

Intervista a Enrico Colombini (2)

(<http://www.spagmag.org/archives/backissues/spag51.html>)

Nel 1982 Enrico Colombini scrisse e mise in vendita la prima avventura testuale in italiano, *Avventura nel Castello*, per l'Apple II. Nel 1985 pubblicò due libri sulla scrittura di avventure testuali, che riscossero un grande successo nel rendere popolare il genere. Insieme ai libri veniva distribuito un tool scritto in BASIC, chiamato *Modulo Base*, e degli altri giochi: *L'Astronave Condannata* e *L'Anello di Lucrezia Borgia*, per cinque diversi microcomputer. Più tardi, nel 1988, ebbe modo di rivedere uno dei suoi libri e di potenziare il Modulo Base, dimostrandone le funzionalità aggiuntive in un nuovo gioco per MS-DOS, *L'Apprendista Stregone*. Sviluppò un altro piccolo gioco d'esempio, *Il Drago delle Caverne*, per un corso di BASIC nel 1989. Questi programmi vennero aggiornati e raffinati fino al 1999, quando furono rilasciati come freeware insieme con il libro *Avventure per MS-DOS*. Nel 2000 rilasciò un altro tool, *Idra*, per la scrittura di avventure testuali a scelta multipla¹ in JavaScript. In breve, è una leggenda vivente, ma non diteglielo o vi rimprovererà.

TDF: Enrico, ho scritto una breve introduzione, ma non è sufficiente. Potresti raccontare ai lettori qualcosa riguardo a te stesso?

ENRICO: Da adolescente, credo che avrei potuto essere classificato come l'archetipo del nerd: dal Meccano all'elettronica, avevo il pieno controllo della tecnologia... e di ben poco altro. Tuttavia leggevo parecchio, per cui c'era dentro di me il germe della redenzione. Ora sono sposato, abbiamo un figlio e conduco una vita tranquilla: sono sempre stato un tipo tranquillo, ma non ho mai cessato di essere un anticonformista e un idealista, qualunque ne fosse il prezzo (e talvolta può essere molto alto). Sono seduto alla mia scrivania, ma la mia mente è a spasso in cerca di avventure, come sempre. Ciò che non smette mai di stupirmi è che, delle tante cose che ho progettato nell'elettronica, nel software e nell'editoria, solo i giochi d'avventura sopravvivono nella memoria collettiva: forse sono semplicemente nati al momento giusto (suppongo ci sia una lezione di umiltà in questo).

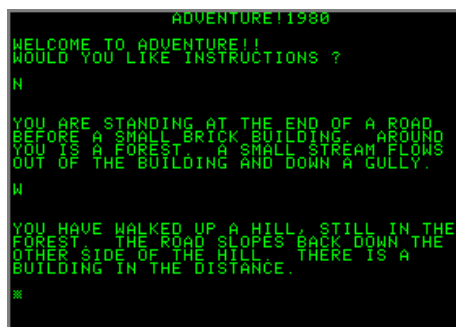
TDF: La gente ti ricorda ancora oggi, soprattutto per via di *Avventura nel Castello*, anche al di fuori della cerchia italiana dell'IF². Parlacene. Come mai hai deciso di scrivere la tua prima avventura testuale? Mi piacerebbe sentir parlare anche della fase di progettazione. Tu accrediti tua moglie, Chiara, come co-autrice... Infine, perché hai scelto un castello scozzese? Riesci a ricordare qualche fonte d'ispirazione?

ENRICO: Come ho scritto sul mio sito, l'ispirazione venne da *Adventure*³, più esattamente dalla versione a 350 punti del 1980 che giocammo sul mio Apple II nuovo di zecca (o meglio, nuovo per me, ma in realtà parecchio usato).

Noi (Chiara, i miei amici ed io) la giocammo molto: era affascinante e, soprattutto, era qualcosa di assolutamente nuovo. Tuttavia, l'impossibilità di salvare il gioco era irritante; non potevo aggirarla copiando il file di salvataggio tra le sessioni di gioco, perché il disco era protetto, così decisi di esaminare la questione più da vicino. Scrisi una primitiva utility di analisi del disco ('DAN'), scoprii il sistema di protezione (era solo un diverso sistema di codifica dei settori), lo rimossi e aggiunsi i comandi di salvataggio e di ripristino. Detti anche un sguardo al codice, ovviamente, ma non rimasi particolarmente impressionato da quello che vidi.



La versione Apple II di Adventure



Un'interfaccia essenziale ma un gran bel gioco

Fu più o meno a questo punto che dicemmo: "È una grande idea, perché non facciamo qualcosa di simile in italiano? Possiamo farlo anche meglio". Così iniziò la fase di progettazione.

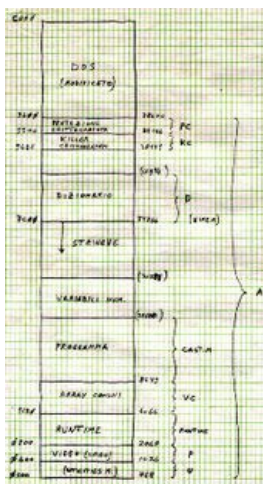
Ora, "fase di progettazione" è un'espressione piuttosto pomposa: in realtà ci pensavamo sopra ogni tanto, tirando fuori idee e discutendole. Questo è il mio modo standard di progettare quando non sono sotto pressione, e di solito funziona bene perché è il mio subconscio che fa tutto il lavoro: devo solo essere paziente e attendere i risultati. Chiara ebbe un ruolo importante nella progettazione: discuteva, correggeva, spesso respingeva le mie idee, oltre naturalmente a contribuire con sue idee originali (è difficile ricordare chi dei due abbia proposto quali idee).

