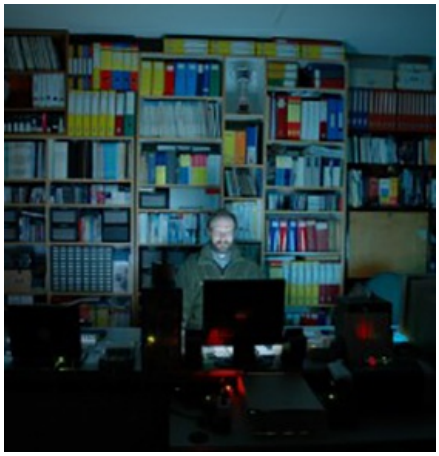
Intervista a Enrico Colombini (3)

(http://www.retrogamingplanet.it/intervista/enrico-colombini-unavventura-infinita-nella-storia-informatica-di-ieri-e-di-oggi/)



Come accade in tutti i campi professionali anche in quello videoludico esistono persone che, dopo aver creato titoli entrati nella storia, si ritirano dalle scene continuando però a lavorare "dietro alle quinte..."

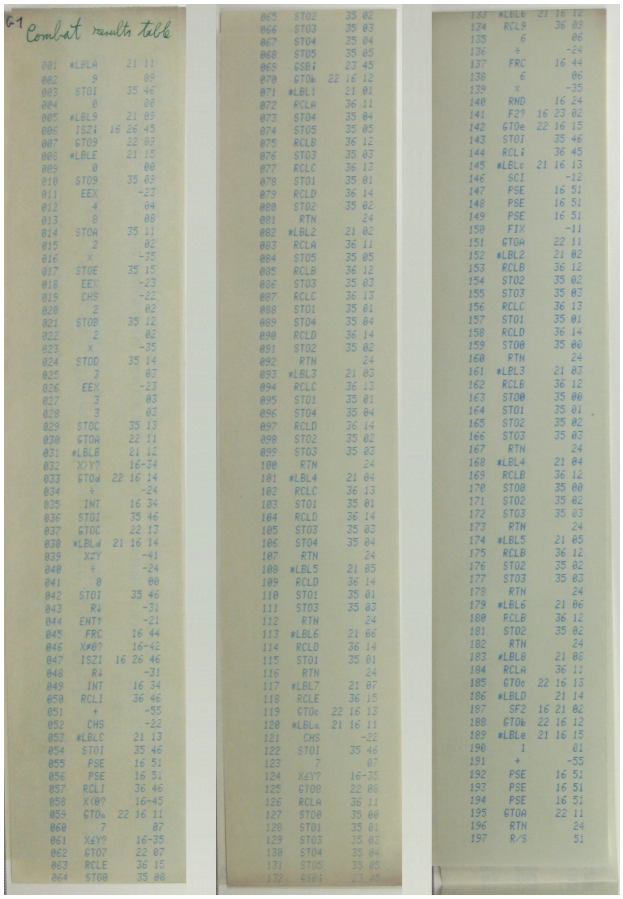
Enrico Colombini è proprio uno di questi: dopo aver pubblicato l'avventura testuale **Avventura nel Castello** nei primi anni '80 ha continuato a lavorare nell'ombra fino ad arrivare ai giorni nostri con la pubblicazione di un nuovo prodotto che andremo ad analizzare nel corso di questa interessante intervista!

RGP: Ciao Enrico e grazie per il tempo che stai concedendo a tutti i lettori di Retrogaming Planet!
La tua riservatezza e discrezione sono note nell'ambiente videoludico pertanto ritengo questa intervista davvero imperdibile per tutti gli appassionati di videogames e retrogaming! Tralasciando la solita domanda "anagrafica" entriamo subito nel vivo dell'intervista: quando hai messo per la prima volta le mani su un computer e per quali scopi?
ERIX: Definisci "computer" 🙂
Tralasciando curiosi dispositivi come la macchina a intelligenza artificiale fatta con scatolette di fiammiferi (costruita seguendo le istruzioni di **Martin Gardner** in "Giochi matematici" su Le Scienze), credo che la mia prima macchina elettronica programmabile sia stata la calcolatrice tascabile **HP-55**, nei primi anni '70.
Il primo computer nel senso moderno del termine fu il **PET 2001** nel 1978, anche se penso di avere messo le mani due-tre anni prima sul **P6060** dell'università: fu il mio primo incontro col BASIC).

