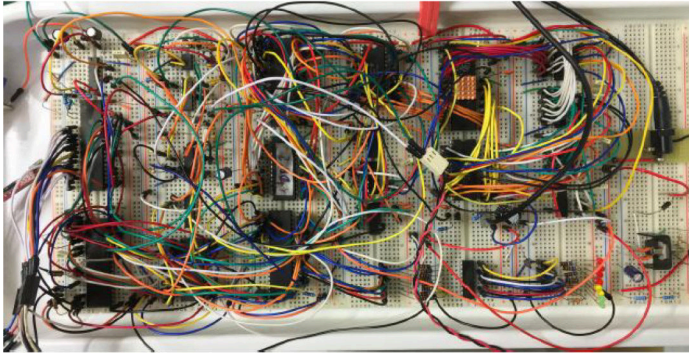




RetroMagazine

semplicemente retro

Numero 21 - Anno 4 - Marzo 2020 - WWW.RETROMAGAZINE.NET - Pubblicazione gratuita



LM80C Color Computer

Un computer autocostruito basato sullo Z80



Katherine Johnson e il suo diritto di contare

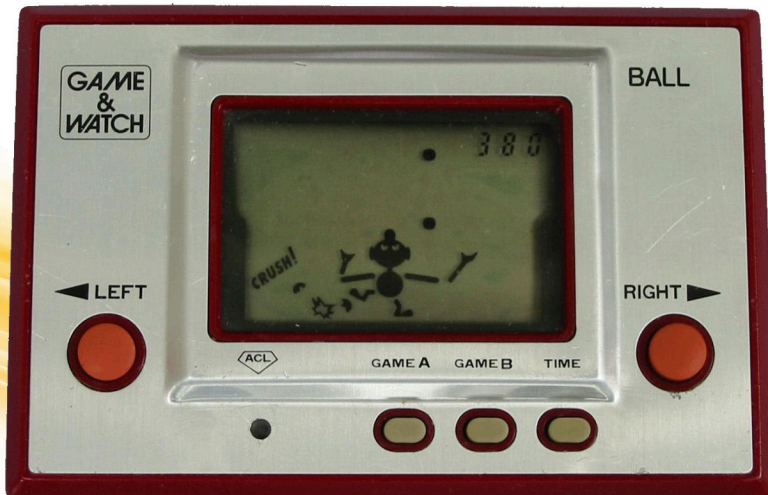


Introduzione alla grafica su
Commodore 128

Retromath:
Dinamica del contagio

Bubble Bobble
La recensione

Giappone 9^a puntata: Mr Game & Watch



La mia banda suona il rock!

Tranquilli, non ci siamo montati la testa con i tantissimi complimenti che riceviamo.

E poi il Festival di Sanremo è passato da un bel pezzo "con i suoi fanti e i suoi Re" (R.Zero).

Mi è capitato di riflettere sulle sensazioni che provo quando scrivo qualcosa per RetroMagazine, quando partecipo alle discussioni nella Redazione su Facebook.

E allora il nostro Gruppo Redazionale non è poi così tanto diverso da una band di musicisti.

Ciascuno contribuisce con le sue abilità e le conoscenze dei propri "strumenti" (C64, Spectrum, console da gioco, emulatori, ecc.).

Insieme ci troviamo in una sala virtuale per unire le nostre forze e alla fine si "incide il disco", nasce un nuovo numero con una bella e curatissima copertina.

Quando il PDF di RetroMagazine è messo in Rete, è come salire sul palco per esibirsi dal vivo.

L'emozione è forte, "l'aria diventa elettrica" (Pooh). Piacerà? Non piacerà?

Posso sembrare ingenuo o sentimentale ma, parlandone nel Gruppo Redazionale, sta uscendo fuori che non è solo la mia impressione personale.

Pieni di passione per le nostre "macchine d'epoca", ci sentiamo un po' artisti.

Non amiamo essere etichettati come quelli "fissati" con i Commodore e i Sinclair. Certamente sono stati i modelli più venduti ed è naturale concedere molte pagine. Ma noi siamo inclusivi e accettiamo volentieri le opere di coloro che abbiano qualcosa di interessante da raccontare su argomenti e piattaforme diverse.

Mi piacerebbe leggere in chiave "Retro" articoli di Elettronica, Robotica, Telematica, Internet, Telefonia, perfino PC compatibili e Mainframe (argomenti che darebbero senz'altro lustro e completezza alla rivista).

La Storia dell'Informatica non è costituita solo dall'avvicinarsi di strumenti di calcolo. La loro realizzazione è il risultato ottenuto da uomini e donne che hanno sviluppato e condiviso idee rivoluzionarie.

Per questa ultima ragione, RetroMagazine farebbe un ulteriore salto qualitativo ed espressivo se in Redazione ci fosse anche il contributo di donne appassionate di Retrocomputing e divulgazione.

Per chi fosse interessato/a a collaborare con la Redazione, può scrivere all'indirizzo:
retromagazine.redazione@gmail.com

RetroMagazine è Rock e i fan aumentano ad ogni pubblicazione nonostante sia solo in forma digitale. Abbiamo notizia che molti sacrificano carta di alta qualità e cartucce di inchiostro per stampare e collezionare le copie appena scaricate. Che siano benedetti!

Alberto Apostolo

SOMMARIO

Introduzione alla grafica su Commodore C128 **PAG. 3**

Fino ad arrivare là dove nessun C64 è mai giunto prima – parte II **PAG. 6**

SKIFF10 e MonTENzuma – due ten-line in Locomotive Basic **PAG. 8**

RetroMath: Dinamica del contagio **PAG. 10**

Sulle orme di Ira Velinsky – parte 1 di 2 **PAG. 13**

LM80C Color Computer-Un computer autocostruito nel 2019 basato sullo Z80 **PAG. 17**

Katherine Johnson e il suo diritto di contare **PAG. 21**

Giappone 9^ puntata: Mr Game & Watch **PAG. 23**

BATTLE CHESS (Amiga) **PAG. 26**

BUBBLE BOBBLE (Varie) **PAG. 27**

ACTRAISER (Super Nintendo) **PAG. 29**

CRUSADER OF CENTY (Sega Megadrive/Sega Genesis) **PAG. 31**

AUTOMANIA (C64 e Spectrum ZX) **PAG. 33**

PSYCHIC 5 (Arcade) **PAG. 34**

Avventure testuali - Intervista a Vincenzo Scarpa **PAG. 35**

RETROEVENTI DEL 2020 **PAG. 40**

Hanno collaborato a realizzare questo numero di RetroMagazine

- | | |
|--------------------------|---|
| • Gianluca Girelli | Pirazzini |
| • Marco Pistorio | • Daniele Brahimi |
| • Francesco Fiorentini | • David La Monaca (Cercamon) |
| • Giuseppe Fedele | • Giorgio Balestrieri |
| • Leonardo Vettori | • Querino Ialongo |
| • Leonardo Miliani | • Supporto grafico: Irene G. Valeri |
| • Alberto Apostolo | • Copertina a cura di Flavio Soldani |
| • Michele Ugolini | |
| • Carlo Nithaiah Del Mar | |





Introduzione alla grafica su Commodore C128

di Gianluca Girelli

Dopo tanti anni, e grazie anche alla passione che trasuda da questa rivista della quale originariamente ero "solo" un lettore, ho finalmente ricominciato a programmare (smanettare??) con i computer a 8-bit.

La mia macchina preferita, senza ovviamente togliere nulla agli altri protagonisti del periodo, era ed è il Commodore C128. Comprato nell'86 perchè al tempo l'Amiga era troppo oneroso per le povere tasche dei miei genitori, questo computer è stato un fedele amico per anni ed anni ed oggi, nonostante il tempo trascorso, è tuttora funzionante e campeggia allegramente sulla scrivania nel mio studio. Al tempo, come già recentemente raccontato su un passato editoriale dall'ottimo Francesco Fiorentini, la documentazione era quasi introvabile e soprattutto per il C128, considerato di "fascia alta" al contrario del "popolare" C64, non c'era nemmeno disponibilità di software, ragion per cui fui "costretto" a crearmi da solo tutto ciò di cui avevo bisogno: word processors, gestori di basi di dati, programmi di disegno 2D esemplifici giochi. Decisamente un'ottima palestra!

Le considerazioni seguenti, derivanti da appunti che ho scritto nell'ultimo periodo dopo tanto tempo di inattività sugli 8-bit, valgono sia per il C128 che per il C64. Il codice riportato è scritto però in BASIC 7.0 (C128), mentre per il C64 bisognerà ricorrere all'Assembly o ad espansioni quali il Simons' BASIC. Inoltre, il codice è volutamente non ottimizzato con lo scopo di renderlo un pochino più leggibile ai neofiti.

Sui computer Commodore a 8 bit la grafica è sostanzialmente di due tipi: "alta risoluzione" (Hi-res, detta anche "standard bitmap mode") e "multicolor" (chiamato anche "multicolor bitmap mode"). A metà degli anni '80 queste modalità sostanzialmente significavano due cose: la prima, risoluzione 320x200 pixel a 2 colori; la seconda, risoluzione 160x200 pixels a 4 colori.

Ritourneremo tra poco sul vero significato della definizione 160x200, ma per iniziare affrontiamo il problema "colore".

Tecnicamente ognuno dei 64.000 pixel dello schermo (320x200=64.000) può essere acceso/spento ed avere un colore proprio, scelto da una palette di 16. In realtà, per quanto riguarda la gestione del colore, lo schermo non è visto come una sequenza di singoli punti ma di griglie di 8x8 pixel. Il perchè di questo approccio, strettamente interconnesso con come vengono gestiti i caratteri sullo schermo di testo ed all'uso di alcuni registri, è al di fuori dell'ambito del presente articolo e sarà magari riaffrontato più avanti. Al momento vi basti sapere che se lo schermo è impostato in modalità Hi-res l'intera griglia può assumere solo un colore (più quello dello sfondo se il bit è spento). Di fatto quindi l'Hi-res è limitato a un colore per il foreground ed uno per il background; se l'impostazione selezionata è invece quella multicolor i pixel nella griglia possono essere di 4 colori diversi

(foreground, multicolor 1, multicolor 2 e background) al "prezzo" di vincolare però i pixel ad una dimensione 2x1. Iniziamo quindi (figura 1) ad esplorare l'hi-res.

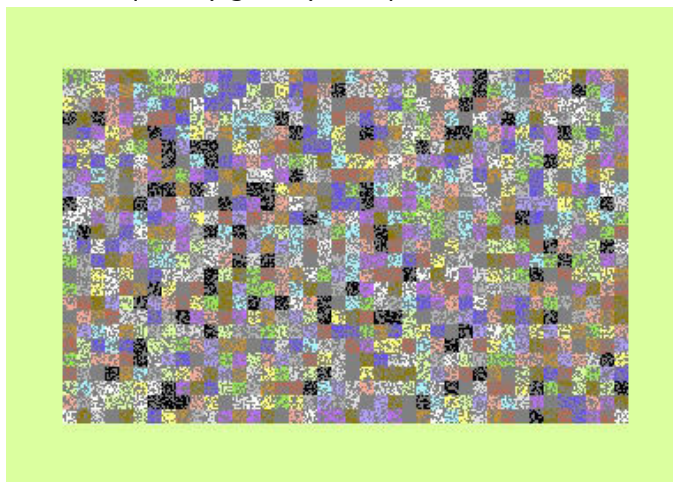


Figura 1

Come si vede, nonostante amalgamati con lo sfondo (di colore grigio chiaro), è chiaramente percepibile la trama di griglie 8x8.

Inoltre, come si potrà apprezzare facendo girare il codice che segue, l'ultimo pixel (in ordine di tempo) disegnato a schermo farà variare il colore dell'intera griglia. Questo comportamento, che di fatto può essere usato per ottenere effetti particolari

(come citato anche nel manuale originale in inglese del C64), è stato lasciato nel codice proprio per porre l'enfasi sulle limitazioni (da un certo punto di vista) o le opportunità che tale modalità comporta/offre.

```
10 GRAPHIC 1,1
20 FOR Y=0 TO 199
30 FOR X=0 TO 319:REM STANDARD BITMAP
31 REM LA RISOLUZIONE ORIZZ E' 320 PUNTI
32 A=INT(RND(1)*2)
34 IF A=1 THEN BEGIN:COLOR 1,RND(1)*16+1
36 DRAW 1,X,Y:BEND
50 NEXT X
60 NEXT Y
```

Il codice riportato, dopo aver selezionato la modalità grafica hi-res ed aver pulito lo schermo (GRAPHIC 1,1) entra in un doppio loop che disegna i singoli pixel per ogni riga e per ogni colonna dello schermo grafico. Visto che in questa modalità sono disponibili solo il colore di sfondo ed un solo colore di primo piano, per ogni iterazione viene generato randomicamente un numero che può avere solo due valori: 0 ed 1. Se 0, nulla viene disegnato a schermo e si passa al pixel successivo; se 1 allora viene estratto a sorte un colore tra i 16 disponibili (per semplicità sfondo incluso) e il pixel viene acceso e colorato. In questa modalità la risoluzione orizzontale è, come detto, di 320 pixel, compresi negli estremi 0 - 319.



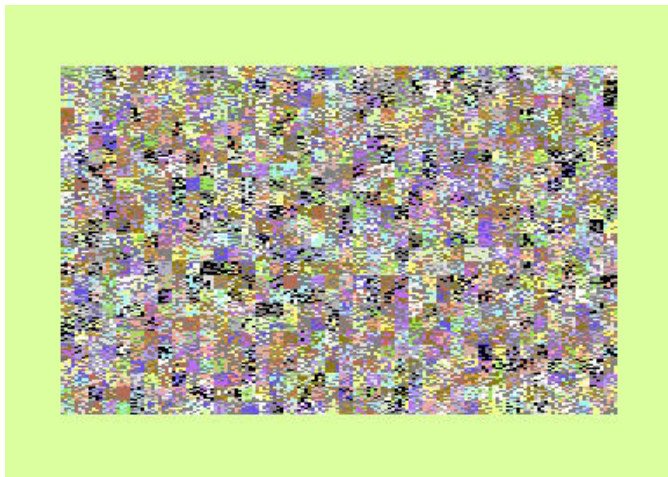


Figura 2

Nell'esempio seguente, invece, viene disegnato a schermo un pattern in modalità multicolor (fig 2).

```

10 GRAPHIC 3,1
20 FOR Y=0 TO 199
30 FOR X=0 TO 159:REM MULTICOLOR
31 REM LA RISOLUZIONE ORIZZ E' 160 PUNTI
32 A=INT(RND(1)*4)
34 IF A=1 THEN BEGIN:COLOR 1,RND(1)*16+1
36 DRAW 1,X,Y:BEND
38 IF A=2 THEN BEGIN:COLOR 2,RND(1)*16+1
40 DRAW 2,X,Y:BEND
42 IF A=3 THEN BEGIN:COLOR 3,RND(1)*16+1
44 DRAW 3,X,Y:BEND
50 NEXT X
60 NEXT Y

```

Il codice è simile al precedente, ma genera una trama di colori ben più omogenea. Anche in questo caso il numero massimo di colori è riferito alla griglia di 8x8 e quindi un eventuale quarto colore estratto randomicamente dall'algoritmo, sempre escludendo quello di sfondo, modificherà quello di alcuni dei pixel adiacenti.

Come si vede dal listato, viene selezionata la modalità grafica multicolor (GRAPHIC 3,1), estratto a sorte un numero tra 0 e 3 e disegnato il relativo pixel. In questo caso: 0=sfondo (background); 1=primo piano (foreground); 2=multicolor 1; 3=multicolor 2.

Come accennato in apertura di articolo, è importantissimo notare che, nonostante le dimensioni fisiche dello schermo (e quindi dell'immagine) rimangano per ovvie ragioni invariate, le dimensioni "logiche" sono ridotte della metà: in altre parole, ad esempio, il centro dello schermo è nella stessa posizione fisica ma corrisponderà alle coordinate 160,100 in modalità hi-res, ed alle coordinate 80,100 nella modalità multicolor. Inoltre, mentre nel primo caso il pixel sarà quadrato di dimensioni 1x1, nel secondo caso avrà dimensioni 2x1 ed occuperà fisicamente 2 set di coordinate: X,Y e X+1,Y. Tale comportamento è messo automaticamente in atto dal chip grafico (VIC II - CMOS8564), ma deve essere riflesso dal codice il cui loop interno non è più fissato tra gli estremi 0 e 319 ma tra 0 e 159. Diversamente, il sistema disegnerà dati in aree non esistenti. Il codice BASIC non dà luogo ad errore e non va in crash se erroneamente si lascia il loop del secondo algoritmo identico a quello del primo, tuttavia

esso impiegherà il doppio del tempo per completare il "render" dell'intero schermo.

Se disegnare uno schermo in stile "Missoni" può indubbiamente avere il suo perché, questo comportamento è un pò più fastidioso nel caso si voglia impostare un lavoro professionale che comporti il tracciamento di linee o altri elementi grafici: l'esistenza della griglia 8x8 comporta infatti che le linee si "sporchino" a vicenda, rendendo il tutto un pò meno piacevole, come si vede nella figura 3 nella quale ho evidenziato alcune delle intersezioni in cui il colore di alcune linee è stato cambiato dalla linea in sovrapposizione. In questo caso la scelta migliore potrebbe essere quella di camuffare tale comportamento mediante l'uso di grafica monocromatica.



Figura 3

Si noti che l'ellisse rossa che circonda i punti da evidenziare è stata aggiunta in seguito con "MS Paint", altrimenti avrebbe ulteriormente sporcato lo schermo rendendo il tutto meno comprensibile.

Sebbene risoluzioni così basse e densità di punti di 2x1 facciano probabilmente inorridire i più giovani, ormai abituati a grafica da 4K o più, è assolutamente possibile ottenere risultati di tutto rilievo anche con soli 160 punti. Riporto di seguito solo un paio di esempi, peraltro nemmeno dei più elaborati.



Figura 4

Il primo (fig 4) è preso dal gioco "The Last V8" (Mastertronic, 1985), forse l'unico gioco di un certo spessore mai arrivato





per il C128 in modalità nativa (e non per il C128 in modalità C64); l'altro (fig 5) è lo screenshot di apertura di "8 Feet Under", espansione del gioco per C64 "Hibernated 1" pubblicato nel 2018 da Pond Software di Stefan Vogt.



Figura 5

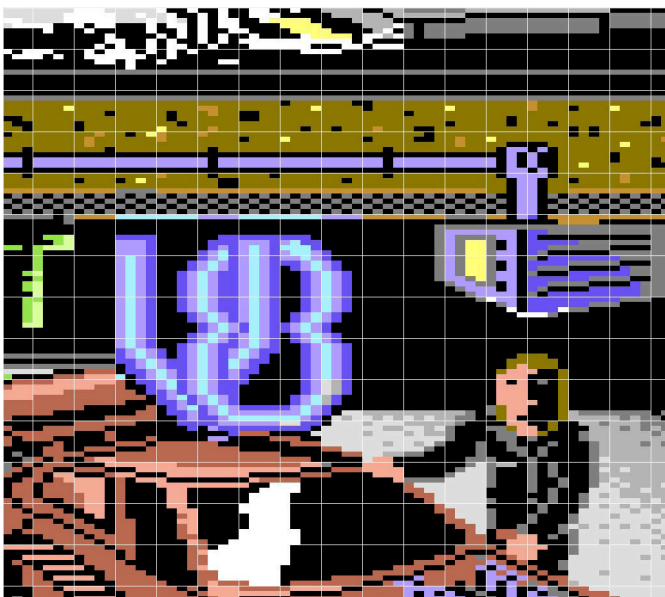


Figura 6



Figura 7

In particolare, per "The Last V8" ho provveduto ad espandere alcuni particolari dopo avervi sovrapposto una griglia. Anche se le foto non rendono molto bene, posso

assicurare che lo screenshot 6 (realizzato in modalità multicolor) evidenzia il rispetto del vincolo di massimo 3 colori (più sfondo) nella griglia di 8x8, mentre la figura 7 pone l'accento sul fatto che i pixel siano tutti di dimensione 2x1.

Nonostante le limitazioni tecniche imposte da un hardware datato spero che queste poche righe siano riuscite a dare l'idea di quante cose sia possibile fare una volta compreso come funziona il sistema.

Ci vuole solo un pò di creatività, tanta passione e tanta buona volontà, ma i risultati possono essere mirabili. La cosa "bella" di un sistema limitato è che ci impone di pensare in modo diverso e più efficiente, non certo facile all'inizio ma sicuramente portatore di enormi soddisfazioni. Concludo quindi l'articolo con un lavoro dell'amico Simone Bevilacqua (fig 8), realizzato tempo fa per uno dei suoi giochi, "MAH".

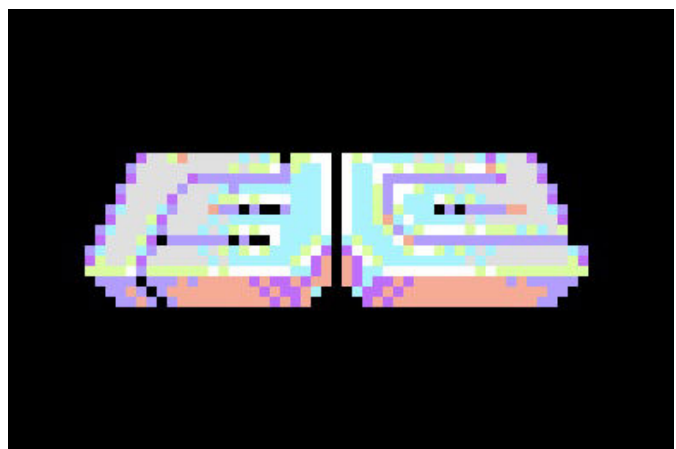


Figura 8

Si tratta del logo del distributore anglosassone RGCD, realizzato con tecnica mista sovrapponendo bitmap di sfondo con serie di sprite hi-res e multicolor, per le quali valgono le stesse considerazioni fatte sino ad adesso (singolo colore per gli sprite hi-res, pixel quadrati; massimo tre colori per gli sprite multicolor, pixel rettangolari). Questo approccio ha reso possibile superare la soglia dei 3 colori (sfondo escluso). Riuscite a contare quanti sono? Il tutorial su come realizzare tale tipo di grafica comparirà su un prossimo numero.

Buon divertimento.

