



Numero 7 – Ottobre 2005

Indice

Editoriale.....	3
Carta dei diritti del giocatore.....	4
Avventure e Cinema: proposte tecniche per un matrimonio.....	10
Reform: un decompilatore per Z-Code.....	13
Aaron A. Reed.....	16
Enigma.....	19
Little Falls.....	22
Little Falls.....	26
Little Falls.....	28
Radicofani.....	30
Gumshoe.....	32
INFORM: La libreria sgw.h per Glulx.....	36

Editoriale

di Roberto Grassi

Benvenuti al numero 7 di Terra d'IF.

Vorrei festeggiare con voi i due anni di attività di questa fanzine. Due anni particolarmente intensi, in cui speriamo di avervi fatto conoscere il mondo delle avventure testuali e della narrativa interattiva con maggiore profondità, per poter così comprendere e giocare meglio i giochi del nostro genere preferito. L'evento di questo mese è certamente la IFComp, la competizione internazionale più famosa per l'Interactive Fiction. Vi daremo i risultati e le traduzioni delle recensioni nel prossimo numero.

Come sempre, gli articoli sono di alto livello. Iniziamo con un bell'articolo di Francesco Cordella sulle relazioni tra la IF e il cinema. Proseguiamo con la traduzione in italiano di un classico: "La carta dei diritti del giocatore" di Graham Nelson, praticamente un decalogo per gli aspiranti disegnatori d'IF. Il terzo articolo inaugura un nuovo tipo di contributi: il focus su alcuni prodotti software che potrebbero interessare gli autori e i giocatori d'IF. In questo numero parliamo di Reform, un decompilatore di file per la Z-Machine.

L'intervista di questo numero è dedicata ad Aaron Reed, il vincitore della Spring Thing. È un'intervista molto interessante che contiene alcuni spunti molto importanti.

La rubrica "Design Review", riguardante il design delle AT, si focalizza questa volta con porte da aprire e chiudere, riprendendo un interessante thread di discussione nato sul NG. Avrebbe dovuto esserci la seconda parte del gioco "Fear", ma è rimandata al prossimo numero.

Come di consueto, il numero si chiude con le recensioni e con l'articolo di programmazione.

Carta dei diritti del giocatore

di Graham Nelson

traduzione di Paolo Vece

(estratto da *The Craft of Adventures* — Second Edition - G. Nelson, 1995)

In una versione iniziale di Zork era possibile venire uccisi dal crollo di una stanza instabile. A causa di una leggerezza nella pianificazione di tale crollo, 50000 libbre di roccia potevano crollarvi in testa mentre passeggiavate lungo un sentiero in una foresta. Senza dubbio meteoriti.

— P. David Lebling

W. H. Auden una volta ha osservato che la poesia fa accadere il nulla. I giochi di avventura sono molto più futili: non si deve dimenticare che importunano intenzionalmente il giocatore per la maggior parte del tempo. C'è una sottile linea tra la sfida e il fastidio: lo scrittore deve pensare, sopra ogni altra cosa, come un giocatore (non come un autore, e certamente non come un programmatore). Tenendo questo a mente, sostengo che i seguenti diritti siano inconfutabili:

1. Non venir ucciso senza avvertimento

Al livello più elementare, vuol dire che una stanza con tre uscite, due delle quali portano a morte istantanea e una porta a un tesoro, è irragionevole senza qualche suggerimento. A proposito del quale:

2. Non dover leggere suggerimenti orribili e per niente chiari

Molti anni fa, ho avuto a che fare con un gioco in cui, dirigendomi a nord di una caverna, si cadeva in una fossa letale. Il suggerimento era: c'era un gruppo di leoni scolpito sopra il passaggio. I buoni suggerimenti devono essere abilmente occultati, o essere molto brevi, ma non devo trovare soluzione dopo che l'evento è accaduto. (Il gioco era *Dungeon* della Level 9, nel quale "chi sta più in alto fa il botto più grosso quando cade" [in originale "pride comes before a fall" - traducibile letteralmente in "l'orgoglio arriva prima di una caduta" o "un gruppo di leoni arriva prima di una caduta" -, gioco di parole tra "pride" (gruppo di leoni, orgoglio) e "fall" (cadere) e il modo di dire che indica che a una persona che ha un'opinione troppo alta di se stessa, prima o poi le accade qualcosa che la fa apparire molto stupida]. Al contrario, il suggerimento al problema della piana delle rocce moventi in *Spellbreaker* è un capolavoro).

3. Essere in grado di vincere senza l'esperienza delle vite passate

Questa regola è molto difficile da rispettare. Ecco tre esempi:

1. c'è una bomba nucleare, sepolta da qualche parte sotto un terreno anonimo, che deve essere disinnescata. Il giocatore sa dove scavare perché, l'ultima volta che c'è stato, è saltato in aria lì.
2. c'è un lanciatore missili, con un pannello di pulsanti che sembra aver bisogno di essere correttamente programmato. Ma il giocatore può disattivare il missile giocando con i controlli prima di aver trovato il manuale.
3. (Questa viene da *The Lurking Horror*). C'è qualcosa che ha bisogno di essere cotto per la giusta quantità di tempo. L'unico modo per scoprire quanto tempo è provando e sbagliando, ma ogni partita consente una sola prova. D'altra parte, il buon senso suggerisce una soluzione ragionevole.