Non appena riuscimmo ad avere un minimo di mappa e alcuni enigmi, iniziai a scrivere codice alla sera. Io ero il programmatore, perché Chiara non era molto interessata alla programmazione sui personal computer (in quel periodo lavorava in linguaggio assembly sui microprocessori). Il primo programma era piuttosto primitivo, più o meno una specie di grandi IF... THEN... ELSE con delle semplici PRINT per l'output, ma funzionava ed era abbastanza promettente da spingerci a proseguire. Andammo avanti per un paio di mesi, aggiungendo locazioni ed enigmi, finché non andai a sbattere contro la barriera della memoria: i testi riempivano i 48 Kb (in realtà, decisamente meno all'atto pratico) disponibili sul mio Apple[[. Per cui imparai ad usare efficacemente i floppy disk e trasferii tutti i testi in un file indicizzato per liberare RAM, quindi riprogettai il programma in modo più table-driven e, in generale, più sano. Come effetto collaterale della crescita del programma, m'imbattai in due nuovi problemi: l'estrema lentezza della garbage collection del BASIC (che poteva bloccare la macchina per parecchi minuti in qualsiasi momento) e il tempo relativamente lungo impiegato per cercare una parola (estratta dall'input dell'utente) nel dizionario. Così feci una cosa buona e una cattiva.

La cosa buona fu studiare le interiora dell'Applesoft BASIC e leggere parecchio in giro, il che mi condusse ad un modo semplice ma molto efficace di partizionare le stringhe in due aree: "collezionabili" e "non collezionabili". Poiché la stragrande maggioranza delle stringhe erano costanti, questo approccio eliminò il problema della garbage collection.

Riguardo alla cosa cattiva che feci, provo ancora un po' di vergogna: conoscendo ben poco d'informatica e algoritmi, stavo ingenuamente facendo una ricerca lineare. Una semplice ricerca binaria avrebbe risolto il problema con il minimo sforzo ma, non conoscendola all'epoca, feci quello che sapevo fare: ricodificai la ricerca lineare in linguaggio assembly. Funzionava, certo, ma è ancora una macchia oscura sulla pergamena della mia (per il resto quasi decentemente pulita) esperienza di programmazione.

Il gioco progrediva e si evolveva, progettazione ed implementazione andavano avanti di pari passo e, a mano a mano che i mesi passavano, nuove idee e nuovi enigmi venivano aggiunti. Avevamo sicuramente molte fonti d'ispirazione, prima di tutto il gran numero di giochi giocati: sono certo che alcune idee furono rub... ehm, ispirate da antichi giochi d'avventura, ma è difficile ricordare cosa potesse provenire da dove (per esempio, qualcuno ha fatto notare la somiglianza tra l'introduzione con l'aereo che precipita e una scena simile in *Cranstor Manor*, che ricordo di aver vagamente giocato, ma non ho idea di quando).



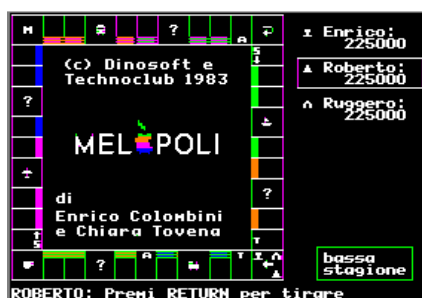
La caduta "grafica" dell'aeroplano proveniva dal baratro di *Adventure*, naturalmente, mentre il labirinto venne modellato su quello de *Il nome della rosa* di Umberto Eco; stanco di mappature insensate, pensai a lungo su come fare un labirinto non convenzionale e la biblioteca in quel libro mi fornì l'idea giusta. Parlando di non-convenzionalità, l'intero gioco fu progettato per essere una sfida al modo "standard" di risolvere i problemi nei giochi (del tipo: "vai e ammazzali tutti") e possiamo ritenerci fieri del risultato.

Il collaudo del gioco ebbe un ruolo molto importante: per vari mesi guardai gli amici giocare e presi nota di tutto ciò che scrivevano, anche le cose più strane o inattese, e poi aggiunsi la maggior parte di esse al gioco (specialmente se strane o inattese!). Penso che in effetti ciò debba essere considerato come parte della progettazione: in pratica stavamo usando la testa di altri oltre che la nostra. Per quanto riguarda la scelta di un castello scozzese come sfondo, non saprei dire: forse fu l'influenza dei libri di Stevenson, o di Poe, o di qualche romanzo gotico... ma quando (molti anni dopo) andammo effettivamente a visitare la meravigliosa Scozia, fummo contenti di vedere che la nostra ambientazione era piuttosto accurata.

TDF: Sei riuscito a vendere *Avventura nel castello*, sei uno dei pochi autori italiani che ha guadagnato qualcosa con questo tipo di software. So che hai iniziato vendendo in proprio (e questo mi ricorda Roberta e Ken Williams). Quali problemi hai dovuto affrontare? È stato difficile trovare una software house, in seguito? Potresti informare i lettori sul mercato italiano del software negli anni '80? Il tuo gioco avuto una lunga vita: tre edizioni commerciali, e la terza era per MS-DOS. Alla fine, ci hai fatto i soldi?

ENRICO: All'inizio, era solo un favore che mi fecero un paio di amici: avevano un negozio di computer (forse il primo nella mia città) dove ci scambiavamo conoscenze e strumenti, e vendettero... beh, vendettero 12 copie entro la fine del 1982, secondo le mie carte. L'anno seguente, l'editore per il quale avevo cominciato a scrivere articoli tecnici (il Gruppo Editoriale Jackson) avviò una divisione di vendita software (J.Soft), e fui quindi in grado di proporre loro un paio di giochi, compresi *Avventura nel castello* (che aveva appena vinto il 1° premio al primo concorso italiano di giochi per computer, Computer Play 83).

Con il supporto delle loro riviste di computer, vendettero circa 600 copie e, soprattutto, riuscirono a raggiungere un accordo con Apple Computer Italia per fornire i giochi in bundle con il nuovo Apple IIc, per cui molta gente fu in grado di giocarli. Gli altri due giochi (il mio gioco da tavolo *Melopoli* e il ben progettato gioco di strategia *Signori della Galassia*, opera di un amico) ebbero tuttavia un impatto meno duraturo. Il commercio del software non fu però un successo stellare, a causa di condizioni ostili in Italia, vale a dire pochi computer in giro, la mancanza di cultura tecnica e la pirateria diffusa (spesso fatta alla luce del sole da parte dei rivenditori stessi e talvolta tollerata se non incoraggiata da alcuni venditori di hardware - mi viene in mente la Commodore).



Speravo ancora di poter vivere con la progettazione e la vendita di giochi, ma si rivelò impossibile. Nel frattempo, altri paesi stavano avviando una vera industria di giochi per il computer; feci anche un mezzo tentativo di contattare un editore francese, ma senza successo (non sono mai stato un buon venditore).