Link utili:

<https://retream.itch.io/mah>

<https://8bitgames.itch.io/hibernated1>





Fino ad arrivare là dove nessun C64 è mai giunto prima - parte II

di Marco Pistorio

Ben ritrovati, amici lettori. Nella prima parte di questo articolo ho parlato dell'obiettivo che intendevo raggiungere, ovvero la riproduzione di una animazione che riproducesse la sigla di apertura di "Star Trek", ed ho cercato anche di descrivere le problematiche legate a questo mio progetto.

In sintesi, la RAM a disposizione del C64 sembra non essere sufficiente a raggiungere questo mio scopo. In realtà sono riuscito ad ottenere il mio risultato, sebbene con una certa approssimazione.

A questo link infatti, a dimostrazione di ciò, potrete scaricare l'animazione che ho realizzato:

<https://csdb.dk/release/?id=185478>

COME SONO ARRIVATO ALLA SOLUZIONE?

Innanzitutto, ho elaborato i circa 120 fotogrammi, a risoluzione 320 x 200 pixel (un solo colore, bianco su sfondo nero), scomponendo ciascun fotogramma in un insieme di 1000 cellette, ciascuna di dimensioni 8x8 pixel.

Successivamente ho inserito in una lista tutte le diverse cellette, escludendo però ogni identica istanza di una celletta già inserita all'intero della lista.

Molte cellette si sono rivelate perfettamente identiche ad altre, e quindi, di fatto, il numero delle cellette all'interno della lista non si rivelato particolarmente eccessivo.

Quindi ho creato una mappa per ciascun fotogramma. Tali mappe mi avrebbero permesso di visualizzare ogni fotogramma indicando ognuna delle 1000 cellette che avrebbero fedelmente riprodotto tutte le scene, una dopo l'altra.

Avrei ottenuto un risultato più o meno analogo comprimendo in qualche modo le bitmap dei fotogrammi compresi all'interno della animazione.

Ho intuito però che, lavorando con i caratteri ridefiniti anzichè con le intere pagine bitmap, il mio percorso sarebbe stato più agevole.

A questo punto ho focalizzato la mia attenzione su questo set di caratteri personalizzati abbastanza ampio (esattamente 4562 caratteri diversi l'uno dall'altro, numero che accenavo già nella prima parte di questo articolo comparso su RetroMagazine 19) che mi permetterebbe di riprodurre TUTTE le scene della animazione.

I 4562 caratteri più le 120 mappe, da 1000 caratteri ciascuna, occupano oltre **121 Kb** di RAM, certamente

molto al di sopra della normale dotazione del Commodore 64.

Il problema delle 120 mappe (che da sole occuperebbero la maggior parte di questo quantitativo di RAM) si potrebbe superare caricando da disco una mappa per per volta, al prezzo però di un certo rallentamento della animazione stessa, a meno di comprimerle in modo opportuno.

Il problema più ostico è invece quello che comporta l'adoperare un set di 4562 caratteri personalizzati.

Il set di caratteri visualizzabile dal Commodore 64 infatti è composto da "soli" 256 caratteri (set di caratteri maiuscolo+minuscolo). Per comporre ogni scena si dovrebbe ottenere un sottoinsieme di 256 caratteri proveniente da quello di 4562 e tale operazione si dovrebbe ripetere poi, ciclicamente, per ciascuna scena.

Come ridurre il set di caratteri personalizzati, da 4562 a soli 256 caratteri, mantenendo una discreta qualità nella riproduzione delle varie scene?

Per risolvere questo problema mi sono affidato ad una metodologia che si fonda su basi matematiche solide, detta "clusterizzazione" o "clustering".

IL CLUSTERING

L'idea, in linea di massima, si può esprimere così:

a) Calcolo la frequenza di ciascun carattere dei 4562 che ho determinato, ovvero ottengo una semplice misura di quante volte ciascun carattere è adoperato.

b) Impiego una funzione che calcoli la differenza tra due caratteri, che mi dia una indicazione di quanto un certo carattere sia diverso (in termini di pixel) rispetto ad un altro, e mi fornisca un punteggio. Un punteggio che sarà 0 quando i due caratteri esaminati saranno identici e che sarà via via tanto maggiore di 0 quanto saranno diversi tra loro i due caratteri in termini di pixel.

Un metodo potrebbe essere l'uso dei **quadrati minimi** (https://en.wikipedia.org/wiki/Least_squares)

c.1) Prendo in considerazione tutte le possibili coppie di caratteri che formano il mio set di caratteri.

c.2) Genero un carattere intermedio tra i due caratteri, dividendo per due o in altra maniera.

c.3) Calcolo l'impatto visivo che otterrei sostituendo il





carattere intermedio alla coppia dei due caratteri che sto elaborando. I termini saranno: il punteggio determinato dalla "differenza" tra i due caratteri, la frequenza del primo carattere e la frequenza del secondo.

c.4) Scelgo la coppia di caratteri con l'impatto visivo più basso e sostituisco i due caratteri con il carattere intermedio, quindi cancello definitivamente la coppia di caratteri di partenza e ricalcolo opportunamente la tabella delle frequenze relative a ciascun carattere del set.

c.5) Torno al punto c.1 finchè non ho ridotto opportunamente il mio set di caratteri.

L'algoritmo è certamente lento, ma alla fine si ottiene il risultato agognato.

E' fondamentale l'impiego di una funzione che permetta un calcolo della differenza tra due caratteri qualsiasi che sia ragionevolmente accurato. Ho adoperato nella mia animazione un metodo di calcolo piuttosto "bruto" ma, tutto sommato, semplice e, a giudicare dal risultato ottenuto, discretamente efficace.

La mia funzione di comparazione si basa semplicemente sull'utilizzo della funzione logica XOR, esaminando i due caratteri comparandoli riga per riga. La funzione XOR restituisce il valore logico "1" in una certa posizione solo se i corrispondenti bit dei due operandi nella medesima posizione sono diversi. Pertanto, due righe da 8 bit esattamente identiche producono un risultato XOR di 0.

Una riga composta da 8 bit "0" (**00000000**), comparata con una riga composta da 8 "1" (**11111111**) produce invece un risultato XOR di **11111111** poichè il primo bit del primo operando è diverso dal primo bit del secondo operando, e così per il secondo bit, per il terzo bit e così via.

Ho definito questa tecnica di comparazione "bruta" perchè non è accuratissima, tuttavia si è rivelata sufficientemente valida ai fini della realizzazione della animazione.

Le mappe si possono immaginare a questo punto semplici insiemi di 1000 numeri compresi tra 0 e 255. Tramite un semplice algoritmo di **compressione RLE** (https://it.wikipedia.org/wiki/Run-length_encoding) sono riuscito a tenere sia le mappe che il set di caratteri personalizzato all'interno della RAM del Commodore 64 "liscio" senza alcun problema.

Nelle prossime puntate di questo articolo esamineremo ancora più nel dettaglio le fasi che ho descritto sin qui, e daremo uno sguardo anche al codice necessario per ottenere questa animazione.

Un saluto a tutti, amici lettori. Alla prossima!

Riferimenti

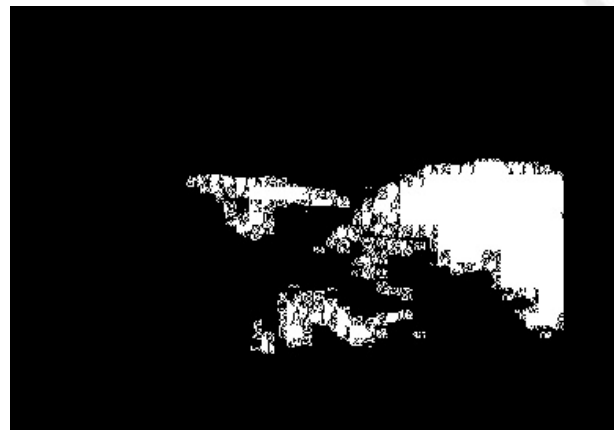
- Link per scaricare l'animazione:
<https://csdb.dk/release/?id=185478>
- Metodo dei quadrati minimi
https://en.wikipedia.org/wiki/Least_squares
- Metodo di codifica RLE
https://it.wikipedia.org/wiki/Run-length_encoding



Fotogramma di esempio 1



Fotogramma di esempio 2



Fotogramma di esempio 3





SKIFF10 e MontENZuma - due ten-line in Locomotive Basic

di Francesco Fiorentini

Il numero scorso ci eravamo lasciati dopo che, avendo partecipato al contest organizzato da RPI - Retro Programming Italia, avevo scritto il mio giochino dedicato a SKIING in Locomotive Basic per Amstrad CPC. Nelle battute finali avevo anche menzionato che ero stato invitato da Fabrizio Caruso a trasformare il mio gioco in un ten-line e partecipare alla competizione organizzata da Homeputerium:

<https://gkanold.wixsite.com/homeputerium/kopie-von-2020>
Prima di cominciare a parlare del gioco (anzi dei giochi...) realizzati in 10 linee di codice, vorrei cogliere l'occasione per ringraziare ufficialmente Fabrizio Caruso che tanto si è prodigato nel gruppo RPI per pubblicizzare il contest e che, non pago di questo, si è anche offerto di aiutare aspiranti ten-liner con suggerimenti ed esempi di codice per riuscire nell'impresa. Personalmente inoltre vorrei ringraziarlo per avermi spronato a partecipare, fino a farmi venire voglia di realizzare anche un secondo gioco.

Ma bando alle chiacchiere ed andiamo dritti al sodo.

Ecco il codice del mio ten-line SKIFF10, realizzato in Locomotive Basic per Amstrad CPC.

Come si evince dalla lunghezza di alcune delle righe del Listato 1, non si tratta di un PUR80 (sorry Fabrizio, ma il tempo è tiranno...), ma di un più comodo PUR120.

Ma cosa significano PUR80 e PUR120?

Le regole del contest prevedono 4 diverse categorie:

PUR80: gioco di 10 righe di lunghezza massima di 80 caratteri. Attenzione solo i comandi built-in del BASIC sono ammessi. Niente scorciatoie in Assembly.

PUR120: gioco di 10 righe di lunghezza massima di 120 caratteri.

EXTREME256: gioco di 10 righe di lunghezza massima di 256 caratteri.

SCHAU: programma di 10 righe di lunghezza massima

di 256 caratteri. Può essere un gioco, un applicazione oppure una demo.

Appurato a questo punto che sono stato sufficientemente bravo (e modesto...) da ridurre il gioco a 10 righe, ma allo stesso tempo pigro nel non comprimerlo entro le 80 colonne, andiamo a vedere in dettaglio il codice.

Una precisazione è d'obbligo prima di cominciare ad analizzare il codice. Il Locomotive Basic, a differenza di altri dialetti Basic (ad esempio il Basic V2 del C64), è sì potente, ma non contempla l'utilizzo di abbreviazioni. L'unica abbreviazione possibile è il ? al posto di PRINT.

Ecco il gioco **SKIFF10** spiegato riga per riga:

riga 1: ridefinizione dei caratteri per rispettivamente bandierina, albero e pietra.

riga 2: inizializzazione delle variabili, definizione del modo grafico e dei colori che verranno usati nel gioco.

riga 3: gestisce la comparsa delle pietre dopo aver percorso almeno 1000 metri.

riga 4: disegna rispettivamente gli alberi e le bandierine. Le bandierine vengono disegnate con una distanza di 15 linee le une dalle altre.

riga 5: è un'aggiunta rispetto alla versione estesa, gestisce i limiti laterali. Tali limiti sono disegnati in modo da simulare un effetto sinusoidale per dare movimento al gioco.

riga 6: gestisce il movimento del nostro sciatore utilizzando i tasti freccia destra e sinistra.

riga 7: gestisce le collisioni tramite la funzione COPYCHR\$ del Locomotive Basic. In pratica valuta il valore della posizione che dovrebbe occupare il nostro sciatore e se si tratta di una roccia, un albero o del limite laterale, salta direttamente alla riga 9, altrimenti il programma continua normalmente.

riga 8: disegna lo sciatore e ritorna alla riga 3 per eseguire il ciclo nuovamente.

```
1SYMBOL 254,64,112,124,126,64,64,64,64:SYMBOL 255,24,60,24,126,24,255,24,24:SYMBOL 253,0,0,0,24,60,124,127,255:RC=0
2K=1:G=0:P=0:D=0:X=18:SY=18:S#=CHR$(248):F#=CHR$(254):T#=CHR$(255):MODE 1:PAPER 0:BORDER 26:INK 0,26:INK 1,0:CLS
3IF P>1000 THEN B=INT(RND(1)*35+1):PEN 1:LOCATE B,25:? CHR$(253)
4B=INT(RND(1)*35+1):PEN 1:LOCATE B,25:? T$:IF D=15 THEN C=INT(RND(1)*(30-5)+5):PEN 3:LOCATE C,25:? F$;"____":F$:D=0
5LOCATE 1+K,25:? CHR$(207):LOCATE 40-K,25:? CHR$(207):LOCATE 40,25:?" ":P=P+10:IF D<8 THEN K=D ELSE K=D-(D-K)-1
6SX = SX + (INKEY(8)>-1)-(INKEY(1)>-1)
7LOCATE SX,1:R#=COPYCHR$(#):IF R#=T# OR R#=CHR$(253) OR R#=CHR$(207) THEN GOTO 9 ELSE IF R#="_" THEN G=6+1
8PEN 1:? S#:D=D+1:GOTO 3
9PAPER 1:PEN 3:LOCATE 1,5:? "MORTO! PERCORSI:";P;"mt. e:";G;"porte ":P=P+G*50:? "PUNTEGGIO: ";P:IF P>RC THEN RC=P
10LOCATE 1,8:? "RECORD:";RC:FOR I=1 TO 3000:NEXT I:GOTO 2
```

Listato 1: SKIFF10 in Locomotive Basic per Amstrad CPC - PUR120





riga 9: richiamata dalla riga 7, ci informa che siamo morti, stampando il messaggio a video insieme al punteggio che viene calcolato sommando i metri percorsi più un bonus di 50 punti per ogni porta attraversata. Se il punteggio è il nuovo valore massimo lo memorizza come record.
 riga 10: mostra il record ottenuto dall'avvia del gioco e dopo un breve lasso di tempo, tramite un ciclo da 1 a 3000, riavvia il gioco partendo dalla seconda riga.



A questo punto avevo preso gusto e mi era venuta voglia di scrivere un altro gioco. Volevo però cambiare sostanzialmente il gameplay per creare qualcosa di diverso e quindi mi sono buttato sull'archeologia. Il gioco in questione si chiama infatti **MontENZuma** ed il nostro eroe, nei panni di un archeologo dilettante, deve raccogliere la chiave che gli consentirà di passare al livello successivo evitando nel frattempo il contatto con i mortali serpenti che piano piano andranno a riempire l'area di gioco sbarrando così la strada verso l'agognato oggetto...

Il concetto è semplice ma il giochino rappresenta qualche insidia in più rispetto al gioco precedente. La prima è quella di dover limitare l'area di movimento del nostro giocatore e la seconda, forse un po' più complessa, è la necessità di ripulire la scia lasciata dallo spostamento del personaggio principale.

Vediamo come il tutto è stato risolto in 10 righe.

riga 1: ridefinisce i caratteri della chiave e dei serpenti. Da notare che per risparmiare caratteri le righe vuote (valore 0) del carattere della chiave CHR\$(254) sono state saltate. Inoltre setta la modalità video ad 80 colonne (MODE 2). Inizializza anche le variabili utilizzate nel programma.

riga 2: pulisce lo schermo e scrive la riga del titolo dove compaiono anche il numero di chiavi raccolte.

riga 3: gestisce l'apparizione random della chiave ed emette un beep tramite il comando ?CHR\$(7)

riga 4: memorizza la posizione precedente dell'omino nelle variabili OX e OY (si veda poi la riga 8) e gestisce il movimento verticale ed orizzontale dello stesso tramite le frecce.

riga 5: calcola randomicamente dove far apparire un serpente e memorizza la posizione in A e B, gestisce inoltre il limite di movimento orizzontale del giocatore

riga 6: disegna il serpente a video (usando la variabili A e B) e gestisce il limite di movimento verticale del giocatore

riga 7: gestisce la collisione dell'omino tramite COPYCHR\$() e nel caso di scontro con un serpente salta la riga 10

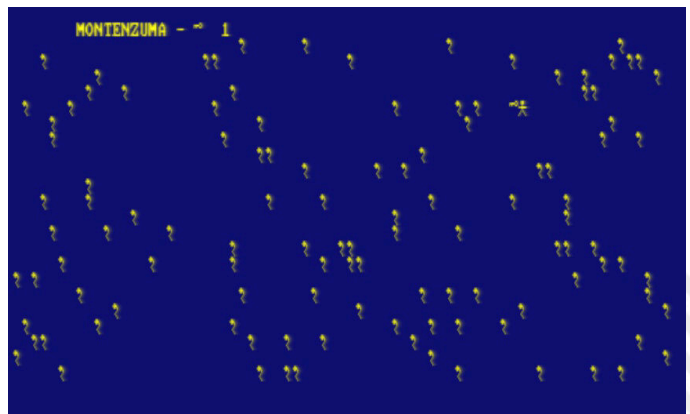
riga 8: disegna l'omino alla posizione X,Y e ripulisce la vecchia posizione dell'omino se differente da quella attuale

riga 9: riavvia il ciclo partendo dalla riga 4 non prima però di aver controllato che la chiave sia stata sovrascritta da un serpente

riga 10: termina il gioco mostrando il messaggio KILLED BY A SNAKE! ed un breve suono

Come vedete scrivere un semplice giochino in BASIC in 10 righe non è poi così difficile. Spero che questo articolo serva di stimolo a tanti di voi per provare a cimentarsi in questa disciplina.

Come sempre se avete bisogno di consigli noi di RM e RPI siamo ovviamente disponibili!



```

1SYMBOL 254,6,249,166:SYMBOL 255,96,240,104,8,16,32,16,8:MODE 2:D=0:K=0:X=9:Y=9
2CLS:LOCATE 10,1:? "MONTENZUMA - "CHR$(254);" ";K:LOCATE X,Y:? CHR$(249):K=K+1
3KX=INT(RND(1)*75+3):KY=INT(RND(1)*22+2):LOCATE KX,KY:? CHR$(254):? CHR$(7)
4OX=X:OY=Y:X=X+(INKEY(8)>-1)-(INKEY(1)>-1):Y=Y+(INKEY(0)>-1)-(INKEY(2)>-1)
5A=INT(RND(1)*75+3):B=INT(RND(1)*22+2):IF X<2 THEN X=2 ELSE IF X>79 THEN X=79
6LOCATE A,B:? CHR$(255):IF Y<2 THEN Y=2 ELSE IF Y>24 THEN Y=24
7LOCATE X,Y:R#=COPYCHR$(#0):IF ASC(R#)>=254 THEN GOTO 10
8LOCATE X,Y:? CHR$(249):IF X<>OX OR Y<>OY THEN LOCATE OX,OY:? " "
9GOTO 4:IF A=KX AND B=KY THEN LOCATE A,B: ? CHR$(254)
10IF R#=CHR$(254) THEN GOTO 2 ELSE ? "KILLED BY A SNAKE!":SOUND 1,300,100,15
  
```

Listato 2: MontENZuma in Locomotive Basic per Amstrad CPC - PUR80





Retromath: Dinamica del contagio

di Giuseppe Fedele

La storia è piena di pestilenze ed epidemie. Un primo caso documentato è la peste di Atene (430 a.C.) che lo storico Tucidide descrisse come una malattia proveniente dall'Etiopia che passò attraverso l'Egitto e la Libia nel mondo greco. La peste nera fu una pandemia importata, forse nel 1346, dal nord della Cina e che si diffuse in fasi successive alla Turchia asiatica ed europea per poi raggiungere la Grecia, l'Egitto e la penisola balcanica. Manzoni dedica alcuni capitoli dei Promessi Sposi e la Storia della Colonna Infame alla peste di Milano del 1630 scatenata dalla discesa delle truppe tedesche che la covavano in forma endemica. L'elenco potrebbe continuare fino ai nostri giorni [1].

Questi tipi di malattie a carattere di pandemia sono fonte di polemica e allarmismo sociale. Quello che a noi interessa è capire come si realizzano le loro previsioni e come modellare il loro comportamento e, perché no, provare a simulare il modello su un retro computer!

Un modello matematico con cui descrivere la diffusione di una epidemia è il modello SIR. Questo tipo di modello può essere fatto risalire al lavoro di Kermack e McKendrick del 1927 [2].

L'acronimo è legato alle classi in cui la popolazione può essere suddivisa durante un'epidemia:

- gli individui sani $S(t)$ suscettibili di essere contagiati;
- gli infetti $I(t)$ che sono a loro volta veicolo dell'infezione;
- i guariti (o deceduti) $R(t)$ detti rimossi.

Se indichiamo con N il numero di individui della popolazione che supponiamo essere costante, ovvero trascuriamo il numero di nascite e morti naturali nel periodo di osservazione dell'epidemia, si avrà ovviamente che la somma delle tre classi dovrà uguagliare il numero di individui:

$$S(t) + I(t) + R(t) = N$$

$S(t)$ non può che avere un andamento decrescente: con passare del tempo i sani vengono contagiati ed aumentano il numero $I(t)$ degli infetti; il numero degli infetti $I(t)$ aumenta in

un primo periodo e ci si augura che diminuisca quando l'epidemia tende a scomparire; $R(t)$ ovvero il numero dei rimossi non può che aumentare

$$0 \leq R(t) \leq N$$

Facciamo alcune ipotesi semplificative:

- non c'è tempo di incubazione, ossia ogni infetto è immediatamente infettivo;
- il contagio avviene per contatto diretto;
- gli incontri tra due qualsiasi individui sono equiprobabili.

Sotto queste ipotesi segue facilmente che il numero di nuovi contagiati e quindi di nuovi infettivi, per unità di tempo, è proporzionale al numero di incontri possibili tra gli individui appartenenti alla classe dei suscettibili e di quelli appartenenti alla classe degli infetti; poiché ogni individuo appartenente alla classe degli infetti può potenzialmente incontrare tutti gli individui suscettibili segue che il numero di contatti è proporzionale al prodotto $S(t)I(t)$; d'altra parte, nella stessa unità di tempo vi saranno degli individui malati e quindi infettivi che guariscono. Abbiamo quindi il seguente modello differenziale:

$$\begin{cases} \frac{dS(t)}{dt} = -\alpha S(t)I(t) \\ \frac{dI(t)}{dt} = \alpha S(t)I(t) - \beta I(t) \\ \frac{dR(t)}{dt} = \beta I(t) \end{cases}$$

dove $\alpha > 0$ è il **tasso di infezione** e $\beta > 0$ il **tasso di guarigione**. Per essere risolto il modello SIR ha bisogno di condizioni iniziali:

- il numero iniziale degli infetti $I(0) = I_0$;
- $R(0) = R_0$ eventuali individui naturalmente immuni dall'infezione considerata;
- $S(0) = S_0 = N - I_0 - R_0$.