Di questi tre esempi, il numero (1) è chiaramente sleale; la maggior parte dei giocatori sarà d'accordo con il fatto che il numero (2) è abbastanza leale; mentre il numero (3), come capita con i casi veri, è proprio sul confine. In teoria, quindi, un buon giocatore dovrebbe essere in grado di risolvere un gioco senza ricorrere ad azioni illogiche, e ha quindi diritto ad:

4. Essere in grado di vincere senza sapere quali saranno gli eventi futuri

Ad esempio, un gioco che comincia vicino a un negozio. Avete solo una moneta e potete comprare una lampada, un tappeto magico o un periscopio. Cinque minuti dopo venite trasportati, senza alcun avviso, in un sottomarino, dove avete bisogno di un periscopio. Avete comprato un tappeto magico? Che sfortuna.

5. Non avere il gioco bloccato senza avvertimento

Intendendo con 'bloccato' l'impossibilità di proseguire più avanti nel corso del gioco. Se c'è una parete giapponese attraverso la quale è possibile passare proprio all'inizio del gioco, è estremamente irritante constatare che un enigma, quasi alla fine del gioco, richieda che la stessa sia ancora intatta, anche perché tutti i salvataggi che avete fatto sono inutili. In maniera simile è abbastanza comune avere una stanza che può essere visitata una sola volta per partita. Se devono essere fatte due cose differenti in quella stanza, dovrebbe esserci un avvertimento in tal senso.

In altre parole, un'azione irripetibile è giusta solo se al giocatore è stato dato un avvertimento che tale azione può essere irripetibile.

6. Non dover fare cose improbabili

Ad esempio, un gioco che richieda di domandare ad un poliziotto qualcosa di cui lui non sappia assolutamente nulla. (Meno estremizzato, il problema delle chiavi dell'hacker in *The Lurking Horror*). Un'altra cosa improbabile è attendere in un posto noioso. Se avete un incrocio a cui arriva, dopo cinque turni,

un elfo portando un anello magico, è difficile che un giocatore passi cinque turni consecutivi lì in attesa, e si perderà quello che pensavate fosse facile. (*Zork III* è il colpevole in questo caso). Se volete che il giocatore attenda da qualche parte, metteteci qualcosa di intrigante.

7. Non dover fare cose noiose solo per doverle farle

Nei brutti vecchi tempi, i giochi rendevano la vita difficile mettendo gli oggetti necessari a risolvere un problema a chilometri di distanza da dove si trovava il problema, in spregio ad ogni logica - cose tipo una barca al centro di un deserto. O, ad esempio, un enigma con una Torre di Hanoi a quattro dischi, giusto per gradire. E perché non una con otto dischi? E due degli enigmi più abusati - essere in grado di portare solo quattro oggetti, armeggiando con uno zaino, o aver bisogno di trovare nuove sorgenti di luce - possono veramente incrinare la pazienza di un giocatore molto rapidamente.

8. Non dover digitare l'unico verbo corretto

Ad esempio, "guardare dentro" un oggetto non trova nulla, ma "cercare" nell'oggetto trova qualcosa. O prendiamo in considerazione il dialogo che segue (sorprendentemente da *Sorcerer*):

```
> unlock journal
(with the small key)
No spell would help with that!
> open journal
(with the small key)
The journal springs open
```

[il dialogo è stato lasciato in inglese, perché con 'unlock' si intende quello che noi indichiamo con 'apri con la chiave', quindi la traduzione sarebbe:]

```
> "unlock" il diario
(con la chiave piccola)
Nessun incantesimo è in grado di aiutarti!
> apri il diario
(con la chiave piccola)
Il diario si apre
```

Questa cosa è così fuorviante da poter essere considerata un bug (un errore, un baco), ma è facile caderci mentre si progetta un'avventura. (Allo stesso modo, l'espressione richiesta per usare il mattone in *Zork II* la trovo poco leale, a meno che non abbia considerato qualcosa di ovvio). Consideriamo in quanti modi un giocatore può chiedere di levare un cappotto:

rimuovi il cappotto / leva il cappotto / svestiti del cappotto / lascia il cappotto /butta il cappotto / *cambia il capotto

(Ero scettico quando i play-tester mi avevano chiesto di aggiungere “don” [in-dossa] e “doff” [rimuovi] al mio gioco *Curses*, ma ho provato un momento di trionfo quando mia madre ha provato quei verbi la prima volta che ci ha giocato). I nomi, inoltre, richiedono di...

9. Avere un adeguato numero di sinonimi

Nella stessa sala in *Sorcerer* c'è un “tendaggio” che può essere chiamato “arazzo” (ma non “tenda”). Non si tratta di un lusso, è essenziale. Ad esempio, in *Trinity*, c'è una bella statua di un ragazzo spensierato che suona un flauto di pan. Può essere chiamata “bella” o “peter” “statua” “scultura” “flauto” “pan” o “ragazzo”. Gli oggetti, di solito, hanno più di una decina di nomi associati.