RGP: Ricordi la configurazione del tuo primo computer?
ERIX: La HP-55 aveva 49 "passi di programma", insomma non poteva contenere più di 49 istruzioni, e non aveva subroutine, quindi occorreva ingegnarsi per scrivere programmi non banali. Inoltre non era possibile registrare i programmi da nessuna parte, né ovviamente stamparli.
L'indistruttibile **HP-97** del 1976 poteva fare entrambe le cose (la registrazione avveniva su schedine magnetiche) per cui i primi listati HP che conservo sono di quell'epoca.
Per quanto riguarda invece il PET 2001, non ricordo quanta RAM avesse (mi pare 8 KB), ma ricordo benissimo che aveva un difetto di progetto del software di sistema: il registratore a cassette era alquanto inaffidabile. Il nostro esemplare aveva inoltre un tasto P che tremolava toccando il case... Commentai *"Se fosse la B sarebbe un tasto Budino"* e Chiara replicò *"Ma no, è giusto così: P sta per Pudding"*.
Comunque il PET si usava volentieri e la tastiera rispondeva bene nonostante fosse ultraeconomica; la semigrafica fatta con i caratteri era la stessa che poi sarebbe stata usata per il Commodore 64.

RGP: Ho notato che in tutte le interviste rilasciate non ti è mai stato chiesto quali linguaggi di programmazione conosci ma soprattutto come hai imparato a programmare! Ci puoi raccontare qualcosa in merito?
ERIX: I linguaggi che conosco sono tanti, ma in genere tendo a usare il C (e derivati) per il massimo controllo a basso livello e il Lua per quello che sta sopra, ma ovviamente dipende dall'ambiente.
In pratica la difficoltà non sta tanto nell'imparare un nuovo linguaggio (cosa che, avendone visti un po', si può fare "per differenza" in pochi giorni se il linguaggio non è Haskell), quanto nello studiarsi le librerie che diventano sempre più vaste... e sempre più burocratiche e costrittive, essendo progettate per forzare i cattivi programmatori su un percorso

(pratico e mentale) recintato dove non possono fare troppi danni, anche se ciò impedisce a un buon programmatore di operare in modo efficiente.
Riguardo a come ho imparato, dopo le calcolatrici HP frequentai due scuole fondamentali: la prima fu la programmazione in assembly 6502 sul **KIM-1** (dal 1976) e su dispositivi elettronici; facevo il progettista indipendente e mi comprai appunto un KIM-1 per studiare quei nuovi strani aggeggi, i "microprocessori".
All'inizio, per programmare un dispositivo si facevano un po' di prove e poi si scriveva il codice su carta, lo si "assemblava" con gomma e matita (cito prima la gomma perché era molto usata...) e lo si registrava su EPROM. Se qualcosa andava storto, ci volevano 20 minuti di cancellazione con gli ultravioletti e non si potevano tenere molte EPROM perché costavano care, per cui si imparava in fretta a stare attenti. E anche dopo, con un assembler e un buon debugger a livello di CPU, imparai parecchio, sia con la pratica che con l'aiuto dei pochi libri che si riuscivano a trovare.



Il listato del primo 'gioco' per la HP-97: un lancio di dadi...

La seconda scuola, qualche anno dopo, fu il Basic (o meglio BASIC con la grafia dell'epoca) del PET 2001 e dell'Apple II: stringhe dinamiche, garbage collection... tanti concetti nuovi e tanti nuovi modi di sbagliare. Aggiungo che la "programmazione strutturata", conosciuta attraverso un libro di **Wirth**, mi fu molto utile: l'applicai subito in entrambi i linguaggi.
Fu nei primi anni '80 che incontrai il C e fu amore a prima vista. Dovevano passare vent'anni prima che trovassi qualcosa di altrettanto sintetico e geniale (il Lua, appunto)!