Dalla prima equazione si nota che la derivata temporale di $S(t)$ è negativa (come ipotizzato il numero dei suscettibili può solo diminuire); inoltre dalla seconda equazione si vede che se $S(t) \leq \beta/\alpha$, allora anche il numero degli infetti tende a





diminuire: In particolare se il numero iniziale dei suscettibili è inferiore al **valore soglia**, $S_0 \leq \beta/\alpha$ (**indichiamo questo valore con δ**), l'infezione non si propaga ovvero il numero degli infetti non aumenta. Se $S_0 > \delta$ allora $I(t)$ cresce finché $S(t)$ non scende sotto il valore soglia.

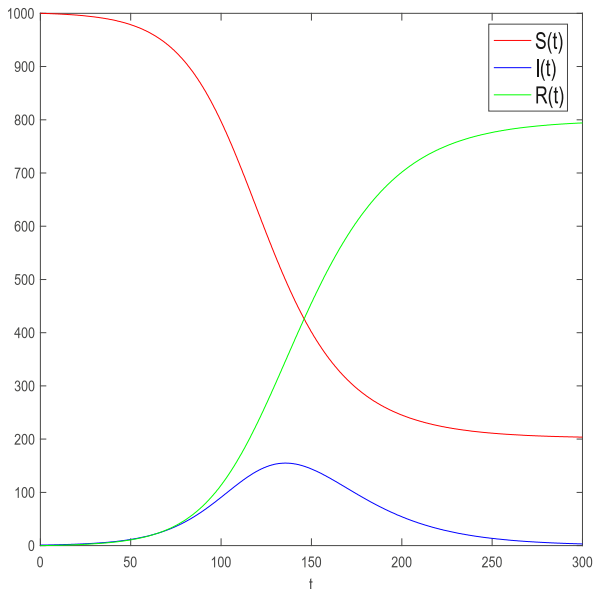


Figura 1. Esempio di dinamica dell'infezione.

Facciamo un esempio. Supponiamo di avere la seguente situazione iniziale:

α	0.0001
β	0.05
S_0	1000
I_0	1
R_0	0

La Figura 1 mostra gli andamenti di $S(t)$, $I(t)$ e $R(t)$. Si nota che al termine dell'epidemia non tutta la popolazione è stata infettata, $S(t)$ si assesta infatti su un valore limite di circa 200 definito **dimensione dell'epidemia**. Il fatto che l'epidemia finisce è testimoniato dal fatto che $I(t)$ tende a zero per tempi sufficientemente lunghi e quindi l'epidemia è destinata ad estinguersi, ma non per mancanza di suscettibili. Notiamo inoltre che in corrispondenza dell'istante in cui $S(t)$ raggiunge il valore soglia $\delta = 500$, il numero di infetti inizia a decrescere fino ad annullarsi.

Contenimento del contagio

Abbiamo visto come la condizione per lo sviluppo dell'epidemia è $S_0 > \delta$. Il parametro α può essere

modificato riducendo l'opportunità di contatto tra gli individui (isolamento, quarantena, ecc.), il parametro β invece può essere modificato dall'assunzione di medicinali. Il numero iniziale dei suscettibili può invece essere ridotto attraverso una campagna di vaccinazione.

Ma quante persone è necessario vaccinare? Supponendo di essere in grado di vaccinare una percentuale η della popolazione, il numero dei suscettibili diventerà allora $(1 - \eta)N$. Affinché l'epidemia non si sviluppi, dovrà essere

$$(1 - \eta)N < \delta$$

da cui si ha che la percentuale di persone da vaccinare è

$$\eta > 1 - \frac{\delta}{N}$$

Gravità del contagio

E' possibile stimare in anticipo quale percentuale della popolazione sarà colpita? Dal modello SIR è possibile esprimere il numero degli infetti in funzione dei suscettibili (vedi riquadro di approfondimento)

$$I(S) = c_0 - S + \delta \log S$$

dove $c_0 = I_0 + S_0 - \delta \log S_0$. Poiché all'inizio si presuppone che $I_0 \ll S_0$, possiamo trascurare il numero iniziale degli infetti e approssimare $c_0 \approx S_0 - \delta \log S_0$.

Quando l'epidemia è terminata allora $I = 0$ e quindi si ha che

$$S_\infty - \delta \log S_\infty = S_0 - \delta \log S_0$$

dove S_∞ indica il numero di suscettibili ad epidemia terminata, da cui è possibile calcolare numericamente il rapporto S_∞/S_0 , ossia la frazione della popolazione che non sarà contagiata.

Il picco dell'epidemia si ha, come già detto, in corrispondenza dell'istante in cui il numero dei suscettibili diventa pari al rapporto β/α , quindi il numero degli infetti sarà

$$I_{max} = I(\delta) = c_0 - \delta + \delta \log \delta$$





Il picco può essere chiaramente diminuito agendo sulla campagna di vaccinazione.

I grafici della Figura 1, in stile retro, possono essere ottenuti su un C128 attraverso il listato basic riportato nel riquadro.

Bibliografia

[1] MCNEILL, William H. La peste nella storia. Epidemie, morbi e contagio dall'antichità all'età contemporanea, trad, it., Torino, 1981.

[2] KERMACK, William Ogilvy; MCKENDRICK, Anderson G. A contribution to the mathematical theory of epidemics. Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character, 1927, 115.772: 700-721.

[3] GAETA, Giuseppe. Modelli matematici in biologia. Springer Science & Business Media, 2007.

Come ricavare il numero di infetti in funzione del numero dei suscettibili

Dividendo la seconda equazione del modello per la prima si ottiene:

$$\frac{dI}{dS} = -1 + \frac{\delta}{S}$$

che può essere riscritta come

$$dI = \left(-1 + \frac{\delta}{S}\right) dS$$

Integrando, si ottiene

$$I(S) - I_0 = S - S_0 + \delta \log \frac{S}{S_0}$$

da cui

$$I(S) = c_0 - S + \delta \log S$$

con

$$c_0 = I_0 + S_0 - \delta \log S_0$$

```

10 rem-----
20 rem Modello SIR
30 rem (c) 2020 Giuseppe Fedele
40 rem-----
120 rem time window
130 T = 300
140 N = 319
150 DT = T/(N-1)
160 :
170 rem tasso di infezione
180 a = 0.0001
190 rem tasso di guarigione
200 b = 0.05
210 rem infetti iniziali
220 I0 = 1
230 rem suscettibili iniziali
240 S0 = 1000
250 rem rimossi iniziali
260 R0 = 0
270 :
280 dim S(N), I(N), R(N)
290 S(0)=S0 : I(0)=I0 : R(0)=R0
300 :
310 for k=0 to n-1
320 : S(k+1) = S(k) + DT*(-a*S(k)*I(k))
330 : I(k+1) = I(k) + DT*(a*S(k)-b)*I(k)
340 : R(k+1) = R(k) + DT*b*I(k)
350 next k
360 :
1000 rem plot
1010 graphic 1,1
1012 char 1,5,1,"dinamica del contagio:"
1013 char 1,5,2,"alpha="+str$(a)+" beta="+str$(b)
1014 char 1,5,3,"S0="+str$(S0)+" I0="+str$(I0)+" R0="+str$(R0)
1015 d=150 : r = 0.1
1016 draw 1,0,d to 319,d
1020 for k=1 to 319
1028 : color 1,3
1035 : draw 1,(k-1)*DT,d-S(k-1)*r
1038 : color 1,7
1042 : draw 1,(k-1)*DT,d-I(k-1)*r
1043 : color 1,6
1044 : draw 1,(k-1)*DT,d-R(k-1)*r
1054 next k
1057 getkey a$
1060 graphic 0

```



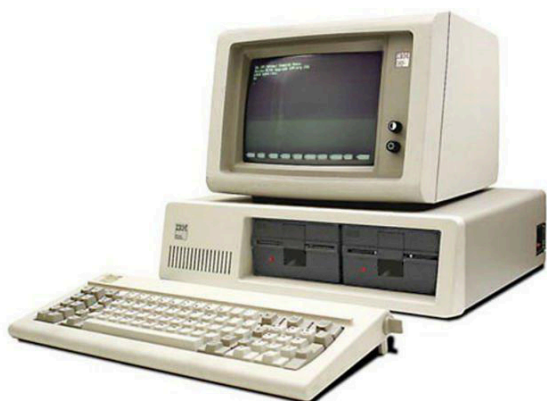


Sulle orme di Ira Velinsky - parte 1 di 2

di Leonardo Vettori

Nei primi anni 80, quando ero bambino, il mio babbo comprava una noiosissima rivista che leggevano solo i "grandi". Quella rivista era l'Espresso.

Tra quegli articoli che parlavano prevalentemente di politica, si trovavano delle pubblicità, e tra quelle pubblicità, ogni tanto, comparivano delle immagini di computer meravigliosi. Erano i CBM della Commodore. Dal mio punto di vista i CBM (Commodore Business Machine) erano certamente superiori agli IBM. (International Business Machine).



Le linee degli IBM mi ricordavano la Fiat 127 bianca del mio babbo parcheggiata nel garage di casa, mentre le curve sinuose dei CBM mi ricordavano quelle meravigliose Porsche 911 straniere che qualche volta sfrecciavano per le stradine del Chianti.



Non c'erano dubbi che le Porsche 911 fossero molto più potenti delle Fiat 127 e di conseguenza, per la proprietà transitiva, i CBM erano più potenti degli IBM. Per anni, come molti, ho pensato che quei computer fossero stati disegnati dalla Porsche, la famosa azienda automobilistica fondata da Ferdinand Porsche, ma non sembra sia così.

Si dice anche che la Commodore abbia usato i servizi dell'azienda Porsche Design, un'azienda collaterale all'azienda automobilistica fondata da Ferry Porsche (figlio del Ferdinand), ma che alla fine non se ne sia fatto nulla del design proposto dall'azienda tedesca e che la Commodore abbia usato il nome Porsche solo per il processo di marketing.

Non sappiamo esattamente come sia andata, ma è cosa certa che il design definitivo fu realizzato da un dipendente della Commodore, Ira Velinsky, secondo il Brevetto U.S. 277.857.

United States Patent [19] [11] Patent Number: **Des. 277,857**
Velinsky [45] Date of Patent: **Mar. 5, 1985**

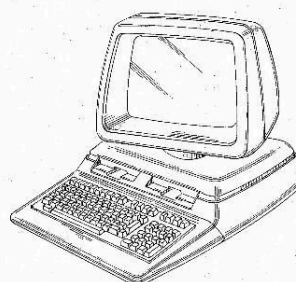
[54] **COMPUTER**
 [75] Inventor: **Ira Velinsky, Plainfield, N.J.**
 [73] Assignee: **Commodore Electronics Limited, Wayne, Pa.**
 [**] Term: **14 Years**
 [21] Appl. No.: **412,310**
 [22] Filed: **Aug. 27, 1982**
 [52] U.S. Cl. **D14/106**
 [58] Field of Search: **D14/100, 101, 105-116; 364/900, 708, 709, 340/365 R, 700, 711**

[56] **References Cited**
 U.S. PATENT DOCUMENTS
 D. 227,899 7/1973 Genaro et al. D14/106

OTHER PUBLICATIONS
 Datamation, 1-1981, p. 168, Informer Computer Terminal; Infosystems, 5-1980, p. 63, Vydex Inc. Terminal #2000; Office Products, 8-1980, p. 33, Cado C.A.T. Computer; Racal-Milgo™ Data Communication Products Catalog 5-1978, p. 16, "40"™ Display System.
Primary Examiner—Susan J. Lucas
Attorney, Agent, or Firm—Davis, Hoxie, Faithfull & Haggood

[57] **CLAIM**
 The ornamental design for a computer, as shown.

DESCRIPTION
 FIG. 1 is a perspective view of a computer showing my new design;
 FIG. 2 is a side elevational view thereof;
 FIG. 3 is a rear elevational view thereof;
 FIG. 4 is an exploded perspective view thereof.





IRA VELINSKY

Per anni ho ammirato il design di alcuni computer degli anni 80 come i CBM, i Commodore Plus4, i Commodore MAX, e con tremendo senso di colpa anche gli ATARI ST (sono un "commodoriano" per natura) senza sapere che dietro quelle meraviglie c'era il lavoro di un uomo: Ira Velinsky.



Di Ira Velinsky si sa molto poco, sia perché era una persona riservata e sia perché ha avuto una vita breve. E' morto nel 2000 a soli 46 anni stroncato da un infarto mentre era in volo per gli Stati Uniti.

Se oggi fosse vivo sarebbe ancora a capo della sua società, la Sozo Design, LLC, a Burlingame in California, che aveva fondato nel 1998, e grazie al ritorno di interesse per il retrocomputer, forse avrebbe rilasciato un'intervista o addirittura scritto un libro per raccontare i bei tempi della Commodore sotto Jack Tramiel e dell'Atari, sempre sotto Jack Tramiel.



Questo non possiamo saperlo, ma di sicuro non sarebbe finito nel "dimenticatoio" di quei tecnici che, con le loro competenze, hanno cambiato il mondo. Di lui, in rete, si trova poco o nulla e non gli viene dedicata nemmeno una pagina Wikipedia in nessuna lingua, nemmeno in inglese. Quelle poche volte che si trova viene accreditato come "CASE DESIGNER".

A pensarci bene anche l'osannato Jonathan Ive dell'Apple è solo un "case designer". I mac, iPhones, Ipad sono solo involucri che contengono l'elettronica. Questi involucri o scatole in inglese si chiamano "case". Senza togliere nulla al geniale Ive, Ira Velinsky ha realizzato i suoi lavori quando i computer non erano ancora considerati oggetti di design da mettere in salotto e mostrare agli amici.

Il suo lavoro è stato riconosciuto e premiato ed ha vinto diversi premi con la Serie Commodore 700, l'Atari Stacy e l'Atari Portfolio

<https://ifworlddesignguide.com/>

Ma qual è stata la ragione che mi ha fatto riscoprire i vecchi lavori di Ira Velinsky? La ragione è stato il nuovo LM80C di Leonardo Miliani.

<https://www.leonardomiliani.com/lm80c/>

Un corpo per l'LM80C

Leonardo Miliani è uno smanettone fiorentino che ho avuto il piacere di conoscere e che sta realizzando il sogno "proibito" di ogni smanettone. Disegnare e costruire il suo computer 8 bit da zero, componente dopo componente, filo dopo filo, codice dopo codice.

In una delle nostre chiacchierate mi disse che aveva iniziato lo sviluppo dell'LM80C perché, dopo aver "programmato" su molti computer progettati da altri, gli era venuta voglia di costruire il suo computer come sarebbe gli piaciuto averlo negli anni '80. Mi ha raccontato che ha "particolarmente sofferto" del fatto che il suo amatissimo Commodore 16 non avesse la gestione degli sprite nel C64 e il TI99.

Quando gli ho chiesto che "corpo" avesse il suo computer LM80C e come se lo immaginava, mi ha detto: "Bella domanda!!! Ma lo sai che non ci ho mai pensato!!! Ho solo pensato a costruire la scheda. Non lo so che forma abbia".

Come un fulmine a ciel sereno mi è tornato in mente cosa disse Doc a Martin in "Ritorno al futuro".

Martin: Hai usato una DeLorean!

Doc: Se dovevo costruire una macchina del tempo dovevo farla con una bella macchina no?

Io quel momento mi sono sentito un "case designer", un Ira Velinsky e gli ho detto "Te lo disegno io" e magari in futuro lo costruiamo per davvero. Non puoi continuare a poggiare la scheda sul vassoio del freezer di tua moglie perché prima o poi glielo dovrai rendere quel vassoio (è tutto vero) La scheda madre la dovrai pure avvitare da qualche parte prima o poi.

Leonardo mi ha detto che gli avrebbe fatto piacere ma...

I computer di Ira Velinsky in ordine cronologico sparso

Ecco una breve descrizione di alcuni dei computer dove





viene accreditato il lavoro di Velinsky e da cui ho intenzione di prendere spunto per il mio progetto.

Dec Rainbow

Il primo computer di cui si hanno notizie fu un "anonimo" DEC Rainbow del 1982.

Il Rainbow ha il "classico" PC case professionale "tastiera-corpo-monitor", ma ai lati del corpo macchina si notano delle "interessanti" alette. Queste alette hanno la funzione di far passare l'aria e raffreddare i componenti del computer ma Velinsky le userà spesso per dare una linea di design ai "suoi" computer.



Serie ATARI ST e XE

Questa scelta di design è "chiarissima" nella serie degli ATARI ST del 1985.

In questo caso le alette di areazione sono ruotate di 45 gradi in modo da essere in sintonia con i tasti funzioni romboidali posizionati sopra la tastiera. Scomodissimi ma bellissimi.

Velinsky posiziona le alette su un buon 30% della superficie frontale del corpo perchè vuole disegnare un computer inconfondibile alla prima occhiata.



Commodore MAX

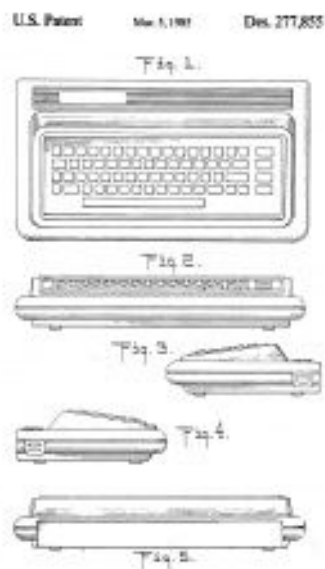
Il "case" del Commodore Max, noto anche come Ultimax o come VC-10, fu progettato da Velinsky nel 1981/1982.

I bordi laterali smussati di raggio costante sono per lo più estetici. La macchina è stata disegnata intorno alla pessima tastiera a membrana dove si vedono già i caratteristici 4 tasti funzione degli 8 bit del Commodore.

La bravura di Velinsky è stata quella di esaltare la tastiera costruendo un secondo ripiano inclinato (proprio per la tastiera) di dimensione diverse da quelle dalla base.

Questo design sembra suggerire "guardate quanto è figa questa tastiera".

In poche parole, Invece di nascondere la esalta.





IL PLUS4 del 1984. Un flop bellissimo!

Dal punto di vista di design è esattamente l'opposto al Commodore MAX, ma il tratto di Velinsky rimane inconfondibile.

Design cattivo, nero con "corpo macchina" spigoloso per uomini duri.

A differenza del Commodore MAX la tastiera è comoda e quindi può rimanere più bassa rispetto alla parte posteriore dove sono alloggiati le alette di areazione verticali.

Il suo aspetto doveva essere completamente diverso dal Commodore 64, e lo è, perchè il Plus4 era nato per essere un computer serio, da lavoro, con 4 programmi da ufficio preinstallati.

I 4 tasti funzione alloggiati al di sopra della tastiera e i 4 tasti per il cursore in basso a destra rompono con la filosofia del VIC20, C64 e C16.

Con il Plus 4 doveva iniziare una nuova era ma in realtà fu solo l'inizio della fine.



Altri Computer

Velinsky è accreditato in molti altri progetti. Sembra abbia lavorato insieme ad altri allo stile "biscottone" del VIC20, C64, C16 e sul bellissimo SX64. Gli viene accreditato anche del lavoro sul C128 ma, siccome era in procinto di lasciare la Commodore per seguire Jack Tramiel in Atari, le informazioni che ci sono in rete non sono attendibili.

In ATARI ha sviluppato lo Stacy e Portfolio e vinto i premi di design.

Ad ogni modo Ira verrà ricordato per aver modificato e industrializzato l'improducibile progetto della Porsche e averci "regalato" la serie di computer CBM II che

sembravano usciti fuori da 2001 Odissea nello Spazio.

UN PRIMO VAPORWARE PER L'LM80C di Leonardo Miliani

Utilizzando il programma 3D Solide Edge, dopo qualche giorno di prove sono arrivato ad avere un primo studio tridimensionale.

Ho seguito le regole che mi ha dato Miliani e alcune regole che mi sono dato da solo e che descriverò nella seconda parte dell'articolo.



Per i miei gusti è ancora troppo Commodore e poco "Miliani" ma è un buon punto di partenza.

E' un buon Vaporware.





LM80C Color Computer

Un computer autocostruito nel 2019 basato sullo Z80

di Leonardo Miliani

Con questo articolo voglio presentarvi un nuovo computer, realizzato nel 2019: si tratta di un sistema innovativo, basato su una CPU ad 8 bit e con ben 80 KB di memoria complessiva! Si tratta dell'**LM80C Color Computer**, che ho... assemblato personalmente io.

Sì, è un computer autocostruito.



Ma non è un computer autocostruito con cui si comunica con la seriale tramite un altro computer oppure che usa piccoli display LCD. Questo computer ha una sua tastiera, si collega ad un comune TV di casa ed opera in tutta indipendenza: si può inserire un programma in BASIC ed eseguirlo subito, come si faceva da ragazzi con i nostri home computer.

Ma ha senso proporre nel 2019 un computer, peraltro, autocostruito, basato sullo Z80, una CPU ad 8 bit sviluppata da Zilog nel 1976? Diciamo di sì, se siete appassionati di retrocomputing e/o nostalgici dei sistemi anni '80. Ed ha ancor più senso perché appunto "fatto in proprio": ma volete mettere il piacere di lavorare e, perché no, giocare su un computer realizzato completamente da voi?

Io questo piacere sono riuscito a togliermelo, realizzando un computer ad 8 bit perfettamente funzionante, completo di tutto: un sistema operativo con un interprete BASIC integrato, una tastiera per l'input, l'immagine su uno schermo, l'output sonoro, la capacità di interfacciarsi con un computer remoto via seriale. E comunque è un sistema perfettamente usabile, con il quale potete realizzare programmi di qualunque tipo: io ci gioco a Dama, a Othello e a Lunar Lander. In figura 1 potete vedere il sistema su basetta:

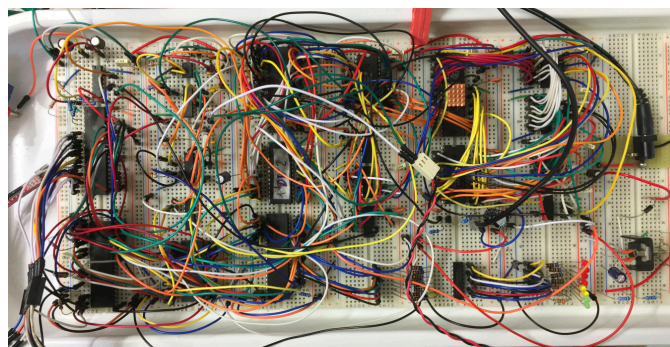


Figura 1: l'LM80C Color Computer

Come ci sono riuscito? Con tanta voglia di fare, testardaggine nel continuare ad andare avanti nonostante i tantissimi ostacoli che ho dovuto affrontare, impegno nello studiare un sacco di documenti tecnici dei più svariati circuiti integrati, l'esser portati per la programmazione (visto

che il sistema operativo è in Assembly). Inoltre il progetto è completamente open-source: chi volesse replicare l'LM80C troverà schemi e sorgenti, utili anche se volesse modificare il progetto per realizzare una versione più adatta alle sue esigenze. Ho deciso di condividere tutte le informazioni perché è grazie al lavoro condiviso da altri che ho potuto costruire l'LM80C ed ho deciso perciò di mettere a disposizione degli altri ciò che ho appreso e realizzato per aiutare altri a replicare ciò che ho fatto io.