Forse qui si può introdurre una nota su un argomento più serio. La donna giapponese all'inizio di *Trinity* può essere chiamata “yellow” [gialla] o “Jap”, ad esempio, e sono dei termini che hanno un suono sinistro. Durante il play-testing di *Curses* mi è stato fatto notare che la frase “Let's just call a spade a spade” (uno scherzo innocente riguardante una vanga da giardino) voleva dire qualcosa di estremamente differente ai politici dell'estrema destra della parte più a sud degli Stati Uniti. [In italiano traducibile con “dire pane al pane”, con “spade” che vuol dire “vanga” o “picche”, mentre nello slang americano viene usata anche per indicare “negro” in termini spregiativi]. Alla fine la frase l'ho lasciata, ma da allora non l'ho più trovata così divertente.

10. Avere un parser decente

(Se solo questo fosse ovvio). Il parser dovrebbe essere almeno in grado di far raccogliere e lasciare oggetti multipli.

Visto che solo la Bibbia si ferma a dieci comandamenti, qui ce ne sono altri sette, anche se credo siano più dei punti di vista:

11. Avere un'adeguata libertà d'azione

Rimanere rinchiusi in una lunga sequenza di prigionie, con solo pochi momenti di libertà, non è affatto così divertente. Dopo un po' il giocatore comincia a sentirsi come se lo scrittore lo abbia legato a una sedia per gridargli la trama. Questa cosa è particolarmente pericolosa quando si ha a che fare con dei giochi di avventura che sono degli adattamenti di libri (e la maggior parte dei giocatori era d'accordo col fatto che i giochi della Melbourne House basati sul *Signore degli Anelli* soffrissero di questo problema).

12. Non dover fare troppo affidamento sulla fortuna

Piccole variazioni casuali aggiungono del divertimento, ma solo se sono piccole. Il ladro in *Zork I* sembra attenersi abbastanza a questa regola, così come la stanza girevole di *Zork II*. Ma un peso da dieci tonnellate che vi cade addosso uccidendovi a un certo punto in metà di tutti i giochi è particolarmente fastidioso. (Oltretutto vi state semplificando il lavoro, visto che per i giochi con elementi casuali è più difficile fare i test ed il debug, sebbene in un mondo ideale questo non dovrebbe essere un problema).

Con gli eventi a bassa probabilità c'è un pericolo particolare che può accadere, qualcosa che può distruggere le probabilità di un giocatore. Ad esempio, nella prima edizione di *Adventureland*, le api avevano un 8% di probabilità di soffocare per ogni turno in cui venivano portate nella bottiglia: era necessario portarle per 10 o 11 turni, il che dava alle api il 40% di probabilità di sopravvivere fino alla loro destinazione.

Ci sarebbe molto da dire sul variare i messaggi che capitano molto spesso (come in "Consultate il vostro libro degli incantesimi") in un modo abbastanza casuale, per il bene della varietà.

13. Essere in grado di capire un problema, dopo che è stato risolto

Potrebbe sembrare ovvio, ma molti problemi vengono risolti per caso o dopo una serie di "prova e sbaglia" [trial and error]. Un posto di guardia che può essere passato se, e solo se, possedete una lancia, ad esempio, dovrebbe indicare in qualche modo che è questo il motivo per il quale siete passati. (L'esempio più estremo è la famosa Banca di Zork, per la quale non ho mai capito le spiegazioni delle altre persone).

14. Non dover subire troppi depistaggi

Qualche depistaggio rende il gioco più interessante. Una caratteristica molto simpatica di *Zork I, II e III* è che ognuno contiene dei depistaggi spiegati negli altri (in un caso spiegato in 'Sorcerer'). Ma gli enigmi più difficili tendono a essere risolti per ultimi, e la tecnica principale che i giocatori usano è quella di consultare le loro mappe alla ricerca di cos'è rimasto tra ciò che non hanno capito. Questo è frustrante quando ci sono molti enigmi insoluti e molti oggetti inutili. Potete, quindi, aspettarvi che i giocatori perdano interesse se non state attenti. Il mio punto di vista è che i depistaggi debbano essere spiegati: ad esempio, se c'è una noce di cocco inutile all'inizio del gioco, allora, più avanti, dovremmo trovare un botanico un po' svampito che va in giro perdendo noci di cocco. Almeno la noce di cocco ha un motivo per essere lì.

Un oggetto non è un depistaggio solo perché non ha una funzione nel gioco: un quotidiano inutile può essere ragionevolmente trovato in una biblioteca. Ma non un caleidoscopio.

Il peggior gioco che abbia mai provato per i depistaggi è *Sorcerer*, che per i miei calcoli si merita un bel 10.

15. Avere una buona ragione per la quale qualcosa è impossibile

A meno che sia anche divertente, una spiegazione molto contorta del perché qualcosa sia impossibile da fare è irritante. (Penso che la ragione per la quale non sia possibile andare sull'erba di Kensington Gardens in *Trinity* sia solo abbastanza divertente).