Il colpo di grazia per le mie ambizioni arrivò quando la Apple dichiarò che non voleva giochi per il Macintosh (in quel periodo ne stavo sviluppando uno). Continuai così a guadagnarmi il mio pane (ed anche la marmellata) con i corsi di computer e le enciclopedie; quanto ai giochi, ahimè, dovetti accontentarmi di giocarli, solitamente sul PC IBM-compatibile, che stava rapidamente diventando il nuovo standard dopo il suicidio commerciale della Apple (ma questa è un'altra storia). Ad ogni modo, volevo che la gente potesse giocare ai miei giochi, così feci una versione MS-DOS. Fu una riprogettazione completa, basata su un linguaggio specializzato che avevo progettato, e mi fece capire molte cose. Imparai, per esempio, che utilizzare un linguaggio specializzato per scrivere dell'IF non era necessariamente una buona idea, almeno quando l'autore è anche un decente programmatore (più tardi, ottenni risultati migliori utilizzando un approccio ibrido).

Ah, la versione per MS-DOS vendette circa 100 copie attraverso la Hi-Tech (per la quale stavo scrivendo su una rivista per gli utenti Apple); furono indubbiamente piratate un numero molto più alto di copie, ma a quel punto mi interessava più la diffusione che gli incassi.

Sul diventare ricchi... guadagnai milioni! Purtroppo sbagliai il tempo: l'euro non c'era ancora, così si trattò di milioni di lire, da ridursi quindi di un fattore di circa 2000:1. Ma, tecnicamente parlando, le avventure testuali fecero di me un milionario.

Hai citato Roberta e Ken Williams: furono senza dubbio dei pionieri, e mi divertii con alcune delle loro prime avventure grafiche, ma gli autori di IF che amavo erano nel campo della Infocom, Steve Meretzky soprattutto, ma anche molti altri (a proposito, *Enchanter* mi suggerì l'idea di base per *L'Apprendista Stregone*). La maggior parte delle loro avventure avevano buone storie, buono stile narrativo, buone idee e buona cura dei dettagli. È stato bello, finché è durato.

TDF: Dopo il tuo primo gioco, approdasti nelle librerie con due libri, *Avventure e Scrivere un gioco d'avventura*. Il primo comprendeva un'audiocassetta, o un dischetto, con tre programmi: due giochi d'esempio (*L'Astronave condannata* e *L'anello di Lucrezia Borgia*) e lo strumento da te utilizzato per scriverli (il Modulo Base). Era uno strumento piuttosto semplice e tu scelsi il BASIC. Avevi un modello in mente, ovvero le avventure di Scott Adams? Avevi guardato a un pubblico più vasto? Ora posso dire che hai avuto un impatto profondo su (quasi tutti) gli sviluppatori italiani d'IF: ognuno si sforzò di aggiungere funzionalità, e qualcuno (cioè Bonaventura Di Bello) vendette anche dei giochi che avevano il tuo strumento come spina dorsale.

ENRICO: In realtà, il titolo che proposi era *Imparare il BASIC scrivendo avventure*, ma al mio editore non piacque e gli editori hanno sempre ragione (voglio dire, il loro libretto degli assegni ha sempre ragione). L'idea fu... beh, si spiega da

sé: la programmazione era divertente e non era necessario alcun titolo accademico per apprenderla.

L'unico modello che avevo in mente era il motore di *Avventura nel castello*; gli interpreti di Scott Adams erano stati progettati per risparmiare ogni bit di RAM in macchine molto piccole, mentre io al confronto lavoravo nel lusso e non avevo bisogno di comprimere i dati. Il mio motore, tuttavia, era troppo complesso sia per i principianti da usare, sia per me da spiegare in maniera decente, così lo resi più semplice togliendo alcune caratteristiche; un oggetto, ad esempio, non poteva più avere degli stati (es. un osso che poteva essere intero o rotto), ma andava invece sostituito con un oggetto diverso (un osso intero, un osso rotto).

Con mio grande stupore, il motore "ridotto" si rivelò per certi versi migliore di quello originale, e certamente più facile da usare. Riprogettare dopo un po' d'esperienza può dare risultati migliori, soprattutto se l'obiettivo è distillare e preservare l'essenza, scartando la robaccia ridondante.

Più tardi, la seconda versione del Modulo Base reintrodusse alcuni concetti utili, come i file indicizzati su disco e alcuni comandi di uso frequente incorporati nei messaggi, raggiungendo (a mio parere) un buon equilibrio fra potenza, flessibilità e facilità d'uso.

A molti, tuttavia, bastavano le funzionalità della prima versione. Era del codice semplice (in qualche punto alquanto primitivo), ma avevo meditato sui concetti di base per anni, per cui era uno strumento utile. Altri, come Bonaventura Di Bello che hai citato, lo sfruttarono fino all'osso e oltre: una domenica mattina mi chiamò (svegliandomi) per chiedermi, se ricordo bene, come aggirare alcuni limiti intrinseci del programma in sé (ad esempio il numero massimo di parole differenti, cosa non semplice come può sembrare). La versione 2 non esisteva ancora, così gli detti alcuni suggerimenti di cui lui fece buon uso: pubblicò una serie di giochi di successo per l'edicola, alcuni dei quali usavano il mio strumento. Almeno è così che lo ricordo; spero di non confonderlo con un altro utente... sai, la vecchietta e tutto il resto... ma della telefonata sono ben certo⁴. 😊



Il libro 'Avventure' in varie versioni

Per quanto riguarda il Modulo Base, il programma aveva un semplice parser a due parole, ma continuo a pensare, dopo tutti questi anni, che un parser e un modello del mondo più completi e 'realistici' non implicino necessariamente dei giochi più divertenti, anche se sarebbero sicuramente più interessanti dal punto di vista dell'AI (intelligenza artificiale) e in funzione di un (sempre-quasi-pronto-ma-mai-del-tutto) vero riconoscimento vocale. Ho la sensazione che sia molto facile innamorarsi della tecnologia dimenticandosi però della giocabilità.