Ma partiamo dall'inizio... Se non ricordo male era il Natale del 1984. Sotto l'albero arrivò un COMPUTER! Il C64, direte voi. No, era il Commodore 16. Non storcete il naso, per favore. Il C16 fu una mia parziale scelta: a parte il fatto che mi interessavano i 121 colori, che per l'epoca erano una cosa rarissima per un home computer, il C64 costava troppo per le disponibilità della mia famiglia. E così fu C16. Ma non mi pento di quella scelta, fu un computer con cui mi divertii un sacco e che mi fece entrare nel sangue l'amore per i computer e l'informatica che non mi ha più lasciato per tutti gli anni a venire.... Ma ritorniamo al C16. Tutti i miei amici avevano il C64 oppure l'MSX. Ecco, quando andavo a casa loro a giocare, in quegli spensierati pomeriggi di più di 30 anni fa, mi facevano sempre invidia loro, gli sprite! Quei computer avevano gli sprite! È vero, io avevo 121 colori (ed erano veramente fighi, sullo schermo!) ma non avevo gli sprite. I giochi, poi, dovevano combattere contro il limite di soli 16 KB di RAM. Difatti questi 2 fattori limitarono moltissimo il C16, un computer che nel complesso fu un fragoroso insuccesso.

Anni fa, quando scoprii la piattaforma Arduino, si risvegliò in me l'amore per gli 8 bit. Iniziai a programmare con lo stesso "feeling" che provavo con il C16, ossia gestire bene le risorse e tirar fuori il massimo da quel piccolo microcontrollore. Ma non ero del tutto soddisfatto, mancava la sensazione di avere tra le mani un computer. Così nel 2018 decisi di fare sul serio, ed iniziai a studiare le logiche della serie 74xx ed i più comuni processori ad 8 bit degli anni 80 ancora reperibili. La scelta si strinse intorno a 3 candidati: il MOS 6502, il Motorola 6809 e lo Zilog Z80. Inutile girarci intorno, scelsi quest'ultimo, alla luce di diversi fattori: volevo un processore ad 8 bit di quel periodo ma che fosse facile da reperire ancora in commercio, con un'ampia documentazione e ben supportato in rete; doveva esserci un buon compilatore, possibilmente multipiattaforma, così da poter compilare il software (scritto in Assembly) con facilità. Scelta la CPU gettai le basi per il mio sistema: volevo 32 KB di RAM e 32 KB di ROM, così da non avere problemi di memoria né per i programmi né per il sistema operativo; volevo la grafica sul TV di casa, sicuramente a colori e senza meno con il supporto agli sprite; l'audio doveva essere all'altezza del resto dell'hardware (non volevo quel chip a 2 voci del C16); doveva essere in grado di interfacciarsi facilmente





col mondo esterno. Lo scelta dello Z80 mi spianava la strada perché potevo trarre spunti dallo standard MSX per un paio di soluzioni da adottare per il resto del computer: il chip grafico e quello audio di questi sistemi sono ancora reperibili su eBay a pochi euro, e sono facilmente interfacciabili allo Z80, e difatti optai per il TMS9918A e l'AY-3-8910. Anche per il software avevo una soluzione già pronta: agli inizi degli anni '80 fu pubblicato il sorgente di un interprete BASIC per lo Z80 completo e commentato che nel corso degli anni è stato poi riscritto per i compilatori moderni. Ho dovuto soltanto (si fa per dire...) adattarlo al mio hardware e, quando tutto funzionava, ho iniziato a espanderlo con nuovi comandi e funzionalità. Adesso il mio computer opera in tutto e per tutto come un home computer degli anni '80, con una tastiera di un vecchio C16 (tanto per cambiare...) gestita direttamente dalla macchina, un chip grafico che genera un segnale video che mando direttamente al mio televisore, su cui riproduco anche l'audio generato dal computer, e l'input/output da seriale per comunicare con il mio computer. Arrivare a tutto questo non è stato facile, ma neanche impossibile.

Quindi iniziamo addentrandoci un po' di più nell'analisi tecnica del sistema e dei circuiti integrati utilizzati: questo studio ci permetterà non solo di comprendere meglio come funziona l'8 bit che abbiamo in casa ma anche di iniziare a pensare più in "grande", ossia gettare le basi per lo sviluppo di un proprio sistema che, a seconda della voglia e delle conoscenze, potrà arrivare alla realizzazione finale di un computer completo.

Prima di partire a incastrare componenti su una basetta oppure a tracciare schemi con KiCad è il caso di studiare approfonditamente la struttura di un comune computer per capire come esso è composto e come opera.

In figura 2 potete vedere lo schema a blocchi di un computer:

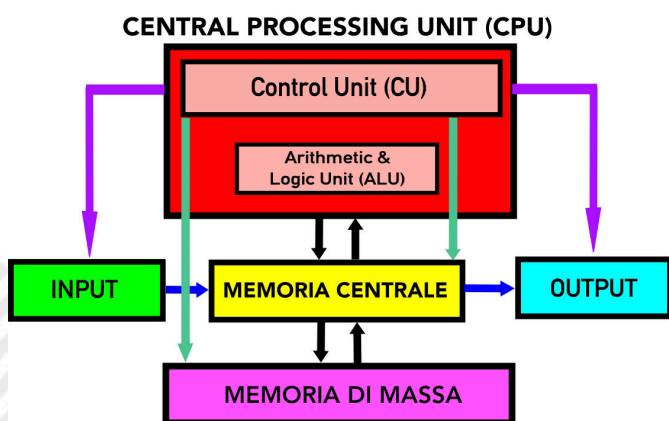


Figura 2: Diagramma a blocchi di un computer

Esistono 4 blocchi principali ed 1 secondario. In ordine, i blocchi principali sono: la CPU, l'INPUT, l'OUTPUT e la MEMORIA CENTRALE. La MEMORIA DI MASSA è un blocco secondario perché non è di vitale importanza per il funzionamento del sistema e può non essere presente.

Vediamo nel dettaglio il funzionamento di ogni blocco:

- **INPUT**: blocco preposto alla gestione dei dati in ingresso. A questo blocco appartengono unità di input come ad esempio un pulsante o un interruttore. La tastiera (una

matrice di pulsanti) è la più comune unità di input.

- **OUTPUT**: blocco preposto alla gestione dei dati in uscita dal sistema. Comuni unità di output appartenenti a questo blocco sono la stampante e lo schermo.

- **MEMORIA CENTRALE**: questo blocco contiene la memoria del computer. Può essere di tipo a sola lettura (ROM), di tipo a lettura/scrittura (RAM), oppure di entrambi i tipi (caso più comune).

- **CPU**: è il "cuore" del computer. La CPU (Central Processing Unit) è l'unità centrale di elaborazione dei dati. E' composta a sua volta dalla CONTROL UNIT (CU), o Unità di Controllo, e dall'ARITHMETIC AND LOGIC UNIT (ALU), o Unità Aritmetico-Logica. La CPU è l'unità preposta a scambiare i dati: essa gestisce i dati in input, elabora le informazioni dialogando con la memoria centrale per recuperare le istruzioni del programma e/o per salvare dati intermedi dell'elaborazione, e invia i dati alle periferiche di output in modo che l'utente possa recuperare i risultati dell'elaborazione.

- **MEMORIA DI MASSA**: se presente un'unità di memorizzazione esterna al computer (ad esempio un floppy disk o un disco rigido), la CPU gestisce anche la lettura e la scrittura di dati verso questa periferica per un'archiviazione e conservazione esterna dei dati.

Come potete vedere un computer è sostanzialmente composto da pochi blocchi funzionali. Il problema non è tanto individuare i singoli blocchi: la "CPU" è il nostro processore Z80, ad esempio, così come l'OUTPUT è, ad esempio, il chip video. Il problema è far funzionare insieme tutti questi componenti differenti. Servono infatti diversi circuiti integrati di collegamento (quelli che gli inglesi chiamano "glue logic", ossia logica collante) e per far funzionare tutti questi integrati serve un minimo di conoscenze di elettronica per risolvere alcune problematiche che possono presentarsi. Ma niente di tutto questo è insormontabile e con un buon manuale che ci spiega i rudimenti di elettronica (cos'è e come opera un resistore, a che serve un condensatore, ecc...) e con un po' di datasheet degli integrati della serie 74xx riusciremo nel nostro scopo.

Partiamo intanto dalle caratteristiche tecniche dell'LM80C. Chi ha più la passione per l'hardware che per il software vedrà che il computer è molto simile agli MSX 1, e difatti molte scelte progettuali sono state mutuare da quei sistemi. Nota: tutti i componenti sono facilmente reperibili in commercio, sia come parti usate (su siti di aste online) sia come componenti nuove, il loro costo di acquisto è modesto e con una spesa di qualche decina di euro è possibile acquistare tutto l'occorrente.

CPU: Zilog Z80 a 3,68 MHz

RAM: 32 KB

ROM: 32 KB

VRAM: 16 KB dedicati esclusivamente al chip video

BASIC: integrato nella ROM, derivato dal NASCOM Basic, a sua volta derivato dal Microsoft BASIC 4.7

Video: Texas Instruments TMS9918A, capace di generare un'immagine a 256x192 pixel con 15 colori (più trasparenza) con fino a 32 sprite (da 8x8 a 32x32 pixel)

Audio: Yamaha YM2149F (identico al General Instruments AY-3-8910), capace di 3 voci indipendenti, 8 ottave, gestione dell'inviluppo, rumore bianco, e dotato di 2 porte





da 8 bit di input/output

Tastiera: 66 tasti (per la cronaca, ho usato una tastiera di un vecchio C16...), con tasti cursore indipendenti e tasti funzione

I/O: Z80 SIO capace di gestire 2 porte seriali con velocità fino a 57.600 bps; Z80 PIO dotato di 2 porte parallele ad 8 bit

Varie: Z80 CTC, timer/counter usato per generare il clock seriale nonché un sistema di tick di sistema con cui incremento un orologio interno ed eseguo varie operazioni a intervalli predefiniti

Partiamo dal cuore del sistema che, come abbiamo visto più sopra, è la CPU. Come detto abbiamo scelto lo Zilog Z80, messo in vendita nel 1976. Il modello che abbiamo usato è una versione CMOS con clock massimo di 6 MHz, abbondantemente oltre la frequenza operativa del nostro sistema. In figura 3 potete vedere la piedinatura del processore, con i pin suddivisi per funzione.

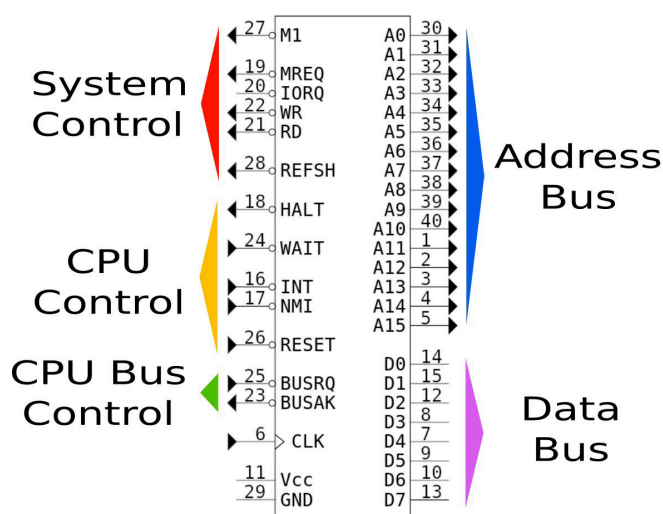


Figura 3: piedinatura dello Zilog Z80

Dall'immagine si nota che il bus indirizzi è di 16 bit, quindi la massima memoria indirizzabile direttamente è 216, ossia 64 KB. Parimenti, il bus dati è ad 8 bit, così come l'architettura interna. Per funzionare lo Z80 ha bisogno di un segnale di clock a singola fase (una sola linea con un segnale ad onda quadra) e di una singola alimentazione a +5 Volt, oltre ad una memoria da cui leggere il codice da eseguire una volta che ha ricevuto l'alimentazione. A questo va aggiunto un circuito con il quale dare il segnale di reset al processore, che serve a reimpostare tutti i suoi registri interni come al primo avvio. Si notano poi diversi altri pin di cui per ora non parleremo e che affronteremo più avanti.

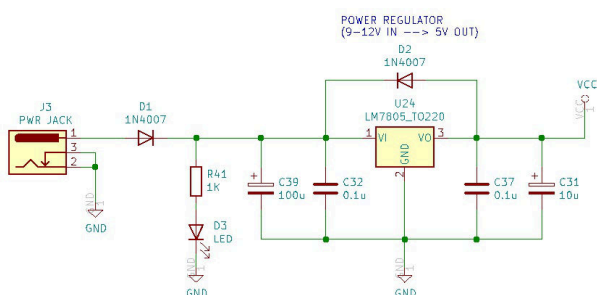


Figura 4: circuito di alimentazione con LM7805

Iniziamo dal circuito di alimentazione, visibile in figura 4: si tratta di un circuito basato sul classico regolatore di tensione lineare LM7805, capace di fornire una tensione di 5 Volt con 1 A di corrente massima in output e che può regolare una tensione in ingresso variabile in un intervallo abbastanza ampio: a noi basta sapere che qualsiasi alimentatore da 9 oppure 12 Volt in corrente continua va più che bene. Consiglio di dotare il regolatore di un'aletta di raffreddamento, soprattutto se pensate di utilizzare in ingresso una tensione di 12 V: più è grande la differenza fra la tensione in ingresso e quella di uscita e maggiore è il calore che esso produce. All'LM7805 potete sostituire anche un LM317: si tratta di un altro regolatore molto robusto la cui tensione di uscita non è preimpostata come nel caso del 7805 ma va selezionata usando un paio di resistori esterni. In figura ci sono 2 diodi: D1, subito dopo il jack, e D2, che collega il pin di uscita e quello di ingresso del regolatore. Il diodo D1 serve come protezione dell'intero circuito, evitando che se per sbaglio si invertono i poli del jack la corrente elettrica non scorra in senso inverso, danneggiando tutti i componenti. Il diodo D2 protegge invece l'uscita dell'LM7805 dai ritorni di tensione dei condensatori posti a valle nel momento in cui si toglie l'alimentazione: i condensatori sono usati come "deposito" di energia per bilanciare le piccole fluttuazioni nell'alimentazione. Nel momento in cui si toglie l'alimentazione essi si scaricano nel circuito: il diodo D2 serve ad evitare che la corrente entri dal pin da cui di solito esce ed arrivi all'interno del regolatore percorrendo vie in "contromano", danneggiando il regolatore stesso, ma passi attraverso di esso verso l'ingresso del regolatore. Nel circuito è presente un LED sull'ingresso per vedere quando il circuito è in tensione.

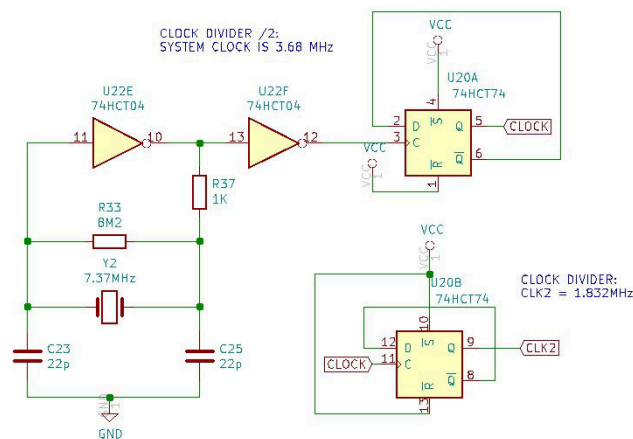


Figura 5: generatore del clock di sistema

Sistemata l'alimentazione, pensiamo al circuito di generazione del clock di sistema. Per il clock ho optato per un quarzo da 7,37 MHz il cui segnale è stato diviso per 2 per ottenere una frequenza operativa di 3,68 MHz. Questo valore è cercato perché è il doppio della frequenza normalmente usata per generare il clock seriale, che è 1,83 MHz. Sui sistemi MSX 1 e su molti sistemi che usavano il chip grafico TMS9918A il clock di sistema era di 3,58 MHz: questo valore non era scelto a caso ma era il risultato di una precisa scelta progettuale atta a contenere i costi. Il TMS9918A funziona con un proprio clock di 10,74 MHz, che è il triplo del segnale di "color burst" di 3,58 MHz usato per generare il segnale video portante del colore.





Il TMS9918A presenta questo clock anche su un suo piedino perciò per risparmiare un altro quarzo molti progettisti usavano questo segnale come clock di sistema. Non avendo necessità di contenere i costi abbiamo optato per una soluzione diversa che ci aiuta invece in altri punti del computer. Il circuito usato per generare il clock di sistema ed il clock seriale è visibile in figura 5. Qui ho usato 2 circuiti integrati addizionali, un 74HCT04, che è un Hex inverter, ossia un circuito che inverte il segnale di ingresso (se è "basso" lo rende "alto" e viceversa), e da un 74HCT74, che è un doppio flip-flop. L'invertitore serve per stabilizzare i livelli del segnale generato dal quarzo: potendo subire leggere fluttuazioni, l'onda generata potrebbe avere dei picchi non chiaramente riconoscibili dalla CPU o da qualche altro integrato perché troppo lontani dalla soglia minima richiesta: con l'inverter ottengo invece un livello stabile e sempre uguale. Il flip-flop è composto da 2 circuiti identici e separati. Ogni parte è stata configurata in modo che operi come divisore per 2 del segnale di ingresso: vale a dire che lo schema che vedete dimezza la frequenza del segnale in ingresso. Nella parte in alto la frequenza del quarzo è stata portata da 7,37 a 3,68 MHz, che è il clock di sistema, usato dalla CPU e dai chip periferici della famiglia Z80 che ho usato (PIO, SIO e CTC) mentre la parte inferiore dimezza ulteriormente questo valore portandolo a 1,83 MHz, frequenza come detto usata per generare il segnale seriale. Non solo, questo valore è usato anche clock del chip audio, dato che questo accetta un clock massimo di 2 MHz.

nell'intero circuito. Trascorso tale tempo il segnale di reset viene disattivato e il sistema può avviarsi normalmente.

Se osservate il circuito, vedrete un altro inverter presente dopo l'NE555: questo si rende necessario perché il timer si attiva con un segnale alto e poi si disattiva portando il segnale ad un livello basso. Ma il reset degli integrati è riconosciuto quando il segnale su questi pin è a livello basso: in questo modo l'NE555 terrebbe i chip sempre sotto reset. Usando l'inverter invertiamo i livelli del segnale in modo che dopo l'avvio il segnale sia basso e che, trascorso il tempo preimpostato, sia portato ad un livello alto, permettendo così l'inizializzazione del sistema. A valle dell'inverter c'è un transistor usato per accendere un LED quando il segnale di reset è disattivato: in questo modo possiamo vedere quando il computer è attivo.

Bene, per questo primo articolo direi che possa bastare. Di nozioni e di schemi da studiare e approfondire ce ne sono in abbondanza. Ci vediamo la prossima volta con lo studio di come la CPU si interfaccia alle memorie ed ai chip periferici, andando a vedere i segnali ed i componenti che servono a dialogare con le memorie (e selezionare la ROM oppure la RAM) e con i dispositivi di I/O.

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>

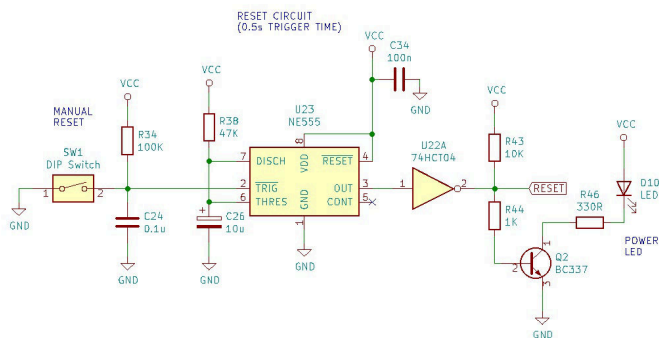


Figura 6: circuito di reset con NE555

In figura 6 vediamo invece il circuito di reset. Diversamente da quanto si potrebbe pensare, anche un buon circuito di reset è un componente fondamentale di un computer. Oltre ad inviare il noto segnale che ripristina sia il processore principale che tutti gli integrati periferici al loro stato iniziale, il circuito di reset entra in gioco anche nel momento in cui si fornisce alimentazione all'intero sistema: affinché gli integrati operino in modo corretto hanno bisogno di una tensione di alimentazione stabile e prossima ai 5 Volt. Durante la prima accensione la tensione non è disponibile subito a questo livello ma sale da zero: anche se in modo rapido, sussiste un certo lasso di tempo durante il quale la tensione è sotto la soglia minima. Questo valore potrebbe portare il processore o uno dei chip periferici a funzionare in modo anomalo, costringendo ad un nuovo reset per risolvere il problema. Il circuito di reset illustrato in figura 6 assolve a questo scopo: grazie ad un NE555, un vecchissimo ma noto integrato temporizzatore, la linea di reset viene tenuta a livello alto per circa mezzo secondo, un tempo più che sufficiente affinché la tensione si stabilizzi





Katherine Johnson e il suo diritto di contare

di Alberto Apostolo

Con grande dolore dobbiamo riferire che Katherine Johnson, ricercatrice matematica che lavorò alla N.A.S.A. nelle prime missioni spaziali negli anni '60 del XX secolo è scomparsa il 24 Febbraio 2020 all'età di 101 anni.