RGP: In Italia sei conosciuto quale creatore, nel 1982, della prima avventura testuale completamente in italiano: Avventura nel castello per Apple II. La domanda quindi sorge spontanea: la tua passione per le avventure ha solide fondamenta nei classici giochi di ruolo da tavolo oppure si è sviluppata direttamente su computer?
ERIX: Solidissime, direi. Anche se l'ispirazione per Avventura nel castello fu la **Colossal Cave** (in versione "Apple Adventure"), sono sempre stato appassionato di giochi da tavolo e, più tardi e in misura minore, anche di giochi di ruolo. Ma non dimentichiamo quella che è forse la fonte principale: le buone vecchie letture; fantasy e avventura, a partire dai romanzi di Verne e Salgari fino a Pratchett, ma anche narrativa in genere.

RGP: Qual è stata la "molla" che ti ha spinto a realizzare un'avventura per computer?
ERIX: La spinta iniziale, dopo aver giocato ad Apple Adventure, fu *"Vediamo se riusciamo a fare qualcosa del genere, o magari anche un po' meglio!"*. Poi, visto che la cosa sembrava funzionare, si alimentò da sé con la soddisfazione del creare qualcosa di nuovo (che è sempre stata la mia molla principale e che mi si ritorce contro in questo periodo di immobilismo generale).

KIM-1 verde prima serie

RGP: Quale linguaggio hai utilizzato per realizzarla?
ERIX: Due linguaggi che conoscevo bene: **Basic** e **Assembly 6502**. Per la verità, l'unica parte che realmente necessitava di essere scritta in assembly era il sistema anticopia; l'altra, la consultazione del dizionario, avrebbe potuto benissimo essere fatta in Basic... ma non conoscevo la ricerca binaria e quindi lavorai più del necessario (delitto assai grave!).
In seguito ho fatto un po' di studi per interpreti di avventure con i linguaggi più vari: C, Pascal, Prolog, Perl, Lua, Bash (script), TCL, Dylan e non ricordo più quanti altri. Le conclusioni più importanti sono state che i linguaggi dinamici (es. Lua, Perl, Python, Javascript) sono di gran lunga più pratici per questo scopo e che il concetto di compilazione, sempre in questo ambito, mi pare non abbia senso perché conviene usare semplicemente un interprete; anche la catalogazione sintattica degli elementi della frase mi lascia molto dubbioso. Insomma, una posizione molto distante da quella "ufficiale" che predilige strumenti metodici come Inform.

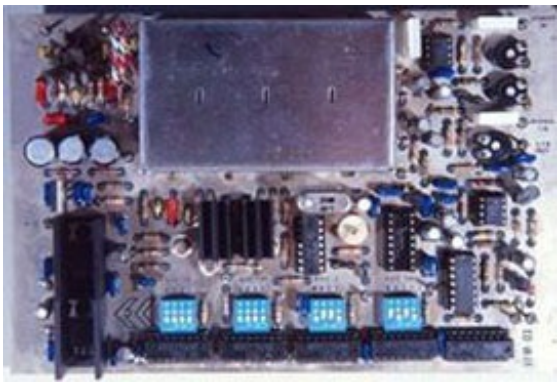
RGP: La leggenda narra che Avventura nel castello venne distribuita assieme agli Apple II venduti da un tuo amico negoziante (grazie ad un accordo reciproco)... confermi questa versione?
ERIX: Quasi... era la fine del 1982 e gli amici de "Il computer" (così si chiamava il negozio) accettarono gentilmente di metterla in vendita. Non era distribuita assieme agli Apple II, cosa che invece avvenne più tardi grazie a un accordo tra **J.Soft** ed **Apple Italia**.

RGP: Quale target di utenza ti eri prefissato di raggiungere con il tuo primo prodotto?
ERIX: Target? Ho la faccia di un venditore? 🙂
Provai per vedere cosa sarebbe successo, senza averne la minima idea.