TDF: L'Apprendista Stregone è il tuo preferito, e piace molto anche a me. Si tratta di un lavoro ambizioso anche se affermi di averlo scritto in una quindicina di giorni. Ti ha aiutato qualcuno? E l'avevi ben pianificato in anticipo, giusto? Immagino che avessi delle scadenze da rispettare: lavori meglio sotto pressione? Hai una caratteristica che posso notare anche nei tuoi giochi d'esempio precedenti: presti particolare attenzione alla storia. E che dire dei personaggi (umani e non) e dell'ambientazione? Dove hai trovato l'idea del sistema di magia?

ENRICO: Chiara contribuì, come sempre; le sue conoscenze classiche furono molto utili, anche se anch'io conosco, ehm, dovrei conoscere, un po' di latino (ma non, purtroppo, il greco). La scelta di nomi appropriati per gli incantesimi fu un esercizio divertente.

L'affermazione che venne scritta in una quindicina di giorni è vera... il trucco sta nello "scrittura". La progettazione richiese molto più tempo, come al solito: lasciavamo che le idee maturassero e prendessero lentamente forma, non diversamente da cristalli (con o senza difetti, sta ai giocatori decidere).

La foresta, per esempio, venne da un viaggio a Salisburgo: l'ammirammo mentre stavamo viaggiando comodamente in pullman, e ci meditammo sopra. D'altra parte, evitammo accuratamente di mettere nel gioco l'incongruo gnomo-minatore che si trovava nelle famose miniere di sale di quella bellissima città. Suppongo che fosse lì per essere ammirato dai turisti americani, o almeno lo spero. Ma sto divagando.

La scadenza ineludibile apparve improvvisamente quando il mio editore mi chiese di aggiungere qualcosa per la nuova edizione. Ci tenevo davvero a vedere *L'Apprendista Stregone* pubblicato, così lavorai giorno e notte per un paio di settimane. Non so se lavoro effettivamente meglio sotto pressione: l'unica cosa certa è che lavoro di più. 😊

L'idea delle parole magiche era spudoratamente presa da Enchanter della Infocom, ma era lasciato al giocatore l'onere di scoprirla, mentre l'approccio narrativo fu naturalmente una scelta di progettazione: volevo che la gente si godesse la storia e le impostazioni senza dover disegnare mappe complesse o risolvere enigmi diabolici. Le sfide che scelsi di mettere erano per lo più del tipo "pensiero laterale" e sono piuttosto soddisfatto del risultato, anche se mi sarebbe piaciuto avere un po' di tempo extra per le rifiniture (ma poi ne avrei sicuramente chiesto di più).

Per i personaggi principali, il vecchio mago Artemio e il suo giovane apprendista (il giocatore), non mi viene in mente alcuna fonte precisa: è un tema comune, dopotutto. Ma ci divertimmo a inserire noi stessi nel gioco, anche se c'è poca somiglianza con gli originali: Chiara, ad esempio, non predice il futuro ma scrive programmi... uhm, in realtà, ora che ci penso, l'imprevedibilità gioca un ruolo importante in entrambi i casi. Per quanto riguarda me e l'illusionista... dopotutto, i giochi sono una sorta d'illusione, no? A proposito, la mia professoressa di matematica mi rimproverò per aver abbandonato l'università, così le assegnai la stessa parte nella storia, travestita da vecchio mago.

Infine, mi piaceva il nome che avevo scelto per il "mio" personaggio, così adottai "Erix" come la mia firma in rete, in quei nebbiosi tempi pre-Web, e la uso ancora felicemente.

[Ricomincia dall'inizio](#)

[Salva la situazione corrente](#)

[Riprendi la situazione salvata](#)

- - -

[Informazioni sul gioco](#)

[Informazioni su Idra](#)

Una storia di campagna

racconto-gioco di Andrea Angiolino
programma di Enrico Colombini

"È l'aratro che traccia il solco, ma è il bifolco che lo difende."
(attribuita al Duca, durante la Guerra del Guano)

Siamo abituati a incontrare principi, cacciatori di draghi, cavalieri erranti, maghi e apprendisti stregoni: ma il popolo dov'è? In una società prevalentemente agricola come quella fantasy, in effetti, il contadino dovrebbe avere un ruolo fondamentale.

Abbiamo deciso di riparare a questo torto. E allora armatevi di scarpe grosse e di cervello fino e immedesimatevi in un contadino della Piana di Exen: a manovrare una spada sono bravi tutti, ma provate un po' a cavarvela con una zappa...

- [Continua](#)

v 1.0 - Febbraio 2000, © A. Angiolino, E. Colombini 2000

Idra in azione con un racconto-gioco di Andrea Angiolino

TDF: Con Idra ti sei indirizzato verso le CYOA⁵. Potresti brevemente presentare ai lettori questo strumento? Perché l'hai sviluppato? Non hai rilasciato alcuna CYOA, infatti i due esempi inclusi nel pacchetto sono di altri autori. Posso supporre che ti piaccia questa forma, e magari hai già letto dei libri simili in passato o hai condiviso questa passione con gli amici.

ENRICO: Non sono sicuro che Idra sia esattamente uno strumento per le CYOA: può certamente essere utilizzato in tal senso (in realtà, è il modo più semplice di usarlo) ma, nelle mani di un buon programmatore, può essere parecchio flessibile. Avrei desiderato scrivere un'avventura complessa per mostrarne le capacità, ma questo implicava una pianificazione complessa... e il tempo... e la voglia... in breve, non mi ci sono mai messo.

Idra nacque da una domanda: la maggioranza delle persone evita le avventure testuali perché non ha voglia di leggere, o perché non ha alcuna voglia di scrivere? Così scrissi un semplice strumento in HTML/Javascript che, in un certo senso, emulava le avventure grafiche punta-e-clicca, ma senza la grafica.

I risultati, devo dire, sono stati inconclusivi: sì, più persone hanno accettato di giocare ai giochi (rispetto alle persone disposte a giocare alle avventure testuali in cui è richiesta la scrittura), ma d'altra parte sono state facilmente annoiate da lunghi testi (dovrei dire, da testi non infinitesimali). Così, alla fine, dev'essere una combinazione di fattori (come l'essere cresciuto con più libri che TV) che controlla l'interesse per la pagina scritta, sia essa un libro o un gioco.