Nata Katherine Coleman il 26 Agosto 1918 a White Sulphur Springs (West Virginia, Stati Uniti), fin da giovanissima manifestò il suo talento per la matematica e al West Virginia State College si laureò in matematica nel 1937 con i più alti onori.

Subito dopo ottenne un impiego come insegnante in una scuola e nel 1939 fu selezionata con altri due uomini afro-americani per frequentare la West Virginia University. Ma dopo la prima sessione lasciò gli studi per formare una famiglia e sposò James Francis Goble. Ebbe tre figlie: Costanza, Joylette e Katherine.

Nell'estate del 1953 riuscì ad essere assunta presso il Langley Research Centre di Hampton in Virginia (Stati Uniti), un dipartimento della N.A.C.A. (National Advisory Committee for Aeronautics divenuta in seguito N.A.S.A. National Aeronautics and Space Administration).

Fu Assegnata alla sezione West Area Computing in un gruppo di lavoro diretto da Dorothy Vaughan (foto accanto).

Doveva occuparsi di analizzare i dati dei test di volo dei prototipi e con altre donne afro-americane si occupava anche della esecuzione dei calcoli manuali.

Nel Dicembre del 1956 perse il marito per un male incurabile. In seguito sposò nel 1959 l'ufficiale dell'Aviazione James A. Johnson. Lottando contro i pregiudizi e la



segregazione razziale riuscì ad affermarsi nel settore aerospaziale come analista delle traiettorie di volo. Nel quadro delle missioni Mercury, nel 1961 effettuò l'analisi della traiettoria per la capsula spaziale Freedom 7 (astronauta Alan Shepard) e nel 1962 effettuò la verifica a mano dei calcoli computerizzati della traiettoria per la capsula spaziale Friendship 7 (astronauta John Glenn). In seguito studiò le traiettorie per le missioni Apollo (in particolare quelle dell'Apollo 11 e dell'Apollo 13), le missioni degli Space Shuttle, all'Earth Resources Satellite e ai piani per una missione su Marte.

Ritiratasi dalla NASA nel 1986 con 26 pubblicazioni in qualità di co-autrice, il suo lavoro rimase oscuro per lungo tempo finché il Presidente Obama le concesse, all'età di 97 anni, la Medaglia Presidenziale della Libertà, la più alta onorificenza civile statunitense. In occasione del 55mo anniversario del lancio del razzo con Alan Shepard, il 5 Maggio 2016 le fu dedicato il nuovo impianto Katherine G. Johnson Computational Research di Hampton. La sua vita e la sua carriera hanno ispirato il film "Hidden Figures" ("Il diritto di contare") del 2016 che ha avuto tre nomination agli Oscar 2017. E' inclusa nella lista della serie

televisiva 100 Women, che contiene un elenco di 100 donne ispiratrici e influenti di tutto il mondo.

L'ATTIVA COMUNITÀ AFRO-AMERICANA DEGLI IMPIEGATI NEL SETTORE AERONAUTICO

La maggior parte di noi, leggendo libri di storia e seguendo documentari in TV, ha l'idea che la segregazione razziale (compresa quella vigente in passato negli Stati Uniti) abbia riguardato solo l'obbligo di occupare o meno spazi riservati negli autobus e nei locali pubblici.

Niente di più sbagliato. La segregazione era assai più invasiva e riguardava a quali lavori si poteva accedere, a cosa studiare nelle scuole, a quali libri leggere nelle biblioteche, ecc. .

Ecco quanto riporta Margot Lee Shetterly autrice del libro "Il diritto di contare" (dal quale è stato tratto l'omonimo film): "C'erano lavori neri, e poi c'erano buoni lavori neri. Smistare il bucato in lavanderia, rifare i letti nelle case dei bianchi, staccare il gambo dalle foglie di tabacco... quelli erano lavori neri. Possedere una bottega di barbiere o un'impresa di pompe funebri, lavorare all'ufficio postale o fare su e giù con le ferrovie come inserviente... quelli erano buoni lavori neri. Insegnare, predicare, fare il medico o l'avvocato: be', quelli erano ottimi lavori neri, che





portavano stabilità e la stima che accompagnava un'istruzione formale."

Lavorare nel settore aeronautico era qualcosa di nuovo e lontano dalla immaginazione collettiva. A partire dagli anni della Seconda Guerra Mondiale si era verificata una mancanza di personale e anche i pochi afro-americani diplomati/laureati in materie scientifiche avrebbero potuto essere impiegati (con stipendi non perequati a quelli dei loro colleghi bianchi). Ma uomini e donne avevano l'occasione per dimostrare quanto potevano valere e diedero un contributo alle imprese spaziali degli Stati Uniti e affinché le leggi sulla segregazione razziale fossero abolite. Le congregazioni religiose e le associazioni ricreative e sociali erano il modo di incontrarsi e di aiutarsi per trovare un alloggio migliore o un nuovo posto di lavoro sempre più qualificato e remunerato. Tra le associazioni più importanti vi era la National Association for the Advancement of Colored People che operava con sezioni a livello cittadino (per es. Dorothy Vaughan era membro fondatore della sezione di Farmville, località dove risiedeva).

LA CALCOLATRICE MECCANICA A CURSORI

Nel film "Il diritto di contare" è incluso l'episodio nel quale John Glenn si rifiutava di partire in missione se prima Katherine Johnson non avesse verificato tutti i calcoli della traiettoria della capsula spaziale Friendship 7. Per esigenze di ritmo narrativo, tale verifica sembra eseguita in modo rapidissimo mentre in realtà la Johnson fu impegnata nei calcoli per diversi giorni, utilizzando una calcolatrice meccanica a cursori.

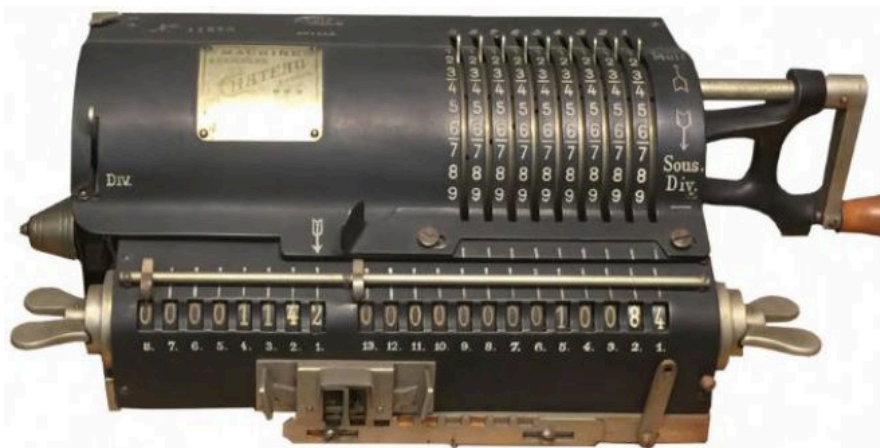


Figura 1

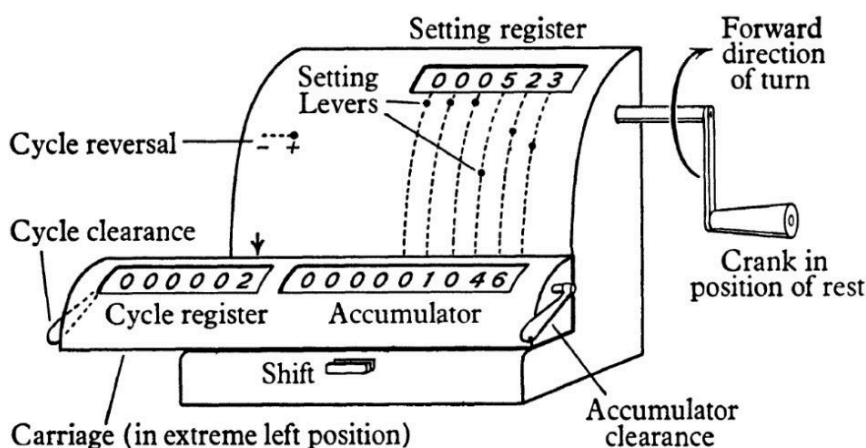


Figura 2

La calcolatrice meccanica a cursori discende dalla calcolatrice di Pascal e dalle macchine pensate da Leibniz, Babbage e altri. Il francese De Colmar, in un tentativo di semplificazione, inventò nel 1820 l'Aritmometro (poi perfezionato nel 1844 e in commercio dal 1851). Lo svedese Odhner nel 1874 progettò e costruì una macchina calcolatrice (anch'essa chiamata Aritmometro) di forma cilindrica e dotata della caratteristica manovella (fig.1). Per inserire un numero si azionavano delle leve (una per ogni cifra) e girando la manovella in un senso o nell'altro si eseguivano addizioni o sottrazioni. Il risultato era visibile sul registro di uscita o

"accumulatore". Altre leve (fig.2) permettevano di azzerare il registro di uscita, il contatore dei giri di manovella o addirittura invertire l'effetto del giro di manovella ("Cycle reversal"). Infine una leva di "shift" permetteva di spostare avanti o indietro il carrello dove si trovava l'accumulatore per moltiplicare o dividere per 10 e rendere possibili moltiplicazioni o divisioni più complicate. Aziende come Brunsviga, Monroe, ecc. cominciarono a costruire calcolatrici, basate su quella di Odhner, fino alla comparsa delle prime calcolatrici elettroniche negli anni '60 del XX secolo.

Bibliografia

[BBC19] AA.VV., "BBC 100 Women 2016: Who is on the list?", consultato il 29/02/2020
<https://www.bbc.com/news/world-38012048>

[Gua19] M.Guarnieri, "Da Habilis a Jobs: due milioni di anni con la tecnologia", Società Editrice Esculapio, 2019.

[Moa65] A.J. Moakes, "Numerical Mathematics", MacMillan, 1965.
<https://archive.org/details/NumericalMathematics/mode/2up>

[She17] M.L. Shetterly, "Il diritto di contare", Harper Collins Italia, 2017.

[She20] M.L. Shetterly, "Katherine Johnson Biography", consultato il 29/02/2020
<https://www.nasa.gov/content/katherine-johnson-biography>

[Zic08] G. Ziccardi, "Informatica giuridica. Manuale breve", Giuffrè Editore, 2008.





Giappone 9^puntata: Mr Game & Watch

di Michele Ugolini

Cari lettori.

Lo sapevamo tutti, prima o poi sarebbe arrivato il momento di parlare di questi importantissimi oggetti della storia videoludica mondiale: i videogiochi elettronici portatili venduti a milioni e milioni in tutto il mondo.

Parleremo dei fantastici Game & Watch. Leggendo gli ideogrammi giapponesi il suono emesso assomiglia più o meno a "Geemu ando uocci".

E' doveroso incamminarsi in questa recensione tenendo presente una importantissima nota discriminante: parleremo di Nintendo "Game & Watch" o di un qualsivoglia altro oggetto dai meri requisiti di bassa tracciabilità riportata sulla propria etichetta? La risposta è articolata, parliamone.

Chi da bambino non ha mai posseduto un italianissimo scacciapensieri elettronico?

Era semplicemente perfetto negli anni 80.

Una plastica dalle colorazioni entusiasmanti.

Un mirabolante pannello LCD con ben due colori: bianco e nero.

Anzi no, un solo colore: il nero, relativo al soggetto del videogioco che veniva illuminato in base agli spostamenti impartiti dal giocatore.

A pensarci bene, erano zero colori: il soggetto in movimento, in realtà, riceveva l'input elettrico di comparire in una posizione prestabilita dello schermo, rispecchiando la medesima logica dei pannelli LCD a sette segmenti, tipici delle calcolatrici, dedicati alla visualizzazione alfanumerica.

Oltretutto, ciascun segmento, non assumeva la forma classica della stanghetta lineare, bensì riceveva dal proprio creatore l'anima del personaggio relativo al videogioco, spesso rappresentata da un omino,

un animaletto oppure un arma da fuoco.

Pochi tasti. Solitamente c'era il pulsante con freccia a sinistra, destra, regolazione dell'ora, il vano batteria da aprire tramite un semplice meccanismo a vite accorpato al bottone di chiusura e null'altro.

In questo stilizzato tripudio di semplicità si sono sbizzarriti gli ingegneri più creativi degli anni 80 per dar vita alle combinazioni più inverosimili, fantasiose ed improbabili al mondo, sfruttando unicamente un limitato numero di segmenti illuminati dell'LCD, attivati da un romantico processore a pochi bit.

Il sostantivo Processore riveste una intrinseca qualità evolutiva che denota un certo grado di nobiltà ingegneristica.

Ammettiamolo, stiamo parlando di hardware rudimentali, eppure meritavano realmente tale appellativo: questi primissimi Game & Watch erano videogiochi, orologi, possedevano l'allarme sonoro della sveglia, emettevano suoni durante il videogioco, in alcuni si poteva selezionare la difficoltà/rapidità del gameplay, alcuni addirittura vantavano la possibilità di eseguire calcoli, infine durante la loro evoluzione tecnologica alcuni addirittura possedevano i requisiti di calcolatore scientifico.

Quanta strada è stata fatta da questi romantici scacciapensieri fino all'attuale gaming in VR!

Sento stratonare il braccio: sarà Mr Game & Watch arrabbiato o saranno i geniali ragazzi della GIG? Tra poco ne parlerò.

Storicamente parlando, Nintendo vanta tutti i diritti di questa famiglia di videogiochi, partoriti per la prima volta al mondo dalla geniale mente di Gunpei Yokoi (Kyoto, 10 settembre 1941 – Komatsu, 4 ottobre 1997).

Questo genio fu assunto dalla Nintendo nel 1965. Nel prossimo articolo vi farò emozionare poiché parlerò soprattutto dell'idea che ha fatto nascere questi adorabili videogiochi. (cfr. figura 1)

Ricordiamo che Nintendo, agli albori, produceva quasi unicamente il gioco di carte tipico del Giappone: l'Hanafuda. Questo mazzo di carte è stato il primo gioco prodotto da Nintendo nel 1889, con enorme successo in tempi talmente remoti che Super Mario non era neppure nei loro pensieri.

Dopo l'ingresso di questo umile nonché grandioso uomo, qualcosa in Nintendo cambiò e noi tutti, oggi, siamo una pura estensione del suo genio.

Il signor Gunpei è famoso per aver ideato anche l'intramontabile Game Boy e soprattutto l'immortale tasto a croce direzionale, ben presto adottato su larga scala, dal mondo videoludico. I Game & Watch hanno vissuto una trasformazione incredibilmente corposa: hanno visto mutare la propria scocca di plastica attraverso numerose forme e dimensioni. Negli ultimi anni della propria evoluzione si è addirittura scomposto in due sezioni pieghevoli, tipo libretto, con un LCD superiore ed uno inferiore. Il passaggio successivo ovviamente è stato di raccogliere tutto il meglio creato fino a quel momento e dar vita ai recenti Nintendo DS e 3DS.

Nei Game & Watch era alloggiato un processore a 4 bit della Sharp con il circuito che gestiva il display e alcune microscopiche aree dedicate a RAM e ROM.

Negli anni 80 il Giappone aveva deciso di esportare circa 3/4 della produzione dei Game & Watch.

Durante il periodo d'oro delle vendite, si stimava l'esportazione annuale di circa 40 milioni di unità





Figura 1

e la conservazione di circa 10 milioni di unità per il mercato interno.

Questa logica, soprattutto recentemente, vista l'evoluzione socioeconomica del paese, è ben diversa. La golosità videoludica/elettronica nipponica, oggidi, vuole soddisfare quasi unicamente l'esigente mercato interno. Le nostre speranze occidentali di export, interessano relativamente poco alle loro megaditte...

Le rigide e lubrificate regole del marketing e soprattutto la possibilità di far attecchire l'oscura trama della fidelizzazione economica relativa ad alcune loro

saghe e titoli, deve solo e soltanto sottostare ad un impietoso giudizio economico.

A tavolino si decide tutto: conviene sfornare un titolo "x" e rendere schiava di quel titolo la fascia di età "target" di una specifica nazione che ha raggiunto un determinato sviluppo socioeconomico?

Se il marketing dice sì, allora quel brand sarà esportato.

Altrimenti quel prodotto rimarrà in pasto al mercato interno, sicuro, rodato, ricco di "Otaku" facenti parte del sistema, che mai potranno nè soprattutto vorranno uscire dal collezionismo relativo a quel prodotto. Accettiamo questa

regola come un dato di fatto: le divergenze socioculturali citate sono mosse da un'anima poco limpida, spesso a noi incomprensibile visto l'incredibile divario evolutivo.

Ritorniamo ai nostri romantici e nostalgici Game & Watch.

Noi italiani abbiamo potuto godere dell'export di questi gioielli della Nintendo? Gli americani e i giapponesi sicuramente sì. I tedeschi, incredibilmente sì, un poco.

Ma noi? Realmente pochissimo. Ecco che sento stratonare il braccio: è Mr Game & Watch molto arrabbiato. Scopriamo perché.

Questi videogiochi, in Italia non hanno goduto di buona espansione come in Giappone oppure in America. Il costo purtroppo era alto. I pezzi erano scarsi.

L'inventiva italiana però non ha aspettato troppo tempo prima di partorire dei cloni altrettanto divertenti. Quanti lettori possono alzare la mano, fieri di aver posseduto un Nintendo Game & Watch originale nella propria infanzia? Quanti invece alzeranno la mano per aver posseduto un clone? (cfr. figura 2)

Clone sembra una parola brutta. Ricordiamo che la GIG Tiger Electronics produceva ottime



Figura 2





plastiche superiori e pannelli LCD seri. Il circuito era stabile e la vita di esercizio è rimasta degna di nota fino ai nostri giorni. I prodotti GIG Tiger, personalmente parlando, meritano un voto alto.

Ne possiedo a decine e solo uno, ad oggi, ha il circuito elettronico compromesso, incredibile!

Alcune imitazioni erano create dalla prolifica VTech (Video Technology Ltd.) di Hong Kong, poliedrica produttrice di articoli elettronici di consumo, oggi celebre per i famosi telefoni AT&T e numerosi altri telefoni ad alta tecnologia per gli uffici.

Al tempo vi era anche la famosa e potente ditta "Polistil".

Questa ditta recuperava l'elettronica della VTech e la impiantava nella ricca serie dei "Time & Fun" tramite numerosi modelli a prezzi abbordabili.

Negli anni 80 erano amichevolmente chiamati Scacciapensieri o Schiacciapensieri.

Inutile dirlo, anche questi sono oggetti da collezione.

Vorrei argomentare anche un triste elemento degno di nota : le quotazioni di alcuni modelli.

Nei principali motori di ricerca dell'usato, questi prodotti originali raggiungono prezzi, a parer mio, ingiusti ed immotivati.

Sicuramente l'inventore non sarebbe fiero di questa situazione difficile e stressante relativa ad un videogioco che doveva togliere i pensieri dalla frenetica attività lavorativa nipponica.

Gunpei voleva far "staccare la spina" del lavoro alla popolazione in cravatta che, esanime, la sera in metropolitana, rientrava a casa dopo vigorose battaglie lavorative! I Game & Watch erano perfetti, la loro anima pura e semplice allontanava lo stress, tramite un confortevole e limitatissimo gameplay.

Il fenomeno Candy Crush Saga, secondo me, ha condiviso molti ideali relativi ai Game & Watch.

Questa odierna ricerca tanto

difficoltosa, unita all'escalation di prezzi proibitivi, però, stanno maturando una particolarissima situazione e, prossimamente, ci saranno grandi novità, probabilmente proprio in Italia.

Ottime notizie per gli ideali del signor Gunpei e soprattutto per noi futuri acquirenti.

Ne parlerò nelle prossime puntate.

Ricordiamo anche che, il caro signor Gunpei, collaborò nello sviluppo di Mario Bros, Kid Icarus e Metroid, un genio immortale al quale vanno tutti i nostri ringraziamenti.

Vogliamo stilare una breve lista dei Game & Watch più famosi?

Balloon fight (chi non l'ha amato?), Ball (il personaggio è addirittura diventato l'icona/mascotte predefinita per i Nintendo Game & Watch), Boxing, Donkey Kong (serie ricca di molte varianti), Fire (universalmente riconosciuto), Manhole, Mario Bros (forse la serie più prolifica), Mickey Mouse, Octopus, Oil Panic, Parachute (una tra le serie più famose), Snoopy, Tetris, Vermin, Zelda, etc...

L'intera lista è consultabile su Wikipedia:

https://it.wikipedia.org/wiki/Lista_di_Game_%26_Watch

Le varie serie dei Game & Watch si sono sviluppate ufficialmente dal 1980 al 1986. Sei anni prolifici che hanno sfornato queste serie:

Silver (1980), Gold (1981), Widescreen (1981-1982), Multiscreen (1982-1989), Tabletop (1983), Panorama (1983-1984), New Widescreen (1982-1991), Super Color (1984), Micro Vs. System (1984), Crystal Screen (1986).

Se siete curiosi delle loro forme e specifiche, troverete tutti i dettagli in questo link:

https://it.wikipedia.org/wiki/Game_%26_Watch

Se siete curiosi di scoprire chi è il signor "Misutaa geemu ando uocci" che mi stratonava il

braccio, nato nel 1980 e comparso per la prima volta in Italia nel 2002 in Super Smash Bros Melee (per Nintendo Gamecube) potrete curiosare in questi due link:

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

https://it.wikipedia.org/wiki/Mr_Game_%26_Watch

Bene, cari lettori, per questa puntata introduttiva, è tutto.

Nella prossima puntata vi parlerò di una divertente intervista.

Vi parlerò anche e soprattutto di come è meccanicamente nata l'idea del Game & Watch: una storia relativa ad un... proiettile!

Infine vi accenno che nelle future puntate, parlerò di resurrezioni dalle ceneri, recenti progetti e prossime speranze di questo mondo dalla stilizzata anima immortale.

A presto!





BATTLE CHESS

Alzi la mano chi non conosce questo gioco, Battle Chess, un gioco di scacchi innovativo e favoloso; adesso le pedine si muovono e quando si trovano sulla stessa casella si affrontano in un combattimento all'ultimo sangue! Ovviamente le regole sono quelle degli scacchi normali ma l'idea dell'animazione aggiunge un tocco di classe a questo stupendo gioco.



Il menù a tendina, accessibile con il click destro del mouse, è ricco di opzioni e permette di configurare tutte le feature del gioco.



E' possibile scegliere se giocare contro un avversario umano, anche in remoto con un modem, oppure sfidare il computer; ci sono ben 10 livelli di difficoltà, e l'ultimo livello darà del filo da torcere anche ad un giocatore esperto.



Per chi non volesse perdere tempo

con le animazioni dei pezzi, è presente la possibilità di giocare in 2D e di conseguenza accelerare notevolmente la partita.