RGP: Ti ritieni soddisfatto del risultato ottenuto e quali modifiche apporteresti, con il senno di poi, al gioco?
ERIX: Tutto sommato non toccherei niente: non penso sia saggio ritoccare o ripensare un'opera in un periodo diverso da quello nel quale è stata concepita. Naturalmente ci sono tanti piccoli dettagli che uno migliorerebbe (o peggiorerebbe?) ma se potessi tornare al 1982 e darmi (anzi darci, non dimentichiamo il contributo di Chiara Tovena) un consiglio, direi: va bene così.
Naturalmente oggi non scriverei un'avventura nella quale sia possibile mettersi in un vicolo cieco distruggendo un oggetto necessario per la soluzione, ma all'epoca era la norma!

RGP: Dalla pubblicazione di Avventura nel castello (convertita in seguito anche per sistemi MS-DOS) la tua carriera è stata un susseguirsi di successi, in campo editoriale e videoludico... puoi indicarci qualche tuo prodotto del quale vai particolarmente fiero?
ERIX: Vostro Onore, dopo una decina d'anni di occupazione precaria e false partenze, con grande difficoltà a trovare qualcosa d'interessante da fare (un mio difetto è di essere un innovatore che non sopporta la banalità e il tirare a campare tipico del nostro Paese, situazione che purtroppo si va aggravando) tenderei a sollevare qualche obiezione a proposito del "susseguirsi di successi"!
In ogni caso potrei citare tre lavori complessi di cui mi sento di andare partcolarmente fiero, al di là delle opere divulgative da edicola: il primo, di cui penso nessuno abbia sentito parlare, è un progetto elettronico della seconda metà degli anni '70: il VFM-03. Si trattava del "cuore" di un trasmettitore FM a controllo digitale a sintesi diretta, che fornivo ad alcuni tra i principali costruttori di impianti per le radio "libere" e che contribuì a darmi da mangiare e a consentirmi di comprare i primi computer. Già allora c'erano varie aziende che me lo copiavano e producevano cloni, ma funzionavano peggio 🙂
Il secondo progetto, altrettanto ignoto anche se distribuito in milioni di copie nei primi anni '90, fu Igloo: un ipertesto grafico vagamente ispirato a HyperCard del Macintosh, nato per essere efficiente e veloce su MS-DOS con 640 KB e

costruito quasi "on the metal" ossia accedendo direttamente all'hardware con un uso minimale del sistema operativo: driver grafico ottimizzato in assembly (FG: Fast Graphics), interfaccia a finestre (WNG: Windows? No, Grazie), player (Igloo: Ipertesto grafico per Libri, Object-Oriented) e un editor grafico per prepararne i contenuti. Lo usai per la parte interattiva del mio corso "PC Subito" che ebbe un notevole successo e mi consentì anche di comprarmi la casa. Ironicamente, era un corso di Windows presentato con un tool MS-DOS e scritto con un Macintosh!
A dimostrazione del fatto che non tutti i progetti complessi ben riusciti sono necessariamente dei successi economici, faccio un salto avanti di altri vent'anni per arrivare al presente: il terzo lavoro che cito è l'ebook-gioco "Locusta temporis", del quale sono molto soddisfatto sia per risultato ludico-narrativo, sia per le soluzioni tecniche (tanto all'interno del racconto che nei meccanismi di sviluppo), anche se mi è costato un lavoro immane.



Uno dei progetti elettronici di Enrico negli anni '70...

RGP: Se potessi tornare indietro nel tempo rifaresti tutto o cambieresti qualcosa della tua carriera (editoriale e videoludica)?
ERIX: A parte l'emigrare in Europa negli anni '80, intendi? Dal lato editoriale non credo che cambierei molto, salvo forse il venirne fuori con qualche anno di anticipo, quando era purtroppo chiara la china discendente sulla quale il settore si trovava. Dal lato videoludico, proprio non saprei... se fin dagli inizi mi fossi dedicato ai videogame (inevitabilmente fuori d'Italia) mi sarei senz'altro divertito, ma non avrei lavorato nella divulgazione.
A me piace scrivere, specie di argomenti concreti; insegnare a fare o a capire qualcosa mi dà l'impressione di contribuire con la mia briciolina a un cambiamento in meglio, il che è molto più di quanto si possa ottenere facendo giochi, almeno nel senso in cui i videogame sono intesi oggi (qui ci sarebbe da aprire una lunga parentesi sulla differenza tra giochi e film interattivi, ma non voglio dirottare l'intervista).