Tra l'altro, la prima volta ho incontrato un libro CYOA, non era affatto un gioco. Ce l'ho ancora: era un' *Introduzione alla*

genetica della serie Tutor del 1967 (un pochino di anni prima della WWW mania...). Il libro poneva una domanda e reindirizzava il lettore ad un'altra pagina a seconda della risposta, per spiegare l'errore o per rinforzare l'apprendimento. Era ben progettato. Molto tempo dopo comprai i libri-gioco e trovai il loro design piuttosto deludente e primitivo... anche se li giocai comunque. 😊

Tornando a Idra, ho iniziato di recente un altro progetto: un motore in DHTML/CSS per scrivere avventure testuali, con alcune interessanti tecniche e un approccio più 'pseudo-grafico'. Ho imparato il DOM e i CSS, lottato con i problemi di compatibilità per un paio di mesi, scritto una libreria, dimostrato oltre ogni dubbio che la cosa era fattibile... e poi l'ho abbandonato, come al solito.

Accade sempre così: una volta che ho acquisito la conoscenza che m'interessa, un'applicazione pratica è solo lavoro sprecato... a meno che, naturalmente, qualcuno non sia veramente interessato ad essa (o, Dio non voglia, disposto addirittura a pagarla).

Ho i miei cassette fisici, digitali e mentali pieni di tali esperimenti, in fasi di completamento più o meno avanzate; molti di essi esplorano approcci e possibili innovazioni delle avventure testuali in una varietà di linguaggi (Basic, C, Prolog, Perl, Dylan, Bash script, Lua...) o anche solo come esercizi di pensiero. Purtroppo, la strada che va dalla dimostrazione di fattibilità al prodotto finito è lunga e noiosa, soprattutto per un tipo pignolo come me. Se a volte un lavoro completo come Idra vede la luce, temo che sia una sorta di incidente. Ma, in questo mondo imperfetto, gli incidenti accadono.

TDF: Dicci di più sui tuoi esperimenti con i linguaggi di programmazione. Non è un segreto che tu pensi che il Lua sia molto adatto per lo sviluppo di IF o uno strumento per l'IF stessa. Qual è il problema con gli altri linguaggi?

ENRICO: Beh, dopo *Avventura nel castello* avevo l'illusione che si potesse costruire un modello del mondo: una sua perfetta rappresentazione logica. Erano gli anni in cui il concetto di "object-oriented" era venuto di moda ed ero un convertito della prima ora, così cercai di costruire un modello a oggetti in C, o forse era objective-C (usando il compilatore Manx che avevo appena comprato per il mio Macintosh) completo di oggetti (nel senso dell'avventura), attributi e così via. Il mio codice funzionava, ma capii presto molte cose: in primo luogo, che la costruzione di una libreria completa (o anche solo passabilmente realistica) era impossibile a causa dell'esplosione combinatoria e del rifiuto del mondo reale ad essere così rigidamente classificato; in secondo luogo, che più complessa era la libreria, più l'interfaccia di gioco era rigida e prevedibile; come terza (e più importante) considerazione, che non vi era alcuna relazione tra la complessità del modello del mondo e il divertimento del giocatore. Se, ad esempio, possiamo guardare sotto e dietro un oggetto, dobbiamo guardare sotto e dietro a tutti gli oggetti presenti nel gioco, e ciò è estremamente noioso e per nulla divertente.

Nello stesso periodo, sperimentai anche altre strade: provai il *Prolog*, un linguaggio dichiarativo (non si scrivono delle istruzioni ordinate: si scrivono invece delle affermazioni, o qualunque sia il termine corretto, e il sistema si prende cura di tutto il resto). I sistemi esperti avrebbero dovuto essere molto potenti, quindi ero interessato a provare qualcosa del genere. Non so quanto avrebbe potuto funzionare, ma so che quasi subito esaurii i 640 Kbyte del mio PC M24 quando provai a portare *Avventura nel castello* sul Turbo Prolog della Borland. Peccato.

Ma sto divagando, torniamo al punto: a mio parere, per fare un gioco di avventura divertente non c'è bisogno di simulare accuratamente un mondo, o di simulare qualsiasi cosa: occorre solo rispondere ai comandi del giocatore in maniera comprensibile, ed evitare di essere ripetitivi o troppo prevedibili. Non importa se la risposta sia logica o no, deve solo essere adeguata.

Dal punto di vista dell'implementazione, ciò si traduce in un approccio diverso (rispetto al modello del mondo): invece di cercare d'averne una comprensione logica di ciò che il giocatore dice, il che implicherebbe un numero assai elevato di assunzioni inesprese, ci si può limitare a fare un semplice (a volte non così semplice) pattern matching. È il modo in cui un bambino impara una lingua, dopotutto, ed è probabilmente il modo in cui il nostro cervello funziona normalmente: il riconoscimento di schemi.

Quindi, non m'importa di quello che le parole significano in realtà, ma devo invece sapere che certe espressioni, o parole, implicano determinate aspettative, in modo che io possa dare una risposta significativa.

In pratica, questo significa eliminare quasi tutte le classificazioni logiche e sintattiche, e raggruppare i dati intorno a ciò che il lettore potrebbe effettivamente dire e aspettarsi. Ad esempio, dati un pulsante e una carota, posso aspettarmi che il giocatore dica qualcosa come:

- premi il pulsante
- mangia la carota
- premi il pulsante con la carota

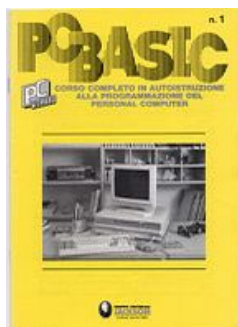
ma, naturalmente, questo è ben lungi dal comprendere tutto ciò che il giocatore potrebbe dire, quindi introdurrò i caratteri jolly:

- premi \$1
- mangia \$1
- premi \$1 con \$2
- premi il pulsante con \$1
- premi \$1 con la carota

- \$1 pulsante
- \$1 carota
- \$1 \$2 con \$3

Si noti che il sistema non ha alcuna idea (almeno in questa fase) di ciò che le parole in realtà significhino, per esempio se "premere" sia un verbo o se "carota" e "pulsante" siano degli oggetti: le risposte sono associate ai pattern. Con adeguate regole di precedenza, tale sistema potrebbe essere percepito come molto più realistico rispetto a una libreria 'world-model'... almeno per un dato gioco.