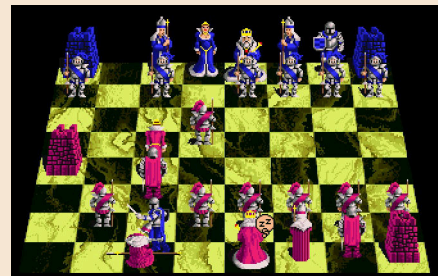
Ma resistere al fascino del combattimento animato è difficile: la torre si trasforma nella "cosa" ed usa la sua forza bruta per distruggere gli avversari, la regina invece è una maga esperta e con la magia polverizza chi le si para di fronte, per non parlare dell'alfiere che, impersonato da un vescovo, usa il pastorale come una lancia. Il cavallo invece, esperto spadaccino, decapita gli avversari con un colpo di spada...

La grafica è a dire poco stupenda e le animazioni dei pezzi sono spettacolari e coinvolgenti. L'unica pecca di questo gioco è il sonoro, avrebbe potuto essere curato maggiormente e forse qualche pezzo di musica classica avrebbe aggiunto spessore ad un titolo già di per se' superbo.

Ah, dimenticavo, è presente sul dischetto la possibilità di installare il gioco su HD perchè le animazioni vengono caricate dal dischetto al momento del bisogno e alla lunga i continui accessi al disco possono risultare snervanti. Installando il gioco su HD questo problema scompare.

di **Francesco Fiorentini**

Anno: 1988
Sviluppatore:
Interplay
Piattaforma: Amiga
Genere: Scacchi



GIUDIZIO FINALE

» Giocabilità 80%

Un gioco che 'prende' immediatamente per l'approccio innovativo dei combattimenti...

» Longevità 80%

...ma proprio questa sua prerogativa potrebbe limitarlo verso il pubblico degli scacchisti più esigenti.





BUBBLE BOBBLE

Developer: Taito
Anno: 1986

Piattaforma: Multi
Genere: Piattaforme

Correva l'anno 1986, Maradona ci regala il gol di mano del secolo, collasso la centrale nucleare di Chernobyl, in edicola si paleso il primo incredibile numero di Dylan Dog.

Eravamo in piena "MILANO DA BERE", eravamo ancora piccoli e al mare, durante le vacanze estive, infilavamo gettoni nei gloriosi cabinati del tempo.

Nell'agosto del 1986 nella sala giochi Galaxy in Piazza 8 agosto a Bologna, mentre la radio sparava a "bomba" Papa Don't Preach di Madonna vedo Lui... Bubble Bobble.

Lo vedo accendere dal titolare e sento il jingle iniziale... Booom!!! Che flash!!!

Subito, con mio fratello piccolo ci proviamo e vediamo di capire come funziona. Il resto è storia. Non vedo l'ora di capirci di più, di conoscere di più, di vedere altri giocatori giocarci, scoprire i trucchi (e ne ha milioni) e giocare in cooperativo assieme ad amici, fratello, ragazza (beh... al tempo attirava anche le ragazzine, una novità in sala giochi...).

Era cominciata un'era.

Siamo quella generazione. La generazione Bubble Bobble e siamo tra chi ama in modo viscerale il capolavoro assoluto di Fukio Mitsuji, padre e game designer che ci ha lasciato troppo presto.

Nella mente di Fukio e dei suoi collaboratori era scattato qualcosa mentre progettavano il gioco, ovvero quello di far fare un viaggio ai giocatori attraverso 100 livelli dove avrebbero aiutato i protagonisti del gioco a salvare le proprie ragazze e a "ritornare" umani. Che colpo di genio. Un

viaggio di crescita che trasforma, che vi mette nelle "mani del compagno" perché da soli è divertente ma in cooperativo è speciale. Solo attraverso la giusta collaborazion si arriverà a sfidare il terribile Grumple Grommit, il Boss finale.

Un gioco generazionale, giocabilissimo e superbo sotto tutti i punti di vista. Un viaggio tra i livelli e un viaggio dentro di noi.

Il tutto utilizzando l'arma più insospettabile in assoluto, le bolle di sapone!

Ma bando alle ciance e ai ricordi, andiamo al sodo!

Bubble Bobble è un videogioco arcade (coin-op) di genere a piattaforme con quadro fisso.

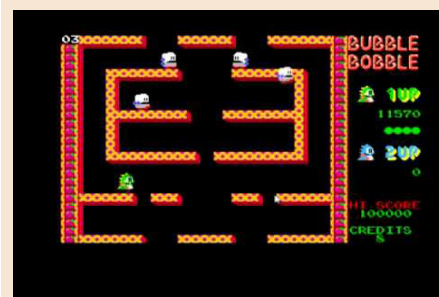
Uno o due giocatori prendono il ruolo di Bub e Bob, due simpatici ragazzini trasformati in draghetti sparabolle che devono farsi strada attraverso 100 livelli zeppi di piattaforme per salvare le loro fanciulle.

Bub e Bob sono armati solamente della loro capacità di creare bolle come vi dicevo, in cui i molti nemici del gioco possono essere intrappolati ed uccisi. Le bolle possono essere usate anche come piattaforme temporanee, per cercare di raggiungere aree del livello altrimenti irraggiungibili.

Lo scopo di ogni livello è quello di sterminare tutti i mostri presenti sullo schermo per poter avanzare al successivo. Se i nemici si libereranno dalla prigionia della bolla vi attaccheranno con più ferocia e maggior velocità (appariranno rossi di rabbia), e se aspetterete oltre il tempo limite comparirà lo Skel Monsta, un nemico invulnerabile che comincerà a darvi la caccia fino a che non completerete il livello (o verrete uccisi da lui).



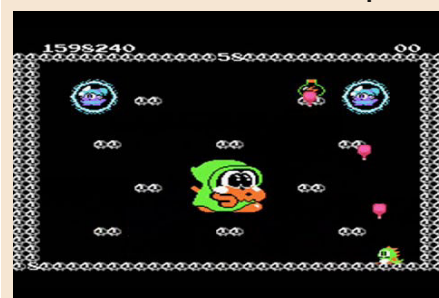
Versione C64



Versione Amstrad CPC



Versione Arcade Coin Up



Versione Nintendo NES



Versione Spectrum Zx





CONVERSIONI PER IL MERCATO CASALINGO

Dal 1987 fino ad oggi Bubble Bobble è uno dei giochi più convertiti di sempre.

Abbiamo numerose versioni per i più svariati sistemi casalinghi e tra le meglio riuscite non possiamo che non citare la superba versione per Commodore C64 del 1987 di Firebird, sviluppata partendo solamente grazie alla volontà e alla passione di Stephen Ruddy che riprodusse il gioco studiando il coin op.

Abbiamo poi le versioni per Amiga e Atari St di ottima fattura e persino le conversioni per Zx Spectrum e CPC Amstrad (davvero colorata).

Anche per le console dell'epoca furono realizzate conversioni. Sia per il Nes della Nintendo, molto colorata anche se sofferente in alcuni punti di rallentamenti, oppure quella per Sega Master System veloce, divertente e con l'aggiunta di ulteriori 100 livelli.

Negli anni abbiamo visto anche le versioni rivedute graficamente in maniera più o meno efficaci per Playstation, Nintendo DS, Wii, Pc, Apple II e per i sistemi più strani come Sharp X68000 o FM TOWNS Marty.

Segnalo anche la splendida versione per MSX/MSX 2, una delle più fedeli all'originale da SALA.

L'EREDITA'

Negli anni la saga di Bubble Bobble ha preso due diverse strade.

Quella dei seguiti che presentano

due gioielli ovvero Rainbow Islands e Parasol Stars, dove i nostri protagonisti non sono più draghetti ma sono ritornati bambini e le dinamiche di gioco sono differenti.

La serie invece ha anche una serie di sequel dove Bub e Bob sono ancora draghetti e proseguono la loro avventura a suon di bolle nei giochi Bubble Bobble part 2 per NES 8bit oppure nella serie Revolution/Evolution per i sistemi più moderni.

Abbiamo anche uno spin off di successo incredibile, quella di Puzzle Bobble/Bust a Move, puzzle game coloratissimo sullo stile di tetris, columns e klax che impazzò negli anni 90 e che tutt'ora vanta una marea di appassionati.

Per finire vorrei citare una super hack rom realizzata da Aladar e da Andrea "Bisboch" Babich chiamata Bubble Bobble Lost Cave. Realizzata partendo dalla Rom originale e inserendo "100 nuovi livelli". Una versione per nostalgici ma anche per chi vuole cimentarsi in una nuova chiave di lettura. Bellissima versione che vi consiglio di giocare.

Siamo arrivati alla fine. Bubble Bobble è ancora un mega sogno, un gioiello che nel 2020 emoziona come nel 1986. Giocatelo! Giocatelo in singolo, in doppio e su qualsiasi sistema possibile.

Ora basta! Devo arrivare al 20 livello senza morire... Il warp mi aspetta. ;)

di Carlo Nithaiah Del Mar Pirazzini

Bubble Bobble Lost Cave



GIUDIZIO FINALE



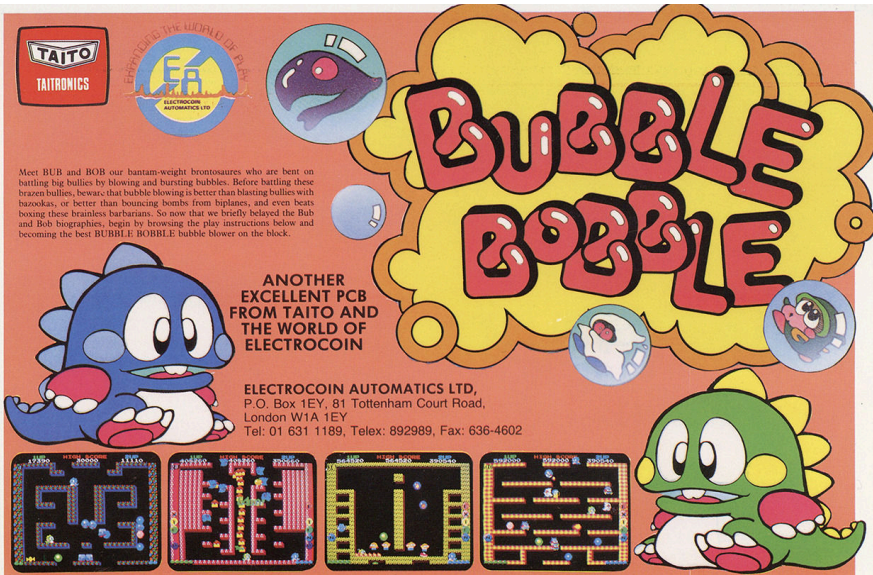
» Giocabilità 95%

Che sia su Zx Spectrum, su Amiga, Nes, Megadrive oppure su Android.. Che sia in versione Arcade originale o bootleg oppure nella sua incarnazione Lost Cave, è tra i migliori videogames di sempre. Bilanciato, dinamico e incredibilmente bello giocato in doppio.

» Longevità 95%

Lo giochi, lo rigiochi... Lo accendi e lo carichi... E lo rigiochi... Devo aggiungere altro?

Puzzle Bobble





ACTRAISER

"Sei tu forse un Dio?"

Con questa frase Gozer il gozeriano si rivolgeva a Ray sul tetto della centrale fantasma prima di fotonizzarlo dopo un secco no (in Ghostbusters di Ivan Raitman nel 1985)

Chi di voi però ha interpretato una divinità nel mondo videoludico e si è permesso di dire Sì agli avversari, popolazioni ribelli e forze della natura? Direi in molti, dai tempi di Populous fino ai più recenti sviluppi della seria Age of.. , ci siamo cimentati nella creazioni e distruzione di intere civiltà.

Oggi parliamo di un prodotto che alla sua uscita fu un punto di svolta nel mercato videoludico, soprattutto in quello per console.

Actraiser è videogame seminale, il punto di raccordo tra 2 generi apparentemente lontani ma che fusi nella maniera di questo gioiello sono riusciti a ricombinarsi in qualcosa di realmente nuovo e inaspettato.

Una novità per l'epoca dove o eri un platform o eri un simulatore di divinità e che diventò una delle killer application (si dice così? ndN). Oggi lo andiamo a rispolverare accendendo il nostro Snes pronti a riscoprire un classico dell'innovazione. Un portabandiera del genere. Insomma in poche parole... anzi in una sola parola... EPOCALE!

La trama è netta. Impersoniamo una divinità nota come The Master. Secondo il libretto delle istruzioni, il demone Tanzra, nel tentativo di impossessarsi del nostro regno, ci

ha coinvolto in una mega battaglia contro i Sei Guardiani, suoi sottoposti. La battaglia è risultata ardua e ne siamo usciti sconfitti, ritirandoci nel nostro Sky Palace e guardando il nostro regno distrutto e posseduto dal demone reietto.

Centinaia di anni dopo, aiutato da un angelo guardiano, ci siamo ripresi e abbiamo deciso di intraprendere la riconquista delle nostre terre e dei cuori delle genti che le abitano.

Per ottenere questo scopo è necessario:

1 - liberare ciascuna delle sei regioni dai mostri invasori, incarnandosi in un potente guerriero.

2 - far ricrescere la popolazione all'interno di queste lande desolate ora libere, grazie ai propri poteri divini (definiti miracoli nel gioco): questo accrescerà la fede delle genti nel Master e causerà la rinascita della civiltà, che nella meccanica di gioco si traduce con l'aumento di punti esperienza necessari a far salire di livello il Master.

3 - Affrontare lo scontro con il Guardiano a capo della zona; una volta sconfitto quest'ultimo, la regione sarà definitivamente libera dal male e potranno essere sbloccate altre aree da riconquistare.

Ciascuna di queste fasi va seguita per liberare definitivamente una regione, e alla prima e alla terza corrispondono un "atto" (Act, da cui molto probabilmente deriva il nome ActRaiser).

Questi "ATTI" sono divisi in atti prettamente platform sullo stile di

Sviluppatore: QUINTET

Editore: ENIX

Anno: 1990

Piattaforma: Super Nintendo

- disponibile anche su Switch

Genere: ACTION/GOD GAME



GIUDIZIO FINALE



» Giocabilità 85%

Una volta appresi i comandi grazie al tutorial "angelico" andremo dritti al centro della storia. Non difficile, ma non sempre perfetto nei comandi. Superlativa la sezione simulativa per tutti gli amanti del mix Sim City/Populous.

» Longevità 90%

La sezione action è velocissima e frenetica, quella simulativa costruttiva, ponderata, lenta al punto giusto. Divertentissimo costruire e gestire la vita dei nostri villaggi. Un bel numero di quest ci terranno impegnati per molto tempo.





Rastan e in fasi di gioco simulate e di costruzione che ricordano il già citato Populous, Powermonger (chi se lo ricorda ?? NdN) e Sim City.

Tutto parte dallo Sky Palace, dove il nostro angelo guida ci illustra le modalità di gioco. Da questo posto possiamo spostarlo sopra il nostro continente da riconquistare e affrontare le varie fasi. Una volta selezionata la sezione dove vorremo intervenire si verrà catapultati nella fase di gioco prettamente platform e, solo dopo aver sconfitto il guardiano, sarà possibile accedere alla fase simulativa.

In questa fase avremo il compito di guidare il nostro angelo a difesa della cittadella che dovremo costruire.

Muovendoci nella mappa avremo le possibilità di costruire le strade dove si comincerà a sviluppare la vita cittadina, ripulire zone da vegetazione e ostacoli di vario genere, effettuare i miracoli per poter accrescere la fiducia del popolo verso di noi e, come se non bastasse, uccidere alcuni demoni che cercheranno di ostacolarci.

Ogni volta che effettueremo una determinata azione (rimozione delle tane dei demoni, piccole quest che ci verranno richieste al tempio, ecc...) ci verrà donato dal popolo un'offerta che potrà essere personale (armi più potenti, nuova magia) oppure cumulativa e per la comunità che potremo utilizzare nelle varie mappe simulate.

Le regioni da liberare sono sei, una volta ripulite avremo accesso all'area finale dove affronteremo di nuovo i sei guardiani e quel delinquente di Tanzra.

Parliamo di sostanza... ovvero del comparto tecnico. La sezione platform è realizzata dignitosamente anche se ovviamente non è il punto di forza del gioco. Dignitosa ma qualche imprecisione come nel caso di alcuni salti che effettueremo non

sempre pixel perfect.

Nella fase simulativa invece c'è molta cura nel dettaglio, animata con dovizia di particolari e molto colorata.

Il comparto sonoro invece è di altissimo livello, con una colonna sonora che anche dopo 30 anni dall'uscita risulta spettacolare. L'autore è infatti Yuzo Koshiro, che ha realizzato alcune colonne sonore per videogames davvero notevoli (come quella di Street's of Rage per Megadrive).

Il punto di forza del gioco sta nella sua dualità. Questo suo essere tra un Sim City e un Rastan lo rende intrigante. Le fasi sono ben spiegate e non ci troveremo mai in punti morti nella fase simulativa grazie ad una buonissima narrazione.

Finarlo non è impossibile, ma ci terrà impegnati per diverso tempo e, se saremo stati bravi, una volta ucciso il cattivo finale si accederà ad una modalità chiamata "pro" che ci permetterà di affrontare il titolo con un maggior livello di difficoltà.

Actraiser, un classico senza tempo che ci farà sognare e ci terrà incollati al pad del nostro snes. Onirico!

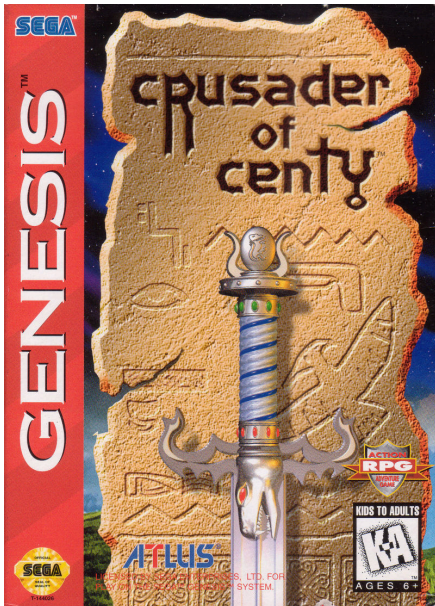
di Carlo Nithaiah Del Mar Pirazzini





CRUSADER OF CENTY

Sviluppatore: NEXTECH
 Editore: ATLUS
 Anno: 1990
 Piattaforma: Sega Megadrive/
 Sega Genesis
 Genere: ACTION/GDR



Come Zelda... ma su Megadrive!

Crusader of Centy, noto anche come Soleil in Europa (lo stesso nome della città principale del gioco) e Ragnacenty in Giappone, è membro di una razza morente di veri Action / GdR (alla Legend of Zelda).

Come i giochi Genesis di ultima generazione, Beyond Oasis, Crusader of Centy e Light Crusader hanno davvero contribuito a rafforzare la libreria di giochi di ruolo della console, che non era abbastanza ampia come la gamma di giochi di ruolo disponibili per il suo concorrente a 16 bit, SNES ma presentava una serie di prodotti davvero interessanti e di indubbio valore.

Ciò che era importante per i gdr su Megadrive/Genesis era la qualità sulla quantità. Giochi come Shining Force 2 e Phantasy Star 4 dalla sua parte nessuno può negare che il Megadrive/Genesis fosse la patria di alcuni dei migliori giochi di ruolo dei suoi tempi. La maggior parte dei suoi giochi di ruolo di origine giapponese sono stati fortunatamente tradotti e hanno raggiunto le nostre coste più o meno

intatti.

Quando chiamo Crusader of Centy un vero Action / RPG, con questo intendo Action / RPG "vecchia scuola"; che contiene un buon equilibrio tra combattimento, puzzle, salto di piattaforma, parlato ed esplorazione.

La maggior parte dei giochi di ruolo / azione moderni si concentra troppo sull'hacking e lo squarcio attraverso eserciti infiniti di mostri con zanne che sbavano dalla bocca, troppo per i miei gusti.

Questo non vuol dire che tali giochi non possano essere divertenti (possono essere molto divertenti), ma qualsiasi vero Action / RPG non dovrebbe girare interamente attorno al combattimento. Grazie all'ascesa alla fama e alla fortuna di Diablo, un esercito di cloni è sceso su tutti noi, rimodellando questo sottogenere RPG in quello che è diventato oggi (forse il termine Action / RPG dovrebbe essere ridefinito come "hack 'n' slash", perché è quello che è diventato sinonimo in questi tempi).

Ma torniamo al gioco.

Per quanto riguarda la storia, interpreteremo il ruolo di un ragazzo che ha appena compiuto quattordici anni. Rifiutando di staccarsi dalla tradizione, la madre gli regala la spada del padre per indicare il passaggio dalla fanciullezza alla maturità. Il prossimo passo sarà visitare il re prima di metterci alla prova nell'addestramento al combattimento nelle pericolose aree infestate dai mostri fuori dal regno.

Nonostante la semplice premessa, una svolta all'inizio della storia vede la capacità del personaggio di parlare con gli umani portati via da lui (il linguaggio umano diventa incomprensibile per lui) e sostituito





GIUDIZIO FINALE



» Giocabilità 90%

Una volta appresi i comandi e capito il meccanismo, il gioco vi immergerà in un mondo incredibile e colorato. Possiamo considerarlo lo "ZELDA" per Megadrive. Innovativa la collaborazioni con gli animali per potenziare il personaggio.

» Longevità 80%

Quest non troppo difficili e il gioco è piuttosto lungo, peccato per la mancanza di localizzazione e la poca reperibilità sul mercato (a prezzi non sempre accessibili). Ovviamente il voto sale di 5 punti se non si desidera la versione fisica ma si vuol inserire in emulatore, megadrive mini o retropie.

con la capacità di parlare con animali... e mostri.

Dal punto di vista tecnico è stato fatto un lavoro davvero notevole. Come molti giochi della sua epoca anche questo presenta una grafica davvero colorata con visuale dall'alto verso il basso. Un lavoro che ha spremuto ben bene la console ed è una gioia per gli occhi.

Il sistema di controllo attraverso il pad è molto intuitivo e reattivo. Si parte con l'eroe che sa solo utilizzare la spada, ma che con il proseguire delle missioni e dell'esperienza aumenta le sue abilità. Come dicevo non sarà necessario affrontare a testa bassa ogni mostro o trappola, ma sarà chiesto soprattutto all'inizio di schivare, saltare e aggirarli per evitare una morte prematura.