RGP: Sul tuo sito è presente una gran quantità di materiale interessantissimo: numerosi software applicativi, tutte le avventure testuali da te realizzate per sistemi Apple e MS-DOS ed Idra, un linguaggio di programmazione per la creazione di avventure. Per quanto riguarda i primi leggiamo che essi, pur essendo coperti da copyright, sono liberamente distribuibili... questo significa che chiunque può metterli in download sul proprio sito/blog e distribuirli liberamente citando la fonte originaria?
ERIX: Sì, a quella condizione senz'altro. Ah, non rigirare il coltello nella piaga con la "gran quantità": ho tanto di quel materiale che dovrei riordinare e mettere in linea...

RGP: Speriamo allora avvenga al più presto!! Riguardo invece al linguaggio di programmazione sopracitato ci racconti com'è nata l'idea e per quale motivo hai deciso di creare un linguaggio tutto tuo per la realizzazione di avventure testuali?
ERIX: Una piccola precisazione tecnica: Idra non è un linguaggio ma una libreria Javascript: in pratica si tratta semplicemente di una serie di funzioni aggiuntive; quello che l'autore scrive è un programma Javascript. Questa impostazione ibrida semplifica da una parte il lavoro... e lo complica da un'altra, specie dal punto di vista degli errori di sintassi: i problemi accidentali con le virgolette sono causa di gran parte delle richieste di aiuto che ricevo.
L'idea di Idra nacque quando mi posi una domanda: la gente oggi non gioca alle avventure testuali perché non ha voglia di leggere, oppure perché non ha voglia di scrivere? Un modo per cercare la risposta era quello di fare avventure che richiedessero di leggere ma non di scrivere, un po' come i vecchi libri-gioco ma con la versatilità consentita da un computer. E così nacque Idra! Quanto a una risposta univoca, non la so ancora oggi; o meglio, mi pare che siano presenti entrambi i fattori e che la poca voglia di scrivere sia prevalente.

RGP: Hai mai utilizzato Idra per scrivere un tuo gioco?
ERIX: No, e questa è una mia grave colpa. Ma per scrivere e documentare Idra avevo esaurito la quantità di energia disponibile, per cui mi limitai a "tradurre" due simpatici libri-gioco di Andrea Angiolino. Non credo che lo strumento sia mai stato sfruttato in tutte le sue potenzialità, almeno in base a ciò che ho avuto modo di vedere.

RGP: Oltre alle avventure testuali, da qualche tempo vediamo comparire il tuo nome, in qualità di "Additional code", in alcuni giochi realizzati dalla nostrana Artematica di Riccardo Cangini (Ciao Ricky!): Subbuteo, ICO Soccer e Diabolik Original Sin vantano infatti la tua presenza fra i Credits! Ci racconti come è nata la collaborazione con Artematica?
ERIX: È molto semplice: visto lo stato comatoso dell'editoria, decisi di provare a entrare nel mondo dei videogame. Studiai un po' e mi guardai in giro: Artematica era l'unico developer italiano che facesse avventure, per cui iniziai a stressare il buon Cangini finché, pur di liberarsi della mia perennne scocciatura, mi diede l'opportunità di fare qualcosa con loro, sia pure come "coder" e non come autore o sceneggiatore.
In questa veste di programmatore lavorai quindi non solo alle avventure, ma anche ad altri progetti...

RGP: Cosa si prova a lavorare su prodotti cosi diversi dalle avventure alle quali hai dedicato gran parte della tua vita?
ERIX: Se devo essere sincero, per me i giochi di calcio sono una noia mortale. Tuttavia possono presentare interessanti problemi tecnici, come ad esempio memorizzare la situazione di un intero campionato in **20 byte** (lo so che in teoria sembra impossibile, ma date certe condizioni...) ma soprattutto, il lavoro in team mi ha dato l'opportunità di conoscere e apprezzare persone di grande valore e simpatia! (*Su quest'ultima affermazione di Enrico concordo in pieno avendo avuto più volte la fortuna di chiaccherare con i vari Riccardo Cangini, Massimiliano Calamai, Natale Fietta, Daniele Montella ed altri collaboratori di Artematica... ed alcuni di loro conoscerli anche personalmente! (**NdRGP**)*