Possono però essere facilmente osservati due inconvenienti: questo sistema è ad alta intensità di lavoro e non conduce ad una libreria general-purpose riutilizzabile. Ma questo è esattamente il mio punto: un buon gioco d'avventura è un'opera d'arte, non un prodotto industriale. Se create una libreria riutilizzabile, la maggior parte del vostro gioco sarà prevedibile, in altri termini, avrete fatto buona ingegnerizzazione... e cattiva arte.



```

# AZIONI #
1510 REM - LABEL INTERPRETE:
1520 DATA "-1 IU97<?</LMorto. L
=>L0=N1/99.>M-40:X-4:R-12:G
-3"
1530 DATA "-2 IU97<?</LFine. L=
>L0=N1/99.>M-49:X-4:R-12:G-
3"
1540 DATA "-3 M-34:U0=N1/3:IU0=
1<L0=0 >IU0=2<R-7.>L0=-3:??</
L/LCiao!/L>"
1550 DATA "-4 M-45:780:??< >:M-4
5:7096:??<.>: REM PUNTI
1560 REM - AZIONI INTERPRETE:
1570 DATA "I39N:M-19": REM ENTR
1580 A
X DATA "I39X:M-19": REM ENTR
A

```



Devo dire che ho sovrasemplificato la faccenda per ragioni di spazio, e ho anche reso in bianco e nero delle cose che in realtà hanno infinite sfumature di grigio. Ho anche forse esagerato un po' la differenza fra il modello del mondo statico e il puro pattern matching, al fine di illustrare i diversi concetti sottostanti, anche se le implementazioni pratiche (la mia compresa) possono prendere da entrambe le scuole.

Parlando d'implementazioni pratiche, mi piacerebbe farla finita con il dizionario, la mappa e altre cose inutili. Il mio ideale di file di definizione di un'avventura testuale dovrebbe contenere solo pattern, cioè frasi complete o caratteri jolly, e risposte: tutto il resto (dizionario, tabelle, mappa, ecc.) dovrebbe essere creato dal programma. È un po' utopistico, lo so, ma penso che sia un obiettivo su cui valga la pena puntare.

Ad ogni modo, lavorando in questa direzione ho presto capito che i linguaggi staticamente tipizzati (per quanto mi piaccia il C per altri impieghi) sono un vero ostacolo in questo campo. I linguaggi dinamicamente tipizzati sono molto più adatti a tale compito; possono, per esempio, creare "proprietà" in loco, consentono di mescolare codice e dati, o di passare in giro qualsiasi tipo di dati, anche quelli che abbiamo appena creato.

Tutto questo, direte, equivale a cercarsi dei guai, ed effettivamente è vero. A meno che tutta questa libertà selvaggia non venga inquadrata in una costruzione (framework) ben pensata. Il mio Idra è un esempio molto limitato di framework di questo genere, ma penso che dimostri il punto in questione: un linguaggio completamente dinamico (Javascript, in questo caso) può essere addomesticato e inquadrato per costruire uno strumento flessibile, in cui il framework dà una chiara direzione e controllo, ma la parte "dinamica" lascia aperte infinite possibilità (beh, Idra non è esattamente un interprete di avventure testuali, ma il principio è lo stesso).

Tra l'altro (e sulla stessa linea) odio i compilatori: con la potenza di calcolo che abbiamo oggi sulle nostre scrivanie, non c'è alcun bisogno di una tale seccatura. Tutta la costruzione di tabelle, i riferimenti incrociati e l'analisi possono essere fatti in tempi molto brevi all'avvio del programma, senza che il lettore nemmeno se ne accorga. Quindi, se qualcuno è ancora sveglio dopo tutte queste divagazioni, posso rispondere alla tua domanda: perché mi piace Lua? Non è una religione (dopo tutto, ho usato molti linguaggi), è solo uno strumento che trovo molto adatto per questo compito, perché, tra le altre cose: è molto compatto, semplice ma potente, totalmente dinamico (in pratica è molto simile a Scheme - un dialetto del Lisp - ma in una forma più conveniente), non ha astrusità sintattiche (mi viene in mente Python...), può essere facilmente integrato con il C e, ultimo ma non meno importante, è libero e assai ben supportato. Essendo anche una sorta di "scatola di montaggio" per la creazione di nuovi linguaggi, è molto comodo per la progettazione d'interpreti specializzati, come quelli che fanno funzionare i giochi d'avventura.

TDF: Ok, non ti piacciono molto le librerie, ma a volte è possibile modificarle o riscriverle. Cosa ne pensi dei linguaggi di programmazione specifici per l'IF, come Inform, TADS, Hugo? Hai provato le loro ultime incarnazioni? Dai, non è possibile scartarli troppo facilmente... la maggior parte delle persone scrive le avventure in questo modo. Hai detto che i giochi sarebbero prevedibili se tutti usassero le librerie; c'è del vero in questa affermazione, ma la mia domanda è: sono gli autori odierni in grado di scrivere della buona IF? Ricorda quello che la Z-Machine ci ha dato negli anni '80... e penso che noi possiamo ancora adesso ottenere dei buoni giochi!

ENRICO: Comincerò confessandoti il mio peccato più grande: non ho giocato alla maggior parte dell'IF contemporanea... almeno non ancora. Il motivo principale è che trovo il gioco cooperativo molto più divertente che quello in solitario, ma

nessuno dei miei amici è più interessato all'IF.

Per gli strumenti e i linguaggi, ho dato un'occhiata al manuale di Inform e l'ho trovato troppo formale e statico (questa è la mia impressione, potrebbe essere sbagliata). Mi pare che il suo principale valore risieda nella libreria, ma (come dicevo), le librerie sono un'arma a doppio taglio: da una parte rendono il lavoro dell'autore molto più facile, dall'altra rendono la vita del giocatore molto più ripetitiva, soprattutto quando sono ampiamente riutilizzate. Parte di questa limitazione è implicita nell'interfaccia utilizzata per l'IF, naturalmente: gran parte dell'interazione di gioco ha assunto forme 'canoniche' ed è quindi altamente prevedibile (no, non ho un'idea migliore da proporre, o comunque non una coerente).