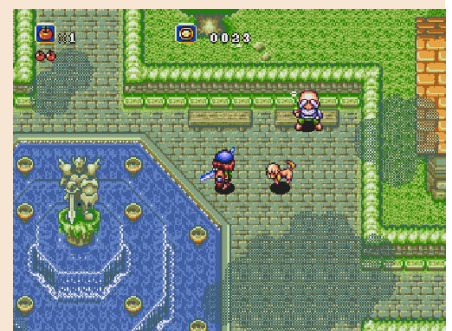
Ciò che separa questo gioco dagli altri giochi di ruolo / azione è che puoi reclutare i servizi di vari compagni animali, che incontrerai durante il tuo viaggio, che ti proteggeranno

individualmente dai danni e ti garantiranno nuove abilità o miglioreranno quelle vecchie. Il compagno animale pinguino trasformerà la spada in una lama di ghiaccio, ad esempio, che consentirà di congelare la lava che scorre quando viene lanciata contro di essa. L'aggiunta del ghepardo permetterà naturalmente correre come il vento, dandoti la velocità extra necessaria.

In soldoni tutto questo aumenta sensibilmente la longevità e la giocabilità del prodotto.

Un unico appunto. Il gioco è solo in inglese e non è stato localizzato in altre lingue e purtroppo non è facilmente reperibile in copia fisica (si trova ma ovviamente costicchia).

di **Carlo Nithaiah Del Mar Pirazzini**





AUTOMANIA

Anno: 1984
Sviluppatore: Mikro-Gen

Piattaforma: C64 & ZX
Spectrum
Genere: Azione

Sono dieci giorni ormai che viviamo in questa specie di psicosi collettiva dettata dal fenomeno coronavirus, quasi chiusi in casa con mille precauzioni e con pochi servizi esterni a disposizione dato che è tutto fermo o con orari ridotti, ma noi non ci fermiamo!

Avremo più tempo a disposizione in casa nostra con pc e tablet per poter scrivere centinaia di righe di recensioni, articoli ecc...

Allora approfittiamone!

Il gioco di questo numero è un platform uscito nel 1984 per Commodore 64 e ZX Spectrum pubblicato dalla Mikro-Gen. Un Platform dove la precisione millimetrica è il segreto. Stiamo parlando di Automania! Il primo della serie di videogiochi di Wally, il simpatico operaio inglese protagonista di questo e dei successivi giochi di cui parleremo in futuro. Il gioco comincia con il nostro protagonista in una fabbrica di automobili che ci ricordano molto le vetture degli anni 70/80, infatti la prima è una Citroen 2cv.

Ogni livello è composto da due schermate fisse, ossia una stanza con i vari pezzi da recuperare e vari nemici da evitare come ruote, oliatori... Tutti attrezzi da officina per intenderci; mentre nella seconda vedremo l'automobile che dovrà essere completata pezzo per pezzo per poi poter sfrecciare a tutta velocità! Recuperare i pezzi non sarà facile, avremo a disposizione il solo salto per fare tutto con i nemici che dovranno essere evitati con una considerevole precisione, inoltre si perdono vite cadendo da certe

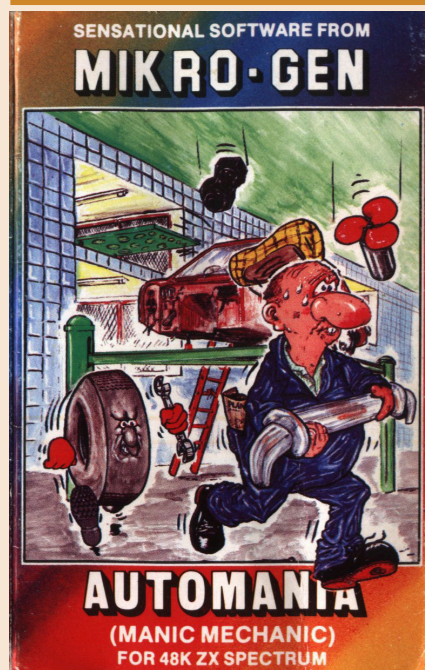
altezze, senza contare il fatto che per recuperare e posizionare ciascun pezzo nel punto giusto, avremo un limite di tempo! Ma non allarmatevi, che già di allarmismi in questo periodo ne abbiamo troppi! Al termine di ogni livello, si guadagnerà una vita extra.

Il nostro scopo è quello di assemblare ben dieci automobili e alla fine della decima si ricomincerà dall'inizio con una difficoltà maggiore (prassi comune per i giochi dell'epoca non avere finale, se non pochi). Per terminare questo gioco dovrete armarvi di tanta pazienza e fare parecchia pratica con il Joystick o la tastiera e soprattutto non disperarvi. Questo gioco offre un gameplay piuttosto lungo, dieci automobili non si assemblano in una decina di minuti, ma a farvi compagnia ci sarà come sottofondo Dance of the Cuckoos, il tema di Stanlio e Ollio. Sono certo che dopo la prima partita alcuni di voi correranno a documentarsi sull'intera saga di Wally, come ho fatto io e devo ammettere che ne è valsa la pena perché ho ripescato parecchi giochini andati perduti nel tempo, tra cui questo.

Il seguito di Automania è Pijamarama, che si svolge in un contesto diverso e con nuovi compiti da eseguire.

Nei successivi giochi faremo inoltre la conoscenza di amici e famiglia del nostro Wally; ma di questo ne parleremo più avanti una volta che Vi sarete guadagnati la giornata in fabbrica montando automobili d'epoca!

di **Daniele Brahimi**



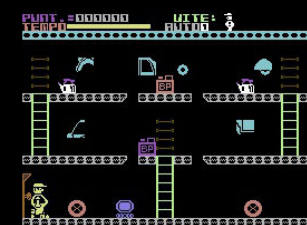
GIUDIZIO FINALE

» Giocabilità 60%

Poche parole: precisione e domestichezza!

» Longevità 65%

Impegnativo ma alla lunga...





PSYCHIC 5

Publisher: Jaleco
Anno: 1987

Piattaforma: arcade
Genere: platform

Ci sono giochi che sono di difficile collocazione e che si presentano così originali che ci fanno innamorare già al primo sguardo. Uno di questi titoli è sicuramente Psychic 5 che è a metà tra un platform, un rimpicapo ma anche un action game.



Lanciato nelle sale giochi dalla Jaleco nel lontano 1987, Psychic 5 colpisce subito per la sua giocabilità straordinaria e per la simpatia dei suoi sprites.

I protagonisti sono 5 esper che possiamo cambiare nel corso del gioco semplicemente sostando per alcuni secondi in una cabina telefonica (e qui il riferimento a Superman è davvero scontato). Ogni player ha delle caratteristiche personali che vengono riassunte in una schermata prima dell'inizio del gioco.

Inizialmente si parte con Naoki, un bambino che risulta essere il più equilibrato dei cinque, ma possiamo anche utilizzare la bambina Akiko, dotata di un grande salto. Gli altri personaggi possiamo sbloccarli durante i vari

livelli.

Le nostre uniche armi sono il salto e soprattutto un martello con il quale possiamo colpire i nemici e rompere vasi all'interno dei quali troviamo bonus sotto forma di punteggi e tempo aggiuntivo.

I livelli di gioco, che si presentano come dei labirinti da percorrere, sono in tutto otto, ognuno con un tema e una grafica particolare.

I mostri che li popolano si rifanno al tema del livello stesso e quindi cambiano di volta in volta.

Alla fine di ogni livello dobbiamo entrare nella stanza del perfido Satan, che ci aspetta seduto sul suo trono mentre ci lancia palle infuocate e cercare di eliminarlo dopo una serie ripetuta di colpi.

Il gameplay è poi arricchito da geniali trovate come passaggi segreti, oggetti nascosti e la possibilità di raccogliere lettere e scrivere la parola extra, che ci darà una vita aggiuntiva.

Inoltre lungo tutto il nostro cammino sarà costante la presenza di una strega che vola su di una scopa, se riusciamo a colpirla potremmo impossessarci per alcuni secondi della magica scopa e volare indisturbati all'interno del livello e distruggere tutto.

Anche il sonoro è all'altezza del resto del gioco, con una musica accattivante che crea la giusta atmosfera ed effetti sempre adeguati.

Insomma in Psychic 5 sembra che gli sviluppatori hanno pensato ad ogni minimo particolare, rendendo la sua esperienza di gioco ancora oggi piacevole ed originale.

di **Querino Ialongo**



GIUDIZIO FINALE



» Giocabilità 90%

Psychic 5 si presentò con un gameplay davvero originale e la giocabilità resta sicuramente il suo punto di forza.

» Longevità 90%

Sono passati più di trenta anni dal suo lancio, ma l'emozione che si prova nell'iniziare una nuova partita è rimasta incredibilmente la stessa.





Avventure Testuali - Intervista a Vincenzo Scarpa

di Giorgio Balestrieri

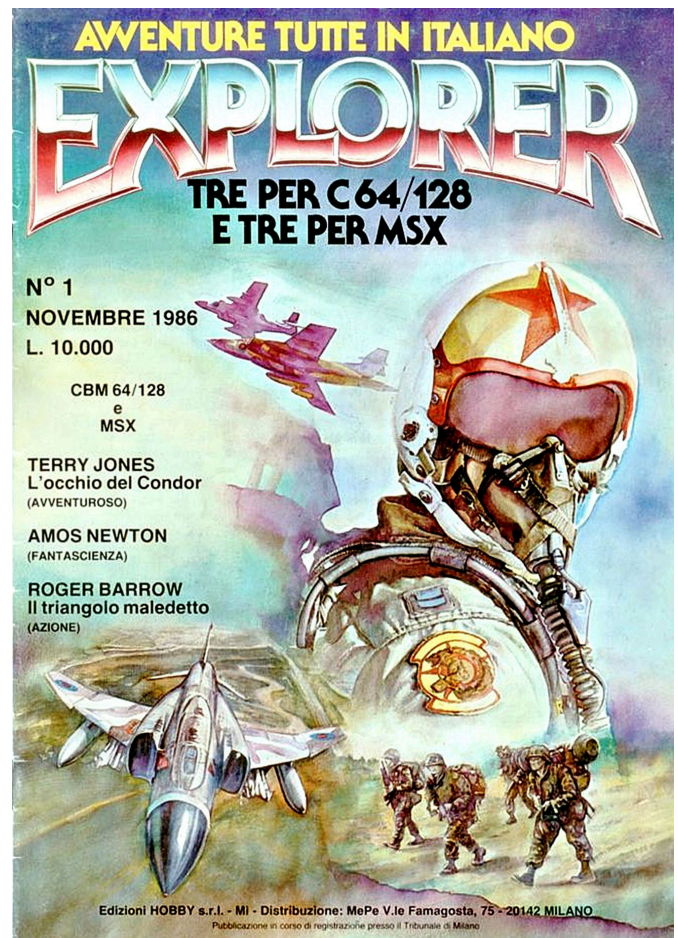
I nostri affezionati lettori sapranno ormai quanto il tema dei giochi d'avventura, sia testuali che grafici, sia caro a noi della redazione ed in particolare a chi vi scrive. In questo articolo vogliamo cogliere l'occasione offertaci da Vincenzo Scarpa, l'uomo dietro EmuWiki e che condivide con noi questa passione, per parlare un po' di queste affascinanti opere interattive. Nate dall'ispirazione derivata in parte dai GdR ed in parte dai librigame, le avventure testuali (AT) vedono la luce nel 1975, con la creazione di Colossal Cave Adventure (o Advent), programmata da Will Crowther in Fortran su un computer PDP-10. Il gioco, nato inizialmente come una riproduzione interattiva del complesso di caverne del parco nazionale di Mammoth Cave, fu in seguito ampliato da Don Woods, che ne rilasciò anche il codice sorgente, e liberamente distribuito. Sulla scia dell'entusiasmo che ne scaturì, Scott Adams decise di programmare e pubblicare nel 1979 un suo gioco d'avventura, Adventureland, in BASIC per microcomputer TSR-80, inaugurando così il filone commerciale per questo genere di prodotti. Da lì, e per tutti gli anni '80, le avventure testuali hanno vissuto il loro periodo d'oro, diventando molto popolari ed apprezzate. In seguito, con l'arrivo delle avventure grafiche e del cambio dei gusti dei consumatori, lo sviluppo commerciale delle AT cessò completamente e le case di produzione che avevano basato il loro business esclusivamente su questo tipo di prodotto, andarono incontro ad un triste destino costrette, praticamente tutte, a chiudere i battenti. Gli estimatori del genere però non hanno mai rinunciato a questo tipo di opera e molti hanno iniziato a svilupparne in proprio, distribuendole in forma gratuita, mantenendo così in vita il genere ancora oggi. Noi di RetroMagazine abbiamo affrontato più volte l'argomento nei numeri scorsi, esaminando diversi aspetti ad esso correlati, stavolta vogliamo farlo con una chiacchierata "a 360°" con Vincenzo Scarpa.

Bentornato Vincenzo, e grazie per l'occasione che ci hai dato di parlare delle avventure testuali, un tema a me particolarmente caro. Qual è stato il tuo primo incontro con questo tipo di giochi?

Grazie a voi per avermi concesso un'altra intervista. Il mio primo incontro con le avventure testuali è avvenuto con le cassette per il Commodore 64 che venivano vendute in edicola. All'epoca, come ben sai, esistevano delle intere collane dedicate a questo genere di giochi e un bel giorno ne acquistai una (credo fosse un numero di Explorer, ma non ricordo più quale) e rimasi fin da subito folgorato.

Cosa ti ha colpito di esse e quali sono quelle che hai amato di più? Avevi una predilezione per una casa o un autore in particolare?

Beh... sicuramente la possibilità, per chi come me ado-



Copertina della rivista "Explorer", per gentile concessione di Edicola 8 bit

rava - e adora tutt'ora - leggere, di poter in qualche modo diventare il diretto protagonista di una storia, anche se spesso incappavo in degli enigmi che richiedevano diverso tempo per poter essere risolti.

Ai tempi amavo in particolar modo le avventure testuali scritte da Bonaventura Di Bello (quelle appunto che uscivano in edicola su Explorer, Viking e non solo) ma anche quelle di Scott Adams e soprattutto le Mysterious Adventures di Brian Howarth. Tra queste ultime ho particolarmente adorato Ten Little Indians, un gioco ispirato all'omonimo romanzo di Agatha Christie che, manco a farlo apposta, è in assoluto anche uno dei miei preferiti.

In questo genere di giochi preferisci quelle avventure che si concentrano maggiormente sulle descrizioni o quelle che prediligono invece la meccanica di gioco? E cosa ci sai dire sulle CYOA?

Partendo dal presupposto che il gioco è sempre lo stesso (sempre cioè di un'avventura testuale si tratta) preferisco in genere quelle che hanno delle descrizioni dettagliate dei luoghi, in quanto mi aiutano ad immergermi (se così si può dire) nella storia in sé. Anche quelle però che si concentrano maggiormente sulle meccaniche di gioco (con di conseguenza delle descrizioni dei luoghi





più scarse) possono essere ugualmente interessanti. “Avventura nel Castello” di Enrico Colombini su tutte.

Le CYOA (acronimo di Choose Your Own Adventure) sono invece le avventure testuali a scelta multipla, che ricordano molto da vicino i classici Librigame: in questo caso il giocatore deve prendere delle decisioni direttamente imposte dai giochi stessi e, sotto certi aspetti, sono anche più facili da giocare rispetto alle AT tradizionali. A me personalmente però non piacciono più di tanto, ma è solo una questione di meri gusti: preferisco invece di gran lunga “l’odioso” parser che mi chiede tutte le volte cosa voglio fare, anche a costo d’arrabbiarmi e di spegnere il computer nei momenti di maggiore difficoltà. Lo so, sono un masochista ma non posso farci nulla...

Ti ritieni un purista delle AT (solo e rigorosamente solo testo) oppure credi che qualche immagine qua e là durante lo sviluppo di un'avventura contribuisca a creare l'atmosfera adatta per il giocatore?

Personalmente credo che la parte più importante di questo genere di giochi sia il testo; non mi ritengo comunque un purista in quanto ho apprezzato anche molte avventure testuali contenenti delle immagini e dei suoni, ma non le reputo tuttavia così fondamentali. Il giocatore, ripeto, si concentra principalmente sul testo e il resto di conseguenza è - almeno per come la vedo io - solo decorativo e di minore importanza.

Quanto la lingua inglese ha influito sulle tue scelte? Nel mio caso, è stato il motivo per cui ho iniziato ad imparare l'inglese, da assoluto autodidatta, negli anni '80, ma in giochi come quelli prodotti dalla Infocom la lingua era un duro scoglio per le mie conoscenze assolutamente precarie. Com'è andata per te?

Il mio percorso è stato abbastanza simile al tuo. Anch’io, come te, ho iniziato ad imparare l’inglese proprio grazie ad esse, ma a quei tempi quella lingua la conoscevo decisamente molto meno di adesso, e di conseguenza indirizzavo le mie scelte su quei giochi caratterizzati dall’aver un inglese semplice e piuttosto sintetico, come appunto le già citate avventure testuali di Scott Adams e quelle del bravissimo Brian Howarth (che, manco a farlo apposta, condividevano pure lo stesso parser).

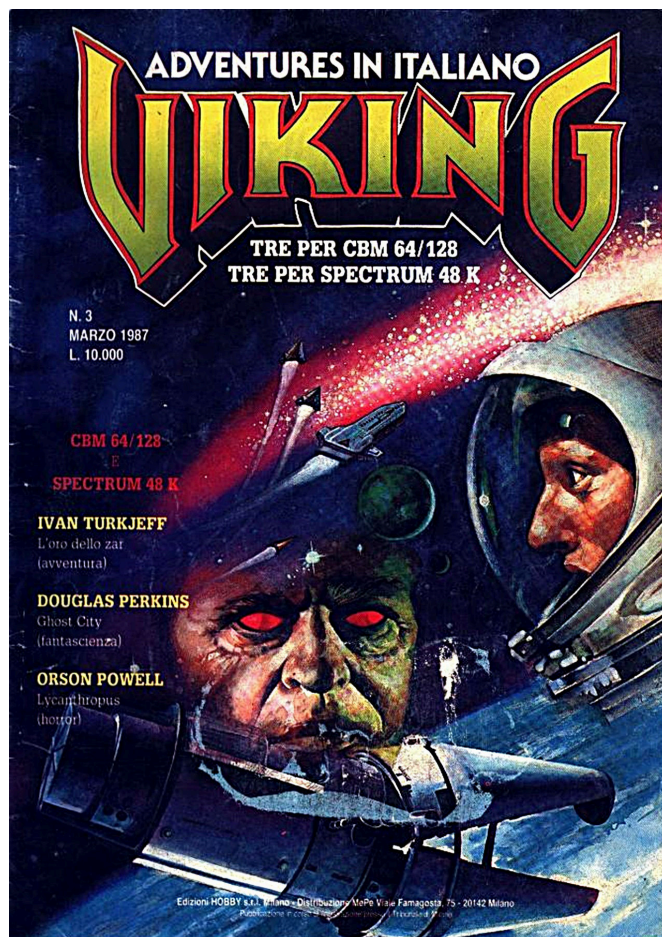
Per quanto riguarda invece i giochi della Infocom, anch’io purtroppo mi vidi ben presto costretto a gettare la spugna per via delle mie conoscenze precarie della lingua. Col senno di poi (e soprattutto dopo averne giocate nel tempo diverse) mi sono reso conto che alcune sono più semplici da comprendere, altre decisamente meno; in fin dei conti, gli autori che le scrivevano sapevano il fatto loro e spesso utilizzavano degli stili di scrittura particolari allo scopo di creare determinati “effetti letterari” al punto di arrivare molte volte ad usare dei termini slang. A chi volesse approfondire la storia della Infocom e delle avventure testuali in generale consiglio vivamente Get Lamp, un documentario sull’Interactive Fiction di Jason Scott.

Le avventure testuali possono secondo te essere relate in qualche modo ai romanzi?

Per me no: in un romanzo infatti, il lettore è uno spettatore passivo che si limita ad “osservare” lo scorrere degli eventi senza poter in alcun modo interagire con essi: in un’avventura testuale, al contrario, il lettore diventa il protagonista assoluto della storia e deve stabilire - attraverso tutta una serie di decisioni - come farla progredire arrivando molto spesso a dover tornare più volte in dei luoghi precedentemente visitati (cosa che in un romanzo non accade).

Va anche detto però che il romanzo così come lo conosciamo nel tempo si è evoluto fino a diventare anch’esso interattivo; in questo caso, abbiamo sì un punto di partenza, ma è poi il lettore a decidere come proseguire in un contesto di testo dinamico basato su una struttura narrativa non lineare. Anche così però, per me un’avventura testuale non è un romanzo, anche se mi sembra giusto precisare che si tratta anch’essa di una forma narrativa (e non solo un “gioco testuale” come alcuni sembrano amarla definire) in quanto sono comunque presenti degli elementi narrativi che vengono però definiti dalle decisioni del giocatore interagendo col parser, ovvero quella parte di codice del gioco che si occupa d’interpretare i comandi ricevuti per poi trasformarli - quando occorre - in azione permettendo così l’avanzamento della storia stessa. Ulteriori considerazioni sul parser potete trovarle in un articolo a cura di Stefano Gaburri apparso sul numero 2 di Terra d’IF - una fanzine italiana interamente dedicata all’Interactive Fiction.

Quali sono secondo te i più importanti creatori/sceneggiatori di AT, in Italia e nel mondo di oggi e di allora?



Copertina della rivista "Viking", per gentile concessione di Edicola 8 bit





Beh... allo stato attuale in Italia abbiamo avuto diverse persone veramente in gamba. Tra queste cito sicuramente Paolo Lucchesi e Francesco Cordella, autori rispettivamente di "La Pietra della Luna" e "Flamel", due vere pietre miliari del genere, senza dimenticare il buon Marco Vallarino, molto attivo ancora oggi con i due capitoli di Darkiss e non solo. All'estero non si possono non citare almeno tre veri mostri del genere, ovvero Andrew Plotkin (noto anche come Zarf), Adam Cadre e la bravissima Emily Short.

Nei tempi passati il mondo delle AT in Italia ha avuto, tra i vari protagonisti, Enrico Colombini, che con il suo Modulo Base ha permesso a moltissime persone di imparare a programmare un proprio gioco d'avventura, e il mitico Bonaventura Di Bello che, proprio grazie al Modulo Base in quegli anni ha potuto produrre la versione MSX delle sue avventure testuali, già sviluppate in precedenza sul Commodore 64 e sullo Spectrum usando i tool "The Quill" e "The Illustrator" della software house gallese Gilsoft. All'estero, cito almeno Steve Meretzky, autore, tra l'altro, di "A Mind Forever Voyaging", "Planetfall" e di "The Hitchhiker's Guide to the Galaxy", Scott Adams ("Adventureland" su tutte) ed infine Philip Mitchell, autore di "The Hobbit" e "Lord of the Rings" per conto della Beam Software, una sussidiaria della ben nota Melbourne House.