RGP: In particolare un gioco fra quelli sopra citati ha colpito la mia attenzione: Diabolik Original Sin e l'engine GALE, da te realizzato e ottimizzato per Nintendo DS! Cosa ci puoi dire in proposito?
ERIX: Si tratta del porting per DS della versione PC di Diabolik Original Sin ma, vista la significativa differenza tra le due architetture, in pratica il codice è stato completamente riscritto basandolo su librerie DS sviluppate internamente da Artematica ad opera dell'abile e imperturbabile lead programmer, **Natale Fietta**.
GALE (Graphic Adventure Lua Engine) è un interprete per avventure, ossia il 'motore logico' del gioco, che include un linguaggio nel quale possono essere programmate in modo semplice le azioni, animazioni, risposte, eventi e via dicendo. In questo modo gli scripter, cioè coloro che trasformano la sceneggiatura in codice, possono operare con un linguaggio molto meno impegnativo del C++, un linguaggio alla portata di un non-programmatore dotato di buone capacità logiche.
Altrettanto importante è il fatto che questo lavoro viene completamente separato dal codice vero e proprio e non si richiede quindi che quest'ultimo venga ricompilato a ogni variazione: si separano i compiti, si risparmia parecchio tempo nello sviluppo e si evita di introdurre errori.
A proposito di risparmiare tempo (noi bradipi siamo abili nella pigrizia costruttiva!), un altro lavoro che ho trovato interessante è la realizzazione di tool e l'ottimizzazione della catena di produzione (toolchain), ossia di tutte le operazione necessarie per arrivare dagli 'asset' grezzi (immagini, suoni, ecc.) fino al programma eseguibile. Automatizzare il tutto e minimizzare il tempo che intercorre tra una modifica a un disegno e il test nel gioco consente di operare in modo più rapido ed efficiente.
Insomma di lavorare meno, che è sempre un nobilissimo obiettivo!
A proposito di efficienza, devo citare il fondamentale contributo del nostro Product Manager **Marco Ponte**, che ha saputo far funzionare come un orologio i team delle avventure, risolvendo problemi organizzativi di ogni genere e mantenendoci "on schedule", ossia in perfetto orario, quando non addirittura in anticipo. Non posso dilungarmi a citare tutti i partecipanti, ma era davvero un *Dream Team*.

RGP: Lua Code... un nome curioso ed interessante! Di cosa si tratta?
ERIX: Si tratta semplicemente di codice scritto in linguaggio Lua. È un linguaggio particolarmente adatto ad essere inserito (embedded) all'interno di programmi C o C++ ed è ideale per la creazione di linguaggi su misura (DSL, Domain-Specific Language). Essendo anche assai espressivo e particolarmente compatto, non sorprende che sia molto usato nei giochi; l'esempio più noto è probabilmente **World of Warcraft**, ma sono in molti a farne uso... a partire dal buon vecchio **Grim Fandango** (grandissima avventura!).

RGP: E come darti torto...? Tornando alla tua produzione nel 2010 hai pubblicato l'Ebook Game Locusta Temporis (del quale potrete scaricare gratuitamente il Demo o acquistare una delle versioni disponibili collegandovi a questo indirizzo); vuoi parlarci di questo tuo nuovo prodotto?
ERIX: Per la precisione, io l'ho scritto e *quintadicopertina* l'ha pubblicato. Precisazione non trascurabile, perché Fabrizio Venerandi e Maria Cecilia Averame hanno dimostrato coraggio e intraprendenza scegliendo di fare i pionieri con questa Casa editrice di qualità, dedita ad esplorare le nuove potenzialità offerte dagli ebook: non passa giorno che non inventino qualcosa di innovativo, come si può vedere dalla ricchezza e varietà di proposte originali su **http://www.quintadicopertina.com**.
Ma torniamo alla Locusta, come la chiamo affezionatamente; mi è difficile classificarla con i criteri esistenti, perché penso