Non ho provato gli altri strumenti che hai menzionato. Credo che siano una scelta saggia per gli autori 'puri' che non hanno interesse a sporcarsi le mani con i problemi tecnici, ma io sono come quei vecchi pittori che preferiscono macinare da sé i propri colori. Non mi manca l'esperienza di programmazione e mi piacciono le sfide, così costruire un framework (specialmente su percorsi inesplorati) è per me una parte significativa del divertimento. Inoltre, utilizzare uno strumento esistente significa viaggiare lungo una strada esistente, e non è ciò che più mi interessa.

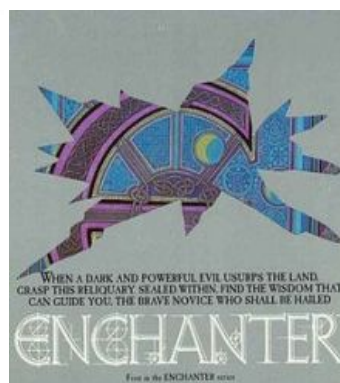
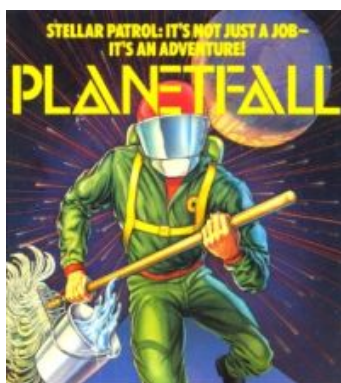
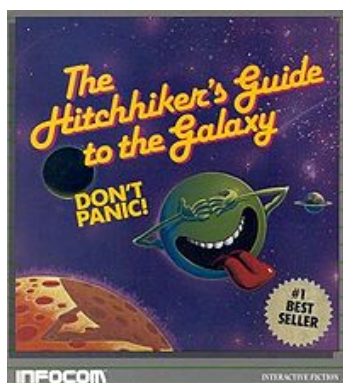
TDF: Se fossi "cattivo", ti chiederei di nominare i pochi titoli d'IF contemporanei che hai giocato, ma non ti costringerò. 😊 Quali sono invece i tuoi pezzi preferiti? Hai detto d'apprezzare il lavoro di Meretzky. Se potessi rubare un'idea dai suoi giochi (o da qualsiasi altro gioco, se preferisci), quale sarebbe?

ENRICO: Avendo giocato a un campione molto ristretto d'IF contemporanea (scelta più o meno a caso), sarebbe abbastanza ingiusto per me parlarne: tenderei inevitabilmente a giudicarle in rapporto alle mie (forse eccessive) aspettative, non in relazione agli altri lavori. Passo quindi su questo punto.

Ho giocato una serie d'avventure grafiche di recente, ma probabilmente non è una cosa saggia da dire in questo contesto... comunque, c'è sempre qualcosa da imparare riguardo le tecniche narrative, i trucchi e le interfacce, anche quando il giocatore non ha bisogno di comunicare in forma scritta. Penso che l'IF potrebbe trarre beneficio da innovazioni radicali dell'interfaccia.

Ho apprezzato Meretzky sia per il suo stile arguto che per le sue sorprese narrative, in particolare il personaggio di Floyd in *Planetfall* (l'idea più bella che mi sarebbe piaciuta rubare!), anche se alcune parti dei suoi giochi possono talvolta essere alquanto vuote e noiose. Naturalmente non era l'unico autore che mi piaceva, anche se il suo nome è rimasto inciso nella mia memoria.

Suspended di Michael Berlyn, per esempio, era un capolavoro d'innovazione con i suoi sei robot che agivano in qualità di singoli sensi del giocatore ibernato. *Hitchhiker's Guide to the Galaxy* di Douglas Adams era estremamente divertente (anche se un po' difficile e a volte sleale verso il giocatore), anche grazie alla collaborazione di... toh, Steve Meretzky di nuovo. Ho recentemente rigiocato *Enchanter* (di Marc Blank e David Lebling) solo per vedere se sono affezionato a quei giochi solo perché sono stati pubblicati nel periodo d'oro. Mi sembra che non sia così: *Enchanter* è divertente da leggere e da giocare ora, come lo era allora.



Gli autori moderni, naturalmente, devono affrontare problemi più ardui di quelli dei pionieri storici e quello più importante è che devono lavorare su dei percorsi ormai tracciati, quindi è assai difficile inventare qualcosa di veramente nuovo. Andare 'fuori dalle righe' diventa sempre più difficile col passare del tempo, perciò coloro che sono in grado di farlo dimostrano grande talento.

In un certo senso, dovendo lavorare all'interno di un framework predefinito, è probabilmente più facile raccontare una storia originale piuttosto che ideare puzzle originali, così i lettori che preferiscono un approccio più orientato al racconto sono più facili da soddisfare rispetto a quelli che cercano rompicapi (io mi pongo da qualche parte nel mezzo).

Come dicevo, penso che una radicale innovazione dell'interfaccia potrebbe fare meraviglie per ringiovanire l'IF. So della nuova versione di Inform; è sicuramente un approccio interessante, anche se mi ricorda vagamente i 'linguaggi facilitati' HyperTalk e Applescript di Apple, il che mi fa correre i brividi lungo la schiena. Ma potrei sbagliarmi di grosso: Inform 7 è giovane, quindi è giusto aspettare e vedere cosa ne verrà fuori. Anche se non dovessero funzionare, gli esperimenti falliti sono sempre meglio della stagnazione.

Nel frattempo, sto pensando pigramente lungo linee differenti. Ad esempio, un parser a una parola (largamente

dipendente dal contesto) che mimi ciò che in realtà spesso si dice in una conversazione tra amici nel mondo reale. Pensateci sopra: quante volte pronunciate frasi complete? Oppure, lungo una linea diversa, sistemi ibridi che combinino testo scritto con clic e trascinamento. Ho visto alcuni esperimenti, ma ancora nulla di notevole (ma magari ho guardato nei posti sbagliati). Stavo lavorando su uno di questi esperimenti qualche mese fa, ma poi ho trovato un nuovo lavoro che occupa la maggior parte del mio tempo.

TDF: Non hai rilasciato nulla da un bel po' di tempo. Stai scrivendo un gioco? Sai che i giocatori italiani vorrebbero vedere un nuovo pezzo d'IF da Colombini... Hai detto che la costruzione di un framework è divertente: puoi accennare a qualcosa? Qualche lavoro in corso?