Da giocatore a programmatore: quando è avvenuto il passaggio e quali sono stati i tuoi primi passi?

La programmazione in sé mi ha attirato fin da ragazzino, quando passavo delle intere giornate a digitare i listati in BASIC che trovavo in giro sui libri e su alcune riviste dell'epoca (il nome "Papersoft" ti dice forse qualcosa?). I miei primi tentativi li feci con il Commodore 64 ma ben presto passai all'MSX, un sistema ben equilibrato e dotato di un BASIC davvero straordinario (almeno per l'epoca). Fermamente convinto di voler lavorare nell'informatica, nel 1998 mi diplomai anche come perito informatico ma le cose andarono poi diversamente e divenni - dopo tre durissimi anni d'università - un fisioterapista (e sono ben contento di esserlo tuttora, lo sottolineo).

Come hai scoperto Inform 6 e perché hai deciso di tradurre in italiano Ruins, uno dei giochi di esempio allegati al kit di sviluppo?

Ho scoperto Inform 6 tramite Google, quando un bel giorno decido di fare su di esso una ricerca con le parole "avventure testuali". Mi rendo ben presto conto che Inform 6 è uno strumento potentissimo, un linguaggio cioè orientato agli oggetti in grado - tramite il suo compilatore - di generare del file in codice per la Z-Machine, la macchina virtuale creata dalla Infocom.

Vivamente incuriosito, inizio a studiare il suo manuale ufficiale, il "The Inform Designer's Manual" di Graham Nelson, ma mi rendo subito conto che, pur essendo ben scritto, risulta essere troppo "condensato" in diversi punti. Decido così di provare a scriverne uno io utilizzando come base di partenza proprio Ruins, il gioco di esempio allegato al DM4, traducendolo prima in italiano con l'aiuto di un amico dell'epoca e solo dopo aver ovviamente chiesto il permesso a Nelson stesso, e il risultato lo si può

tuttora vedere su Wikibooks cercando "Inform e Glux".

Cosa pensi invece di Inform 7?

Inform 7 è un linguaggio naturale, e la sua elaborazione è diventata molto popolare negli ultimi anni in quanto permette a un calcolatore elettronico di trattare in automatico le informazioni scritte in una lingua naturale. Va da sé che questo strumento permette ad un autore di scrivere un gioco con una maggiore facilità rispetto a Inform 6, che prevede invece la programmazione imperativa tipica dei programmatori veri e propri, ma in Italia non sembra aver avuto molto successo in quanto sembra soffrire, fin dagli esordi, di un problema di localizzazione nella nostra lingua.

Personalmente non amo i linguaggi di questo tipo, ma invito chiunque voglia farlo a provarlo dal sito ufficiale ad esso dedicato. Ulteriori considerazioni su questo prezioso strumento potete trovarle sul sito di Lega Nerd.

In base alla tua percezione, la programmazione e la pubblicazione di un'avventura testuale ha ancora un senso o un mercato?

Un senso sicuramente sì, in quanto esistono ancora oggi delle nicchie di appassionati del genere. La lingua inglese va ovviamente per la maggiore ma si possono trovare diverse avventure testuali (sia vecchie che nuove) scritte anche in altre lingue, come ad esempio il francese, lo spagnolo e naturalmente anche l'italiano, come testimoniato dall'ultima edizione della Marmellata d'Avventura 2019. Sul discorso del mercato, la situazione non la vedo invece così rosea. Oggi il videogiocatore tipico è ormai abituato da tempo alla cosiddetta logica PlayStation, ovvero "inserisco la spina e ho tutto pronto all'uso". In un'avventura testuale di pronto non hai un bel nulla: devi scoprire da solo cosa devi fare, e spesso ti occorre un bel po' tempo per capire come risolvere determinati enigmi; tempo, che il videogiocatore tipico di oggi non è disposto più a spendere rispetto al passato, nonostante la presenza delle soluzioni che però - se usate troppo - ti tolgono il gusto del gioco in sé.

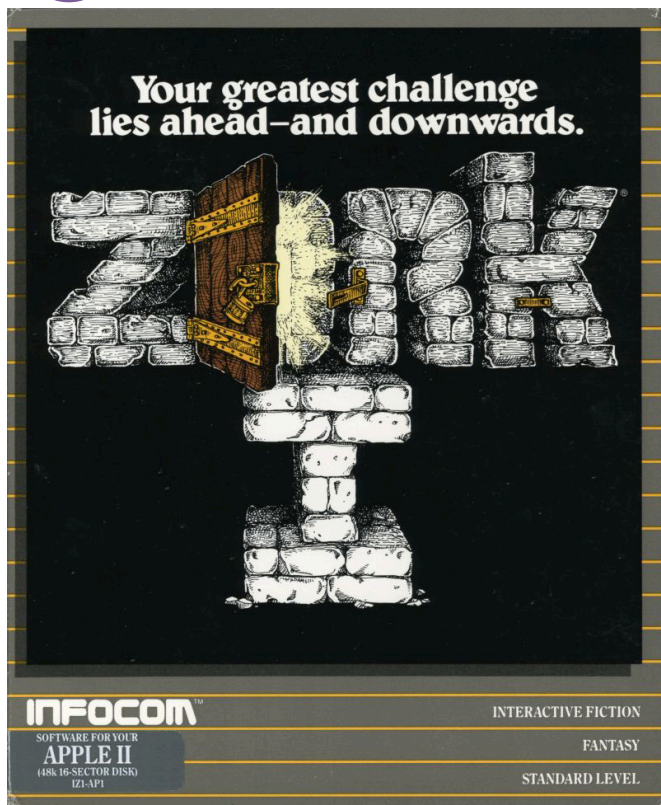
Senza contare che le avventure testuali tradizionali mal si prestano all'uso del mobile. Certo, è possibile provarle anche su un telefonino o su un tablet, ma come la mettiamo con le mappe senza le quali molte di esse non possono neanche essere giocate? E ve lo immaginate un tipico avventuriero che prende la metro, si siede e inizia a scartabellare dei fogli sparsi qua e là con un tablet appoggiato sulle ginocchia mentre nel frattempo una folla di persone lo distrae continuamente?

Io francamente proprio no e penso che questi giochi siano fatti e studiati per i computer da casa. E considerando che oggi si vendono più tablet che PC, non è così difficile secondo me trarre le debite conclusioni...

Alla luce delle tue considerazioni, esistono secondo te dei prodotti simili che potrebbero in qualche modo avere un mercato?

Che io sappia, allo stato attuale in Giappone sembrano aver avuto un buon successo le Visual Novel, un genere videoludico dove - parafrasando Wikipedia - il personag-





Copertina di "Zork I" per microcomputer Apple II

gio giocante può effettuare alcune decisioni che influenzano la trama del gioco in un ambiente caratterizzato da una grafica in stile anime e un'animazione limitata o assente. Tempo fa poi, Serious Joker (una vecchia guardia della comunità italiana dell'IF) mi ha fatto notare come sul sito americano Choice of Games sia possibile giocare a delle CYOA in inglese gratuite e non. Questi loro giochi hanno tra l'altro la caratteristica di tenere conto non solo delle singole scelte effettuate del giocatore, ma anche di come esse influenzino di fatto il carattere del personaggio che lui stesso interpreta, una caratteristica questa che può essere considerata sotto certi aspetti anche innovativa, almeno in questo genere di giochi.

Io stesso poi ho betatestato un prodotto molto interessante. Trattasi del "Segreto di Castel Lupu" un libro interattivo per bambini che coniuga le storie classiche dell'editoria per ragazzi al linguaggio moderno dei videogame, incoraggiando i lettori/giocatori al ragionamento e alla riflessione necessari per risolvere gli enigmi interattivi e portare a termine l'avventura.

Sempre parlando di libri interattivi, non si possono poi non citare le collane di Quintadiscopertina, una casa editrice fondata nel 2010 da Fabrizio Venerandi e Maria Cecilia Averame specializzata in editoria elettronica, con diversi testi di hypertext fiction.

Quali sono secondo te le caratteristiche che deve possedere un creatore/programmatore di AT?

Ovviamente deve saper scrivere in quanto, se è vero che da una parte c'è la possibilità di creare un'avventura testuale concentrandosi più sulla meccanica del gioco piuttosto che sulle descrizioni (situazione tipica degli anni ottanta) dall'altra oggi vanno per la maggiore quelle che prevedono invece una maggiore attenzione per queste ultime.

Occorre poi considerare che durante la creazione di un'av-

ventura testuale si devono rispettare alcune regole basilari. Sul numero 3 della rivista Terra d'IF è stato pubblicato al riguardo un articolo di C.E. Forman che consiglio vivamente di leggere a tutti coloro che vogliono per la prima volta crearne una.

In linea di massima, comunque, occorre non esagerare con le descrizioni dei luoghi, in quanto tutto ciò che è superfluo può confondere il giocatore stesso. Senza contare poi che gli enigmi, per quanto difficili possano essere, devono sempre seguire una certa logica, onde evitare che una difficoltà inutilmente eccessiva possa poi portare il giocatore ad abbandonare il gioco anzitempo, situazione tipica, questa, che si riscontra ad esempio nella creazione dei labirinti. Un altro consiglio è sempre quello di allegare nei vostri giochi un breve manuale delle istruzioni, soprattutto se al loro interno sono previste delle azioni meno comuni come ad esempio la possibilità di poter guidare un veicolo per spostarsi da una locazione all'altra, l'utilizzo di una particolare apparecchiatura in generale e così via. Alcuni autori poi, come lo stesso Bonaventura Di Bello, fanno notare come all'interno delle istruzioni sia importante scrivere anche una sorta di "trama introduttiva" che aiuti da una parte a inquadrare la storia e di conseguenza il tipo di contesto e di enigmi con cui ci si troverà a giocare, e dall'altra ad immergere il giocatore nella "storia". Inoltre, una certa importanza la riveste anche la capacità, in alcuni sistemi di sviluppo che lo prevedono, di scrivere una descrizione lunga ("verbose") e breve del luogo ("location") in quanto appunto alcuni sistemi di sviluppo visualizzano la seconda - come sintesi - quando il giocatore torna nello stesso luogo. Un'altra cosa assolutamente da non trascurare è la fase di betatesting. Una volta cioè che l'avventura è pronta, occorre farla testare ad altri giocatori (i betatester appunto) per poter così scoprire bug e problemi inerenti al parser, ed essere sicuri che il gioco sia il più possibile esente da errori prima del suo rilascio ufficiale. Ulteriori informazioni potete trovarle in un articolo di C.E. Forman apparso sul numero 5 di Terra d'IF.

Per concludere, quali AT consiglieresti di giocare a chi per la prima volta si avvicina a questo tipo d'intrattenimento?

In linea di massima è sempre meglio in questo caso giocare prima a dei tutorial, ovvero a delle avventure testuali studiate per coloro che si avvicinano per la prima volta a questo genere di giochi. In lingua italiana consiglio "Vil-la Morgana" di Paolo Lucchesi, mentre in lingua inglese potete provare l'ottima "The Dreamhold" del grande Andrew Plotkin.

Tornando alla lingua italiana e parlando adesso di avventure testuali "normali", potete poi provare le già citate "Flamel" e "La Pietra della Luna", ma anche Beyond, che nel 2005 si è classificata al secondo posto della 11° Annual Interactive Fiction Competition (IFComp), e ha vinto i premi per la miglior storia (Best Story) e il miglior enigma (Best Individual Puzzle) dell'edizione 2005 degli XYZZY awards. In lingua inglese, oltre a "Colossal Cave", la prima avventura testuale della storia, "Mystery House", la prima avventura testuale grafica della storia, e a dei grandi classici della Infocom come "Planetfall", "The Hitchhiker's Guide to the Galaxy" e "Zork", potete pro-





vare anche “Anchorhead” di Michael S. Gentry, ispirata ai miti di Cthulhu di H. P. Lovecraft, Galatea di Emily Short ed infine “Varicella” e “Photopia”, scritte entrambe dal geniale Adam Cadre.

Ricordo infine - per chi non lo sapesse - che se per giocare alle avventure testuali “moderne” è sufficiente in genere installare un apposito interprete, ad esempio Winfrotz su Windows (per Linux è disponibile per praticamente tutte le distribuzioni ndr) per le avventure scritte nel formato Z-code, per quelle più vecchie occorre invece utilizzare un emulatore, ovvero un programma che replica le funzioni di un determinato sistema, come ad esempio il Commodore 64, lo Spectrum o l'Apple II, su un secondo sistema differente dal primo come Windows, Linux, Mac OS, Android e molto altro ancora. Ulteriori informazioni al riguardo potete trovarle su degli appositi siti che li raccolgono come ad esempio EmuFrance, EmuCR, ed il mio, EmuWiki. Se si è invece alla ricerca delle avventure testuali vere e proprie, consiglio sempre di visitare - per quelle inglesi - i siti The Interactive Fiction Database, The Interactive Fiction Archive e soprattutto The Interactive Fiction Wiki. Per quelle italiane un ottimo punto di partenza è sicuramente OldGamesItalia e, a seguire, i portali dove sono archiviate e rese disponibili le riviste su cassetta Edicola 8 Bit, Edicolac64.com e Ready64.org.

Ringraziamo Vincenzo per la sua disponibilità e per averci dato un'occasione in più per esplorare l'affascinante mondo delle AT. I lettori che volessero approfondire l'argomento, possono recuperare i numeri 2, 7, 8, 12 e 16 di RetroMagazine, oltre a seguire i link indicati da Vincenzo, che trovate riepilogati nel riquadro in questa pagina. Ma ovviamente, il modo migliore per farlo resta sempre quello di giocare, magari approfittando dell'obbligo di restare in casa a causa dei disgraziati eventi di questi giorni, che speriamo di lasciarci presto alle spalle con il minor numero di danni possibile, per cui fatevi sotto e buona avventura!

Riferimenti

- Terra d'IF
<http://ifitalia.oldgamesitalia.net/pmwiki/pmwiki.php?n=Riviste.TerraDIf>
- Inform e Glulx
https://it.wikibooks.org/wiki/Inform_e_Glulx
- Inform 7
<http://inform7.com/>
- Inform 7 su Lega Nerd
<https://leganerd.com/2013/10/19/inform-7-programmazione-linguaggio-quasi-naturale-avventure-testuali/>
- Marmellata d'avventura
<http://www.oldgamesitalia.net/content/marmellata-davventura-2019>
- Visual Novel su Wikipedia
https://it.wikipedia.org/wiki/Visual_novel
- Castel Lupo
<http://www.castellupo.it/>
- Quinta di copertina
<http://www.quintadicopertina.com/>
- 10 Suggerimenti per un ottimo 'Game Design'
<http://ifitalia.oldgamesitalia.net/pmwiki/pmwiki.php?n=Riviste.TerraDIf>
- Oh! No! Betatest!
<http://ifitalia.oldgamesitalia.net/pmwiki/pmwiki.php?n=Riviste.TerraDIf>
- Villa Morgana
<http://www.paololucchesi.it/at/villa.html>
- Beyond
<http://www.paololucchesi.it/at/beyond.html>
- Frotz per Windows
<http://www.davidkinder.co.uk/frotz.html>
- EmuFrance
<http://www.emu-france.com/>
- EmuCR
<https://www.emucr.com/>
- EmuWiki
<https://www.vincenzoscarpa.it/emuwiki>
- The Interactive Fiction Database
<https://ifdb.tads.org/>
- The Interactive Fiction Wiki
http://www.ifwiki.org/index.php/Main_Page
- OldGamesItalia
<http://www.oldgamesitalia.net/if-italia-avventure-testuali>
- Edicola 8 Bit
<https://www.edicola8bit.com/>
- Edicolac64.com
<http://www.edicolac64.com/>
- Ready64.org
<http://ready64.org/>





RETROEVENTI DEL 2020

Videogame Art Museum – Bologna

Un museo dedicato ai videogiochi?? Si ora c'è... si trova a Bologna ed è stato aperto il 2 febbraio 2020 grazie all'associazione Insert Coin che si occupa di divulgare la cultura del videogioco e preservare la memoria storica dell'arte videoludica.

Il museo sarà aperto dalle ore 16 alle ore 19 nei locali del MAST, in via Vittoria 28/G nella zona chiamata "Pontelungo" facilmente raggiungibile in auto oppure con il servizio bus.

All'interno delle mura troviamo console rare, prototipi, home computer, videogiochi storici e tanti gadget e memorabilia.

Attraverso dei pannelli informativi, video, curiosità e vetrine si può scoprire la storia delle prima otto generazioni di console e home computer e dei videogiochi. Il museo inoltre comprende una sala con videoproiezioni sulla storia del videogioco e un'area "GIOCO LIBERO" con tante postazioni di gioco gratuite.

L'associazione INSERT COIN propone un ingresso gratuito ma con un primo tesseramento di 7 euro, che andranno nelle casse associative per migliorare i locali e aggiungere perle e collezioni.

Nel corso degli anni saranno inserite diverse iniziative retroludiche, ospiti e eventi.

Se passate da Bologna non vi resta che farci un salto.

Bologna Nerd Show

Piccola premessa personale, ma proprio personale!!

Non amo i cosplayer... non ce la faccio! Non ci riesco... è qualcosa più forte di me. Che vogliamo fare?!?!?

Hanno assaltato ogni manifestazione di comics, gdr e videogames d'Italia (il resto del mondo lo avevano già mangiato) peggio delle orde di the Walking dead. Li vedo ovunque... bellissimi costumi, bellissime ragazze... ma non tollero gli spettacolini, i momenti "tenerosi" e tutto il resto.

Finito... ohhhh!! Devo sfogarmi. Non

vado più a Lucca per queste ragioni, sono più settoriale.

Purtroppo il Nerd Show bolognese ne è bello pieno. In tutte le salse, dai costumizzati, a chi canta a chi fa persino wrestling... massimo rispetto per tutto... ma immaginatevi ciò che penso io!!!

Passiamo alla parte che più ci interessa l'area retrogames e retro computers.

Diverse decine di postazioni di gioco delle più svariate console dove provare tutti i successi del passato a 8 e 16 bit, alcune chicche come il Vectrex o il Pc Engine, zona cabinati da sala ampia con un po di tutto, dalle riedizioni ai restauri e una serie di conferenze dedicate alla programmazione su macchine Commodore Amiga.

Notevole la presenza di alcuni esponenti della demoscene proprio su Amiga, che si presenta sempre viva e di grande impatto.

Purtroppo non sono riuscito a seguire le conferenze sulla programmazione.

Lo spazio vendita come sempre presenta una presenza massiccia di espositori con le più svariate versione di macchine e giochi. Ideale per chi vuole rimpolpare la propria collezione.

I prezzi?? Alti per le versioni boxate, anzi altissimi. Siamo in quella fase di estrema "moda" dove tutto ciò che fino a 10 anni fa veniva buttato via o poco considerato ora vale più dell'oro. Per fortuna per i meno esigenti (tipo il sottoscritto) tra gli espositori erano presenti centinaia di versione loose di giochi e console.

Ora vi lascio alle foto di rito.

P.S. Il giudizio sulla fiera? E' molto soggettivo, io preferisco cose differenti con maggiore enfasi su cosa si può fare piuttosto che sull'esposizione in se, quindi ad ognuno il proprio pensiero.

di **Carlo Nithaiah Del Mar Pirazzini**

Videogame Art Museum – Bologna



Bologna Nerd Show



Riflessioni in tempi di COVID-19

In questo marzo 2020 non è facile parlare di retrocomputing come l'abbiamo sempre fatto, con la spensieratezza e la nostalgia di quei ragazzi che, stando alle più avanzate teorie della psicologia moderna, sopravvivono dentro di noi. Un evento epocale e mai verificatosi prima ci costringe a riconsiderare le nostre abitudini, le nostre convinzioni ed il nostro stile di vita.

Ma proprio in questi giorni ho conosciuto una persona sui social network che mi ha fatto riflettere contemporaneamente sulla caducità della nostra esistenza e sull'enorme forza di volontà di cui siamo capaci noi bipedi, esseri umani di tutte le latitudini.

Alex è un programmatore Java, padre di famiglia e non vedente. Da ragazzo ci vedeva benissimo e, come è accaduto a molti di noi, il papà gli comprò un MSX VG-8010 e poi un Amiga, coi quali ha passato tante ore di divertimento giocando e imparando a programmare. Dopo essersi trovato al buio non ha perso la voglia di fare e non ha abbandonato la sua passione per l'informatica, tanto che è riuscito a farne una professione, grazie ad ausili specifici per PC oltre che alla sua ferrea volontà.

Dopo tanti anni si è ricordato della sua antica passione per le macchine a 8/16-bit e, tramite un amico, gli ho procurato proprio un MSX VG-8010, lo stesso modello che Alex possedeva e che, dopo un guasto, era rimasto presso il negozio di riparazione. Mostrerà a suo figlio gli stessi giochi per MSX che lui amava e poi proverà a programmarlo tramite un software di sintesi vocale, perché il suo obiettivo è trovare un modo per utilizzare i vecchi computer a 8 e 16 bit anche per tutti coloro che condividono l'handicap alla vista. Oltre all'MSX, presto gli farò avere un C64 che, con la disponibilità del software SAM/Reciter si presta forse meglio di altri sistemi alla programmazione di funzioni Text-To-Speech.

Ecco, in un momento di profonda incertezza qual è quello che viviamo ora, il cambio di prospettiva che ci offre Alex con la sua energia e con i suoi progetti è proprio quello che ci vuole per rimettersi in gioco e affrontare un mondo totalmente nuovo che confina con la nostra passione per i retrocomputer. Inutile dire che chiunque voglia aiutare Alex nel suo proposito di usare i vecchi computer come non vedente è assolutamente il ben accetto, con idee, suggerimenti, esperienza diretta e (parola chiave) passione.

Scrivete pure a retromagazine.redazione@gmail.com

David La Monaca / Cercamon

Disclaimers

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

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

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

Attribuzione

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

NonCommerciale

Non puoi utilizzare il materiale per scopi commerciali.

StessaLicenza

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

Divieto di restrizioni aggiuntive

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



RetroMagazine
Anno 4 - Numero 21

Direttore Responsabile
Francesco Fiorentini

Vice Direttore
Marco Pistorio

'MARZO 2020'