Un'obiezione morale, invece, è giusta. Ad esempio, se siete a casa del vostro

migliore amico, dove c'è un diamante in una vetrina, rompere la vetrina per prendere il diamante sarebbe fisicamente semplice, ma completamente fuori luogo. Possiamo risparmiare al signor Spock di sparare nella schiena del capitano Kirk.

16. Non dover essere Americano¹

Il **diamond maze** (labirinto a forma di diamante) di *Zork II* è un ottimo esempio. In maniera simile, è una gentilezza quella di consentire al giocatore di inserire parole con l'ortografia americana o inglese. Ad esempio *Trinity* si accattiva la simpatia dei giocatori inglesi facendogli chiamare il pallone da calcio ("soccer ball") anche "football" - visto che "soccer" è una parola che non si usa quasi mai in Inghilterra. (Da quando queste parole sono state scritte, diverse persone mi hanno gentilmente fatto notare che il mio gioco *Curses* è, per così dire, decisamente Inglese. Ma, come ogni buon dittatore, preferisco scrivere le costituzioni che adeguarmi ad esse).

17. Sapere come sta evolvendo il gioco

In altre parole, sapere quando si avvicina la fine, o come si sta sviluppando la trama. Tanto tempo fa il punteggio era l'unico modo per saperlo, speriamo non lo sia più.

¹ Estendendo la regola al mercato internazionale potremmo dire: non dover essere di nessuna nazionalità particolare [NdT]

Avventure e Cinema: proposte tecniche per un matrimonio

di Francesco Cordella

Può sembrare incredibile, eppure è così: non esistono avventure testuali tratte da film. Ma la cosa davvero sorprendente è che, per quanto mi risulta, non esistono neppure avventure testuali chiaramente ispirate a un film, nell'intreccio, nei personaggi, nell'ambientazione. E questo, se vogliamo, è ancora più strano.

L'unico esempio, ma si tratta di un'avventura grafico-testuale, me lo ha ricordato Paolo Vece, frequentatore del newsgroup it.comp.giochi.avventure-testuali: *Dallas Quest* (Datasoft, 1984). Tuttavia, oltre a non essere un'avventura testuale, è un gioco ispirato a un serial televisivo, non a un film. Divertente, a quanto ricordo, però senza spunti particolarmente originali e con scarsissima attinenza alla trama reale della fiction americana.

Diverso il discorso per le avventure grafiche: qui abbiamo la serie *Indiana Jones* (iniziata nel 1992) e *Blade Runner* (Westwood, 1997). Soltanto due titoli, comunque.

Quanto ai telefilm, abbiamo *X-Files* (Ubisoft, 1998) e *C.S.I.* (Ubisoft, 2003).

Tutti gli altri esempi, che in un primo momento possono venire in mente, in realtà non sono avventure tratte da film, bensì da libri: *Dracula*, *Frankenstein*, *Lord of the Rings*, *The Hobbit*, *Sherlock*, *Il mago di Oz* e il bellissimo *Wonderland*.

Dunque, la domanda è: perché non esistono avventure testuali se non tratte, almeno vagamente ispirate a un film? Risposta scontata: perché è reato scrivere un'avventura tratta da un film senza comprarne i diritti e, allo stesso tempo, comprare i diritti è enormemente dispendioso e per niente redditizio. Forse solo la Infocom nell'età dell'oro avrebbe potuto permetterselo e comunque avrebbe rischiato il fallimento anzitempo perché, in genere, la posta non vale il rischio quando si parla di investimenti che riguardino giochi solo testuali.

I casi di *Indiana Jones* e *Blade Runner* sono eccezionali. Innanzi tutto, si tratta di giochi grafici e, lo sappiamo, il budget è decisamente superiore per questo tipo di prodotti. Poi, stiamo parlando di due film di enorme successo che ben si prestano a un adattamento avventuroso e che (almeno in teoria) assicurano un certo ritorno economico nei negozi (anche se *Blade Runner* fu un mezzo fallimento).

Detto questo, non è ozioso, e di certo è divertente, chiedersi: quali film sarebbero adatti a diventare un'avventura testuale (non grafica)? Quali offrono spunti interessanti? Quali sembrano adatti, ma in realtà non lo sono? Ho buttato giù una lista, l'ordine è dal film più "adattabile" al meno

“adattabile”. Ovviamente, vista la monumentale vastità dell’argomento, non potrà mai essere completa. Ma può essere un punto di partenza. Eccola.

Paycheck (John Woo, 2003). Ha una trama che fonde thriller e fantascienza, un protagonista che ha perso una parte della memoria (un “classico” delle avventure) e, soprattutto, un risvolto prettamente “avventuroso”: il personaggio principale si ritrova con dieci oggetti (uno spray, un accendino, un anello ecc.) e deve scoprire il modo di usarli per raggiungere un obiettivo.