si tratti di una forma narrativa nuova per un ebook. È una narrazione interattiva di più ampio respiro rispetto alle mie vecchie avventure, nella quale gli elementi di gioco fanno parte della vicenda stessa; entrambi gli aspetti sono importanti, anche se inizia in modo simile a un gioco d'avventura e finisce come un racconto.
Io la vedo come un fumetto o un'avventura grafica con la trama particolarmente curata... e senza disegni.
La realizzazione di quest'idea in forma di ebook ha posto dei problemi particolari, ed è qui che sta l'innovazione: se in un'avventura (grafica o testuale che sia) posso usare un linguaggio di programmazione e le sue variabili, in un ebook ho solo a disposizione un insieme di pagine finite e immutabili, 'stampate' una volta per tutte e collegate tra loro solo dagli hyperlinks tra i quali il lettore può scegliere.

Locusta Temporis

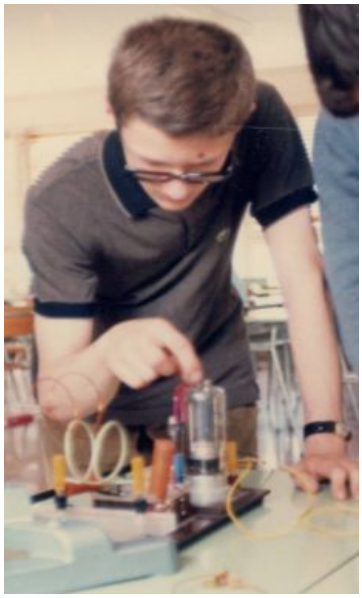
Niente programma e niente variabili per ricordare le scelte passate e condizionare quelle presenti; nessuna possibilità di salvare la situazione. La sfida stava nel ricreare, con questi limiti non trascurabili, la complessità e il senso di libertà di un gioco di avventura su computer; insomma qualcosa che fosse molto più evoluto rispetto ai vecchi libri-gioco degli anni '80.
Per farlo ho costruito uno strumento particolare, **Medusa** (Macchinario Esente Da Un Significativo Acronimo) che mi ha consentito di programmare come su un computer, o quasi, producendo poi in modo automatico tutte le pagine necessarie per rappresentare i possibili stati in cui la vicenda può venirsi a trovare: ad esempio, una diversa variante della stessa pagina per ciascuna possibile combinazione di oggetti posseduti o di azioni compiute. In questo modo, dalle mie 943 pagine iniziali (non spaventi il numero, sono paginette brevi) Medusa produce le **5800 pagine** dell'ebook; ovviamente il lettore non vedrà tutte le varianti delle pagine, ma solo quelle del proprio percorso di lettura. Il bello è che questo sistema non richiede di 'salvare la situazione': è sufficiente proseguire la lettura dalla pagina dove si era arrivati (e naturalmente non ci sono vicoli ciechi; ho scritto un apposito tool per verificarlo).
Dal punto di vista pratico, confesso che all'inizio temevo fosse difficile realizzare dei puzzle non banali con questo sistema; invece poi ho dovuto perfino semplificarne alcuni, perché i collaudatori si trovavano in difficoltà.
Lungo i nove capitoli del racconto, ho man mano scoperto nuove possibilità: posso far prendere oggetti, azionare meccanismi, controllare azioni a tempo (inteso come numero di mosse) e ad un certo punto vi sono addirittura due personaggi che possono agire in modo indipendente.
Ma tutto ciò sarebbe sprecato se il racconto non fosse gradevole da leggere; su questo punto il giudizio sta naturalmente ai lettori; io ho cercato di curarlo al meglio delle mie capacità.

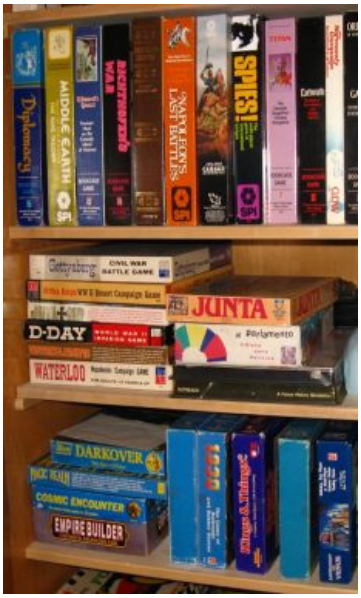
RGP: Attualmente stai lavorando su altri prodotti (editoriali o videoludici) o hai deciso di prenderti una meritata pausa?
ERIX: La meritata pausa me la prenderei volentieri ma il mio conto in banca non è d'accordo, per cui sto guardandomi in giro in cerca di qualcosa che possa magari soddisfare sia la creatività che lo stomaco. Come notavo, non è facile in questo periodo.

