso del nostro cammino dobbiamo vagare per boschi, villaggi abbandonati e castelli infestati.

Dal punto di vista del gameplay abbiamo a disposizione due tasti, uno per il salto e l'altro per lo sparo che consiste in un fuoco sacro che lanciamo dal nostro dito. Ma nel corso del gioco possiamo aumentare la nostra potenza di fuoco raccogliendo powerup o item che ci proteggono come un

mantello o piccoli folletti. Tutto ciò è possibile acquistarlo anche in uno shop che possiamo trovare per strada e nel quale possiamo spendere le monete che raccogliamo.

Il nostro eroe deve vedersela con diversi tipi di mostri ma allo stesso tempo deve saltare tra piattaforme ed evitare ogni genere di ostacolo. Gli sprites sono molto colorati e in perfetto stile pixel art che da sempre quel tocco magico ai titoli arcade.

Il ritmo non è frenetico come in molti altri giochi del genere, anzi dobbiamo muoverci con attenzione e valutare bene ogni nostro spostamento o salto.

Il comparto sonoro non grida certo al miracolo ma accompagna bene ogni scena.



A mio avviso il punto debole di questo titolo è che se da una parte trappole e trabocchetti sono da evitare e studiare con attenzione, dall'altra i boss di fine livello, compreso Wardner stesso, non sono del tutto irresistibili.

Il titolo è comunque molto piacevole da giocare e ha avuto un discreto successo che ha portato alla nascita di una conversione per Nintendo e Mega Drive.

di **Querino Ialongo**



GIUDIZIO FINALE

» Giocabilità 80%

Il gameplay di Wardner non presenta nessuna novità per il genere. Un tasto per saltare e uno per sparare, ma il titolo resta lo stesso molto divertente.

» Longevità 80%

Il ritmo non troppo frenetico rispetto agli altri titoli di questo genere potrebbe sembrare una limitazione, in realtà secondo me è stato una scelta accurata che rende questo titolo piacevole ancora oggi dopo più di trenta anni.



La soddisfazione di andare controcorrente

Negli ultimi due anni l'interesse nei confronti del nostro amato hobby del retrocomputing è in forte crescita. I numeri parlano chiaro. Strumenti vetusti ma tuttora utilissimi come i forum, i social network "tradizionali" come FB, Twitter, Instagram, YouTube e quelli più giovani come Discord, TikTok e SnapChat testimoniano una lenta ma costante crescita di appassionati e fan "stanziali" ma anche di fruitori occasionali e nostalgici. Come in qualsiasi attività umana, ci sono tanti tipi di utenti e altrettanti approcci alla materia del retrocomputing, dato che questa passione offre mille aspetti e numerosi punti di attracco per tutti coloro che vogliono farne parte o contribuire con idee e progetti concreti.

Si parte dal collezionismo puro di macchine, periferiche, programmi e gadget, fino ad arrivare a costituire veri e propri musei ("spenti" o funzionanti che siano). Dall'accumulo scriteriato di hardware dal dubbio valore, ai progetti di nuove macchine homebrew sullo stile dei vecchi home computer, Dall'unboxing nudo e crudo, alle sfide di retroprogramming, dal type-in di listati presi da libri e vecchie riviste al refurbishing di sistemi ormai obsoleti. Dalle più nobili iniziative volte a preservare la storia dell'informatica a quelle più leggere che trattano lo streaming di giochi. Innumerevoli altre sono le attività connesse al retrocomputing che noi di RMW cerchiamo di documentare e portare all'attenzione dei lettori ad ogni uscita. Alcune più attraenti di altre, alcune più complesse da affrontare, tutte degne di attenzione, perché, lo sappiamo bene, ogni volta che si parla di microcomputer, home computer, personal computer e console a 8 e 16 bit, le nostre orecchie si drizzano, assetati come siamo di novità moderne e di riscoperte legate al passato.

Di recente ho potuto assistere, con un certo dispiacere, ai diversi riscontri che ottengono le diverse attività divulgative che spesso vengono organizzate nei vari gruppi Facebook sottoforma di meeting online e videoconferenze. Pochi, sparuti spettatori quando si è parlato del funzionamento intimo dei computer sui quali spesso passiamo del tempo a lavorare, sperimentare e giocare. Tanti, tantissimi partecipanti quando si è trattato di descrivere e mostrare all'opera giochi arcade o legati alle maggiori piattaforme a 8-bit. Ovviamente nessuno si aspetta che frotte di utenti si ritrovino desiderosi di sapere tutto su CPU, schemi elettrici, porte logiche e memorie ROM/RAM, data l'intrinseca necessità di un minimo di background da parte degli spettatori per affrontare certi argomenti. Così come non ci stupiamo che, quando invece si parla di raggiungere un high-score a Pit Stop II o come superare il mostro di fine livello di R-Type, anche un piccolo meeting riesca a calamitare l'attenzione di più di cento fan collegati contemporaneamente.

Rispetto a 30 anni fa sono tante e tali le fonti per documentarsi, studiare e comprendere come funzionano davvero le cose all'interno di un computer, persino quando facciamo girare uno dei nostri giochi preferiti, che ci aspetteremmo un maggiore interesse verso discipline innegabilmente più complesse ma certamente in grado di dare grandi soddisfazioni. Ma il "segno dei tempi" non è certo rassicurante in questo senso. Purtroppo non sono molti coloro che vogliono avvicinarsi con passione e dedizione allo studio e alla comprensione dei sistemi che abbiamo amato tantissimo in gioventù.

Dal canto nostro, cercheremo di promuovere sempre tutto ciò che ha scritto sopra "retrocomputing". È la nostra stessa missione, ma è necessario, a nostro avviso, andare un po' controcorrente, anche e soprattutto affinché tutto un bagaglio di nozioni, principi e pratiche complesse (come la costruzione e la riproduzione di circuiti elettronici, la programmazione a basso livello, le tecniche software che riescono a superare i limiti imposti dall'hardware, la comprensione profonda dei meccanismi che regolano il funzionamento delle nostre amate macchine, che siano in versione fisica, emulata o implementata in FPGA), non vada perduto o che resti solo su alcuni oscuri testi del passato. Provate ad entrare in uno di questi argomenti e vedrete che non sono poi così complessi. Siamo ancora in tempo. Prendete carta e penna (o tastiera e wordprocessor) e mandateci le vostre esperienze hardcore o hard-coded!

David La Monaca

Disclaimer

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

RetroMagazine World viene concessa al pubblico con licenza: Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale (CC BY-NC-SA 4.0 INT) <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it>

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

Attribuzione

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

NonCommerciale

Non puoi utilizzare il materiale per scopi commerciali.

StessaLicenza

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

Il licenziante non può revocare questi diritti fintanto che tu rispetti i termini della licenza.

Divieto di restrizioni aggiuntive

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



RetroMagazine World
Anno 4 - Numero 24 - LUGLIO 2020

Direttore Responsabile

Francesco Fiorentini

Vice Direttore

Marco Pistorio

Coordinatore Redazione

David La Monaca

Responsabile Area Web

Giorgio Balestrieri