Fuori Orario (Martin Scorsese, 1985). Il protagonista è un tipico personaggio avventuroso: è un uomo comune e, soprattutto, un “blank hero” (definizione di Roger-Giner Sorolla nel saggio *Crimes against mimesis*), cioè un “eroe senza passato, senza una storia personale”, ma semplicemente una “pedina” da manovrare. Il che non è molto in linea con i canoni della narrativa interattiva, che richiede una storia, se non addirittura un’ introspezione psicologica del protagonista, però potrebbe dare un retrogusto da “vecchia scuola”. Oltre a questo, *Fuori Orario* è una straordinaria avventura nella notte, tra personaggi ambigui e bizzarre situazioni, con un preciso traguardo da raggiungere: il ritorno a casa.

Rain Man (Barry Levinson, 1988). Convertito in avventura, questo popolarissimo film si presterebbe a uno spunto ancora poco usato nelle avventure: il cambio di personaggio. Sarebbe interessante, adattando *Rain Man*, potere impersonare, in base alla situazione, Tom Cruise o Dustin Hoffman, nel corso del loro viaggio “on the road”. *Rain Man*, poi, offre molti luoghi da visitare. Più difficile sarebbe ricavarne degli enigmi: ma potrebbe essere un’avventura esplorativa e “dialoghistica” più che enigmatica.

Saw — L’Enigmista (James Wan, 2004). Se *Rain Man* offre lo spunto del cambio di personaggio, *Saw* sarebbe perfetto per costruire un’avventura che si svolge, in parallelo, in due scenari. Il primo: quello dei due tizi che si ritrovano in una “cella” con un cadavere nel mezzo. Il secondo: lo scenario esterno, con i flashback del serial killer che uccide e il presente della polizia che gli dà la caccia. Il giocatore PC (noi alla tastiera) potrebbe essere proprio il serial killer che pone enigmi ai due tizi rinchiusi, uccide in alcuni flashback e sfugge alla polizia nel presente. Ne verrebbe fuori un gioco interessante e originale.

La 25ma ora (Spike Lee, 2002). La storia di Monty Brogan-Edward Norton che passa la sua ultima giornata prima di andare in carcere è perfetta per un’avventura scandita dallo scorrere del tempo. Ognuna delle ventiquattro ore potrebbe essere un segmento, un capitolo del gioco, con due o tre enigmi da risolvere (l’indagine sulla fidanzata sospetta, la fuga dai mafiosi). In più ci sarebbero i flashback. Con un “rush” finale, il segmento della 25ma ora, in cui, per sfuggire alla giustizia, bisogna procurarsi i documenti per cambiare identità e attraversare il deserto per rifarsi una vita.

Nodo alla gola (Alfred Hitchcock, 1948). Perfetto esempio di One Room Game, dato che Hitchcock girò questo film come un unico piano sequenza, creando un effetto teatrale. Il giocatore PC potrebbe essere non tanto il

“detective”, quanto uno dei due giovani, magari il più buono, che devono passare una serata a casa con un cadavere nella cassapanca e circondati da ospiti. Gli enigmi? Beh, si può creare qualcosa di originale: per esempio, escogitare vari modi per distrarre gli ospiti e non farli curiosare nella cassapanca. In più, ci dovrebbero essere enigmi nei dialoghi: magari con una conversazione a menu, bisogna rispondere in maniera corretta alle pressioni degli ospiti più sospettosi, come Rupert Cadell-James Stewart.

Memento (Christopher Nolan, 2000). In tanti, anche all'estero, hanno pensato a Memento come film convertibile in gioco. In effetti, di spunti ne offre. Il film si svolge in una serie di segmenti, che corrispondono ai ricordi del protagonista, il quale perde la memoria ogni dieci minuti. Ogni segmento è il pezzo di una storia, che parte dalla fine: e ogni segmento, come nel caso della 25ma ora, può essere un buono scenario avventuroso. Dalla perquisizione della casa ai documenti da recuperare.

Chiudo questa lista citando altri tre film. Il primo è **Il delitto è servito (Jonathan Lynn, 1985)**. Ispirato al gioco da tavolo Cluedo, questo film propone lo stesso scenario della storia interattiva *Deadline* (Infocom, 1982): un gruppo di persone in una casa in cui viene commesso un delitto. Ma ha una vena ironica, che permetterebbe di creare una variante interessante dell'avventura Infocom. Il secondo film è **The Game (David Fincher, 1997)**, cui è liberamente ispirato anche il mio *Flamel*. Le situazioni incredibili in cui si trova invischiato il protagonista, vittima di un terribile scherzo, offrono enigmi a go-go. Infine, come ultimo film, segnalo **Arancia Meccanica (Stanley Kubrick, 1971)**. Ebbene sì, questo capolavoro sarebbe una fantastica avventura, anche se guidare una banda di assassini-psicopatici non è proprio edificante.

La lista finisce qui, anche se decine di altri film meriterebbero di essere inseriti, a cominciare da *Shining* (Stanley Kubrick, 1980) a *The Hitcher* (Robert Harmon, 1986).