RGP: Una domanda e un consiglio: secondo te c'è ancora spazio, oggi, per le avventure testuali, e che linguaggio consiglieresti a chi volesse cimentarsi nella creazione di una propria avventura testuale?
ERIX: C'è spazio se si accetta di far parte di un piccolo club, quasi una società segreta: purtroppo l'interactive fiction tradizionale non sembra avere molta attrattiva per il grande pubblico; tuttavia vi sono diversi appassionati che producono opere interessanti.
Un punto di ritrovo, oltre ai vari siti dedicati alle avventure, è il newsgroup *it.comp.giochi.avventure.testuali* (per leggere un newsgroup basta un qualsiasi client di posta).
Sulla questione del linguaggio, preferisco non pronunciarmi: considerato il mio modo particolare di lavorare e progettare, la mia preferenza per costruirmi gli strumenti e la scarsa attrattiva che hanno su di me i sistemi formali, non mi ritengo in

grado di dare consigli. Mi limito a dire che molti usano Inform; ci sono comunque in giro diversi tool (sconsiglio i miei strumenti in Basic degli anni '80 perché sono ormai del tutto obsoleti).

RGP: Invece, che strumenti di sviluppo consiglieresti a tutti quelli che desiderano imparare a programmare in un qualsiasi linguaggio ma non sanno da dove partire?
ERIX: Non c'è una risposta semplice: come accennavo prima, oggi non basta imparare il linguaggio, anzi direi che è la fatica minore; gran parte del lavoro è svolto infatti dalle librerie, che quindi occorre conoscere.
Faccio un esempio: l'anno scorso mi sono studiato la programmazione iPhone e iOS in genere sulla documentazione Apple; ho impiegato tre giorni per imparare l'Objective-C (sia pure col vantaggio di partenza di conoscere diversi linguaggi) e tre mesi per le librerie principali, senza alcuna pretesa di completezza.
Farei anche un'altra considerazione: è utile imparare a programmare, ma prima di tutto occorre avere chiaro l'obiettivo che ci si pone, che spesso implica certe tecnologie. Voglio imparare a programmare su Android? È meglio se imparo Java. iOS o Mac? Objective-C. Windows? probabilmente C#, ma dipende da cosa voglio fare. Linux? forse C++ (idem). Sistemi embedded? C. Web? Javacript e annessi. Programmazione generica senza necessità di alte prestazioni? Python o Ruby o Perl... e via dicendo.
Come primo step, tra i linguaggi generalisti Python potrebbe essere una buona idea (anche se a me non convince troppo, per altri motivi) avendo una curva di apprendimento graduale, senza gradini significativi all'inizio, ed essendo comunque ampiamente usato, versatile e con ampia scelta di librerie.
Non vorrei fare il vecchietto che dice "Signora mia, ai miei tempi..." ma mi tocca proprio farlo: negli anni '80 con i computer era tutto assai più spontaneo e meno burocratico e quindi ci si divertiva molto di più. Il programmatore, come mi disse profeticamente intorno al 1990 un amico dalla vista lunga, è l'operaio specializzato del XXI secolo. Per fortuna io sono prima di tutto un autore.





RGP: Beato te che ci metti solo tre giorni per imparare un nuovo linguaggio di programmazione...!
Siamo purtroppo giunti al termine di questo incontro con Enrico Colombini che ovviamente ringrazio in modo particolare per la disponibilità e la simpatia con le quali si è sottoposto al mio consueto massacro di domande!

Ricordo inoltre a tutti i lettori di Retrogaming Planet che il sito di Enrico Colombini, costantemente aggiornato, è disponibile all'indirizzo **www.erix.it**.

Settembre 2011, Robert (RGP) Grechi