ENRICO: Sapevo che sarebbe venuta fuori questa domanda. 😊

Ora, odio deludere i miei lettori, ma la mia mente funziona in un modo strano: quando ho del lavoro da fare sono quasi totalmente assorbito (diciamo in stato 'on') e quando non ne ho sto in 'off' ed è difficile per me trovare l'energia sufficiente per portare a termine un progetto completo (al contrario di un semplice studio, o del prendere nota di idee grezze).

Quindi sono un po' come Sherlock Holmes, con la differenza che uso i giochi per il computer al posto della cocaina per evitare che la mia mente arrugginisca: è più economico, divertente (suppongo, non sono interessato a verificare) e decisamente più sicuro.

In effetti, la maggior parte dei miei giochi sono stati scritti come lavori con scadenze editoriali (o almeno in vista della pubblicazione), con la notevole eccezione di *Avventura nel Castello*. Ma quella, essendo un'esplorazione di un territorio sconosciuto, era abbastanza stimolante in sé. Ma poiché la probabilità che qualcuno mi paghi per scrivere avventure testuali nel XXI secolo è alquanto bassa, ho paura che dovrà passare del tempo prima che possa pubblicare delle nuove IF (anche se voglio lasciare la porta aperta a questa possibilità).

A proposito, il mio non essere in grado di sopportare i lavori ripetitivi è il motivo principale per cui io sono a volte disoccupato, specialmente in questo paese dove innovazione e creatività sono parole sconce. Ma ho appena trovato un lavoro interessante che promette di tenermi in stato 'on' per qualche tempo, quindi forse (ripeto: forse) c'è speranza per un futuro imprevedibile⁶.

TDF: Ultimo ma non meno importante... Come prevedi il futuro della comunità italiana dell'IF? E l'IF così come noi la conosciamo, potrà sopravvivere come hobby o è destinata all'estinzione in assenza di un vero e proprio mercato? Si sposterà sul Web o sui cellulari?

ENRICO: La mia sfera di cristallo è temporaneamente fuori servizio. Tuttavia, sono piacevolmente sorpreso dalla vitalità della comunità italiana relativamente alle sue dimensioni: la maggior parte dei suoi membri non dà segni di stanchezza, anche se la creatività sembra venire a sprazzi (com'è prevedibile); ho anche notato qualche interesse nello sperimentare nuovi concetti, sia nella forma letteraria che nella tecnologia. Il problema principale di questa comunità è probabilmente la sua piccola dimensione, anche se a volte appare qualche volto nuovo.

Io credo (e spero) che questa comunità sopravviverà, nonostante l'attrazione a pieni colori della grafica; ovviamente il numero relativamente piccolo di lettori (rispetto alla comunità inglese dell'IF) significa meno feedback e quindi meno incentivo per gli autori. È impossibile sopravvalutare l'importanza del feedback: qualcuno che ti dica: "Ehi, mi è proprio piaciuto il tuo lavoro" può davvero fare la differenza tra l'andare avanti a tutto vapore e l'abbandonare il progetto.

Per quanto riguarda l'IF in generale, credo che sopravviverà anch'essa, probabilmente nella stessa nicchia che occupa oggi, anche se la gente 'normale' non sembra più essere molto interessata a leggere. D'altro canto, il Web è un campo affascinante da esplorare (trovo strano che la sua potenzialità sia stata lasciata in gran parte inesplorata dagli autori e programmatori d'IF) e potrebbe di certo dare vita a qualcosa di valido. Sono meno sicuro riguardo ai telefoni cellulari, poiché sono tradizionalmente legati ad un uso frettoloso e superficiale.⁷

Una tecnologia che potrebbe rivitalizzare l'IF è sicuramente il riconoscimento vocale, che è sul punto di arrivare... e lo è stato per circa vent'anni. Forse l'IF potrebbe dargli un po' d'aiuto; sono ancora più sorpreso di trovare molta poca sperimentazione qui: pensavo a giochi audio-controllati fin dagli anni '80 e l'IF era sicuramente sulla mia lista. Non è necessario il pieno riconoscimento vocale per far funzionare un gioco d'avventura; al contrario, con un po' di fantasia si potrebbero usare gli errori di riconoscimento a proprio vantaggio.

Forse uno dei problemi maggiori è la rigida separazione tra gli artisti e gli autori, a causa della complessità crescente della tecnologia: avere il pieno controllo sia della creatività che della tecnica, come un artista dovrebbe saper fare, è diventato assai raro. È difficile creare un'opera d'arte in collaborazione, o limitando la creatività all'interno di uno strumento preconstituito, ma non è impossibile. Quindi... spero ancora di vivere abbastanza a lungo da vedere una nuova era creativa dell'IF.

TDF: OK, questo è tutto. Grazie Enrico per il tempo che mi hai concesso.

Ricordo a tutti i lettori che il sito di Enrico Colombini (dal quale provengono la maggior parte delle immagini qui presenti) è raggiungibile a questo indirizzo.



Aprile 2008, torredifuoco
traduzione di Vincenzo Scarpa

¹ Chiamate anche racconti-gioco. (†)

² Acronimo d'*Interactive Fiction*, il termine inglese che - nell'uso comune - si riferisce alle avventure testuali in generale. (†)

³ Adventure (conosciuta anche come Colossal Cave o ADVENT) è stata la prima avventura testuale della storia. Originariamente scritta nel 1976 da Will Crowther su un PDP-10 e successivamente ampliata da Don Woods, venne poi resa disponibile sulla rete ARPAnet (per la gioia di tutti coloro che non potevano accedere a questo mainframe). (†)

⁴ Enrico conferma che era proprio BdB e mi dice che in pizzeria si sono fatti una bella risata ricordando insieme quell'episodio. (†)

⁵ Acronimo di *Choose Your Own Adventure*: è il termine inglese riferito ai racconti-gioco. (†)

⁶ Enrico Colombini ha pubblicato di recente *Locusta Temporis*, un libro gioco della dimensione di un romanzo che, facendo uso degli ipertesti messi a disposizione dai formati ePub o pdf, riesce a simulare l'intelligenza artificiale di un software di avventure. Ulteriori informazioni potete trovarle [qui](#). (†)

⁷ Potrebbe essere invece molto interessante valutare con attenzione il mondo dei tablet computer (come ad esempio l'iPad) e degli smartphone (iPhone et simili). (†)