In coda, propongo un'appendice sui film che potrebbero sembrare convertibili in avventure, ma in realtà a mio parere non lo sono. Qualche esempio?

Million Dollar Hotel (Wim Wenders, 2000). Il fatto che si svolga in un hotel funestato da un omicidio potrebbe fare venire l'acquolina in bocca, però in realtà nel film l'indagine è pari allo zero, mentre i dialoghi, i silenzi e i movimenti di macchina hanno il sopravvento.

Cube (Vincenzo Natali, 1997). Anche questo film è stato indicato da molti come perfetto per costruire un'avventura. Ma mi spiegate quanto interessante sarebbe un gioco che si svolge in stanze tutte uguali, senza oggetti, con enigmi legati a un solo calcolo matematico? Il film è bello, trasformarlo in avventura gli farebbe un torto.

Pulp Fiction (Quentin Tarantino, 1995). Il fatto che proponga varie storie, autonome ma allo stesso tempo intersecate, potrebbe invogliare. Ma *Pulp Fiction* è un film che ha la sua forza soprattutto nei dialoghi: come nel caso di *Cube*, trasformarlo in avventura gli farebbe un torto.

Reform: un decompilatore per Z-Code

di Roberto Grassi

Reform è un decompilatore — scritto da Ben Rudiak-Gould — per i file della Z-Machine; esso produce file di testo con una sintassi molto simile al codice sorgente di Inform. Supporta tutti i formati della Z-Machine (compresi quelli della versione 6 della Infocom) e può anche decompilare la porzione esclusivamente testuale dei file BLORB.

Vederlo all'opera è molto semplice.

Scaricatelo dal sito del programmatore (esistono le versioni per Windows e Linux¹): <http://www.darkweb.com/~benrg/if-decompilers/> e decomprimete il file zip².

Nel caso più semplice di utilizzo non dovrete far altro che eseguire, a linea di comando:

```
reform storyfile.z5 > output-file
```

Ad esempio, usando il file .z5 di Little Falls (Alessandro, perdonami! :D)

```
reform LittleFalls.z5 > LittleFalls.txt
```

Estratti del file:

PORZIONE INIZIALE

```
! "LittleFalls.z5", 155648 bytes
! Z-machine version 5, release 1, serial 050527
!
! Decompiled by Reform release 6, serial 040226

! Grammar

Verb 'origlia' 'osserva' 'spia'
  *                               -> LookIn
;

Verb 'calcia'
  * noun                          -> Kick
;
```

1 Per scrivere questo articolo ho usato la versione per Windows.

2 Nella distribuzione trovate un ottimo file reform.html che potrete usare come guida.

```

Verb 'slega'
* noun                               -> Slega
;

Verb 'esplora' 'indaga' 'investiga' 'perlustra' 'perquisis'
* noun                               -> Search
;

```

PORZIONE CENTRALE

```

.label55537;
  if (global49 == 1[?]) {
    ! no code
  } else {
    if (global46 & 2) {
      routine54736(2 * (local2 + global50));
    }
    if (local9 == 3[Routine?]) {
      local13 = 0;
      local10 = 0;
      .label55575;
      if (local10 < global49) {
        local13 = local13 + local5->(local10 + local6);
        ++local10;
        jump label55575;
      }
      routine53616(local13);
      print " ";
      global239 = local7.list_together;
      print (string) global239;
      if (global46 & 8 ) {
        print " (";
      }
      if (global46 & 2) {
        print ":";
      }
    }
    local13 = global46;
    if (local9 ~= 3[Routine?]) {
      global51 = 1[?];
      global76 = local7;
      global77 = local2 + global50;
      if (routine51272(local7,25[Menu?]) == 1[?]) {
        jump label55738;
      }
    }
  }

```

PORZIONE FINALE

```

! 114312 (0x1be88): "La porta principale @`e chiusa. Sembra stata
                  cambiata da poco. @`E come nuova."
! 114368 (0x1bec0): "Mmm... La porta dell'ingresso @`e socchiusa.

```

```
~Strano~, pensi. ~Un istante fa sembrava chiusa.~^"  
! 114444 (0x1bf0c): "La porta principale @`e socchiusa.^"  
! 114472 (0x1bf28): "Provi con una spallata, ma la porta @`e  
blindata. ^Ti rendi conto subito che @`e pressoch@'e  
impossibile sfondarla.^"  
! 114556 (0x1bf7c): "Ti avvicini alla porta, inserendo le chiavi  
nella serratura, ma prima che tu riesca a chiudere la  
porta, senti una fitta sulla testa."  
! 114648 (0x1bfd8): "^Qualcuno ti ha colpito con il calcio di una  
pistola. Cadi pesantemente a terra."  
! 114708 (0x1c014): "^Una donna torreggia sopra di te con la tua  
Beretta in pugno.^"  
! 114756 (0x1c044): "Una camera interna. Sembra uno studio."  
! 114784 (0x1c060): "^@`E elegantemente arredato con un grande  
scrittoio e una libreria. Una finestra consente di  
vedere all'esterno. "  
! 114864 (0x1c0b0): "^L'uscita @`e verso est, al salone.^"  
! 114892 (0x1c0cc): "L'unica direzione in cui puoi muoverti @`e a  
est (verso l'ingresso). Oppure fuori dalla finestra...  
Ma non riusciresti a passare attraverso le inferriate."  
! 115004 (0x1c13c): ", collega lo studio con l'ingresso.^"
```

In sostanza, Reform può essere utile per andare ad analizzare un sorgente di un file per motivi di studio ed è senza dubbio uno dei migliori decompilatori per file Z-Code.

Aaron A. Reed

intervistato da Roberto Grassi

Aaron, parlati di te.

Mi sono laureato da poco all'Università dello Utah nella facoltà di cinematografia e mi sto muovendo per fare la mia carriera in campi che non hanno niente a che vedere con i film. Oltre ad occuparmi di fiction interattiva, mi piace anche scrivere la classica fiction lineare — una mia storia sulla realtà virtuale è apparsa sulla rivista “Fantasy and Science Fiction Magazine” - e ho un vasto catalogo di altre cose che potrebbero essere riviste e pubblicate.

Gli altri miei interessi spaziano dall'astronomia al gioco di ruolo, dall'escursionismo alla speleologia (quest'ultimo mi è derivato dal fatto di aver giocato ad *Adventure* da piccolo ed esserne rimasto affascinato). Gli amanti dell'IF, o i curiosi, potrebbero essere interessati alle mie foto della piccola valle al di fuori dell'entrata della caverna, in cui era basata la sezione del gioco non sotterranea.



Parlati dei tuoi lavori d'Interactive Fiction.

Sono abbastanza vecchio da aver sperimentato la fase finale dell'IF come fenomeno commerciale — ricordo di aver acquistato *Hitchhiker's Guide to the Galaxy* e di essere rimasto incantato dalle confezioni dei giochi della Infocom che spiccavano sugli scaffali. Qualche anno fa sono ritornato a interessarmi dell'IF moderna e mi ha colpito il fatto che la sua freschezza e unicità come forma d'arte non erano ancora state esplorate completamente. È anche unico che una persona sola possa creare l'intero gioco, qualcosa di virtualmente impossibile in un gioco per computer basato sulla grafica.

Nelle mie storie, sono interessato ad esplorare tipi di storie che sono una peculiarità di questo mezzo espressivo. *9:05* di Adam Cadre è una storia che

non funzionerebbe affatto come film, piece teatrale o storia breve: la connessione tra il giocatore e il protagonista e la rottura delle convenzioni dell'IF la rendono un'esperienza unica e affascinante. Ho cercato di creare qualcosa di simile con il mio ultimo gioco *Whom the Telling Changed* e anche se molti possono dire che io non sia riuscito nel mio intento (compreso me stesso) ho imparato molte cose sulle possibilità del mezzo durante la scrittura. Più di tutto sono interessato a provare cose nuove e vedere quello che si può e non si può fare con il mezzo.

Cosa ne pensi dei vecchi giochi dell'Interactive Fiction?

Come la fiction normale, quella interattiva consiste in una collaborazione tra le parole dell'autore e l'immaginazione del lettore. Le cose migliori dell'IF si muovono secondo questa regola allo stesso modo in cui Giulio Verne, J.R.R. Tolkien e i grandi autori hanno fatto: l'effetto magico si realizza quando senti di trovarti lì, insieme con i personaggi, o magari uno di loro. Esplorare tombe, navi aliene o castelli abbandonati, i primi giochi si caratterizzarono per i plot fragili e gli NPC deboli ma catturavano il giocatore con l'emozione della scoperta.

Con un occhio più saggio e critico, il mio gioco preferito (tra i classici) è *Trinity*, perché rompeva la tradizione in un modo nuovo ed interessante.

Che cosa ne pensi della scena attuale?

Non gioco tanto quanto dovrei. Gli ultimi giochi che mi sono piaciuti sono *Slouching Towards Bedlam*, perché si confaceva ai miei gusti in quanto ad ambientazione, stile di gioco ed idee, e *1893: A World's Fair Mystery* che mi ha commosso per la quantità di dettagli e di lavoro di cui è stato oggetto - devo ancora riprenderlo e terminarlo. Mi piacciono molto i giochi di Adam Cadre (e il suo libro "Ready, Okay!" è anche molto bello).

Penso che sia un ottimo momento per essere un fan dell'Interactive Fiction - per poter proporre esperimenti, scoperte e discussioni nella definizione di un nuovo genere di narrazione. La natura intimistica e raccolta della comunità di affezionati all'IF è anni luce avanti rispetto alle forme comuni di intrattenimento interattivo offerte, ad esempio, dai giochi per PC. Credo che le lezioni che si stanno apprendendo nel mondo dell'IF su ciò che è possibile o che non è possibile fare con questo tipo di medium stiano formando le colonne portanti per i prossimi cento anni, o più, di divertimento.

Dove sta andando la IF, secondo te?

"Per quanto tempo l'IF sopravviverà come medium particolare?". È la domanda che tutti si pongono. La sua semplicità (testo in entrata, testo in uscita), e il fatto che utilizza della tecnologia che non è legata particolarmente all'innovazione, mi fa ipotizzare che sarà ancora tra di noi per molto tempo, anche se forse non diventerà mai popolare. La velocità computazionale crescente e gli avanzamenti nell'intelligenza artificiale apriranno nuove strade di ricerca nell'ID, in particolare nel dominio dell'interazione con altri personaggi, senza dubbio il punto più debole nel medium, al giorno d'oggi.

Tempo fa scrissi un semplice transcript di una sessione IF con un parser senziente, come esperimento per dimostrare le limitazioni che sono assunte come standard nelle interfacce odierne. Quando il giocatore potrà comunicare

con il gioco in un modo complesso tanto quanto il gioco comunica con lui, le possibilità che si potranno aprire sono enormi.

Stai lavorando a qualche nuovo progetto?

Poiché scrivere IF per i termini di consegna delle competizioni (la cosa, tra l'altro, è l'unico motivo che mi spinge a finire i giochi) è un processo estenuante, di solito ci metto molto tempo a raccogliere le motivazioni per iniziare un nuovo progetto. Ho in mente alcune idee, ma nessuna di esse è ancora ben sviluppata — saremo senza dubbio nel 2006 prima che esca ancora qualche mio gioco.

Il mio sogno — sai, quello che ognuno ha e che ogni volta sembra troppo scoraggiante da affrontare — riguarda la “Mammoth Cave”, la sua storia e la sua connessione con l'IF. Non divulgherò i dettagli del plot e le meccaniche del gioco, ma rimarrò più sul generico e vi dirò che riguarda le relazioni tra la fiction, l'esplorazione e il bisogno di saziare la curiosità. Ci sono così tante storie affascinanti collegate con quel sistema di caverne, e la caverna stessa è così fuori dal tempo - ancora immutata da quando ognuna di quelle storie è accaduta - che sembra una tela affascinante su cui dipingere i temi fondamentali della natura umana.

Per quanto riguarda i miei progetti più a breve termine, sto collaborando con un amico programmatore a una specie di giallo 'high-tech', con un congegno tecnologico che può consentire delle belle varianti a vecchie convenzioni. Immaginate fotografie della scena del crimine che possono essere portate avanti e indietro nel tempo. Una recente discussione sul newsgroup ha riguardato la gestione convenzionale del flusso del tempo in un'IF. Come fai a interrompere qualcosa che accade nel mezzo flusso del testo tra un input del giocatore e l'altro? È possibile riscrivere il testo così che le tue descrizioni finiscano nel mezzo di un'azione anziché alla fine?

Per concludere, comunque, il mio desiderio è quello di condividere con le persone storie che li facciano pensare o far provare loro sensazioni che non avrebbero potuto provare altrimenti.

Auguratemi buona fortuna!

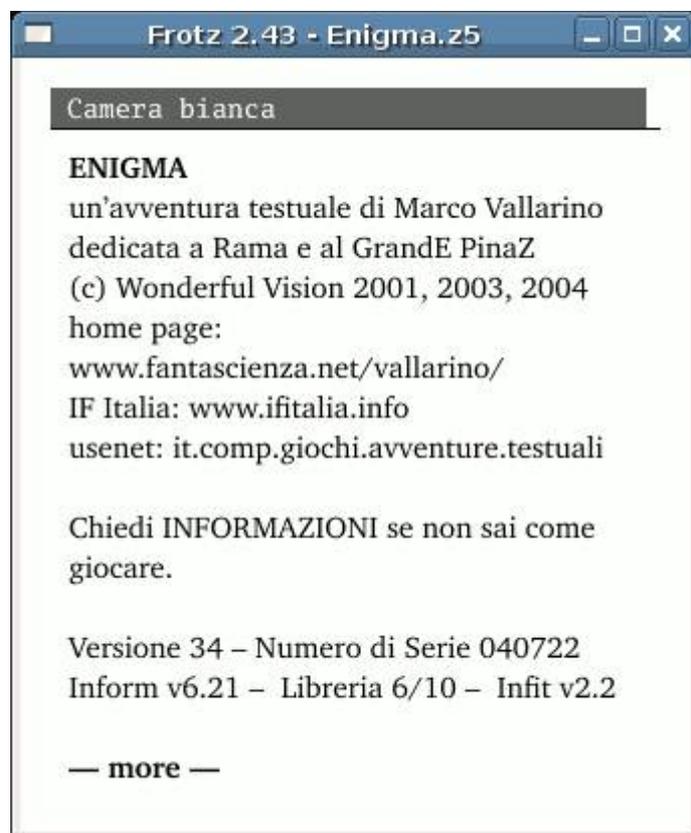
Enigma

di Marco Vallarino

recensita da Roberto Grassi e Paolo Lucchesi

Quando ci si mette a giocare a *Enigma*, specialmente per scrivere una recensione, l'aspettativa è naturalmente molto alta, dato che si tratta dell'AT italiana più scaricata (e presumo giocata) in assoluto e che ha inoltre vinto il premio avventura del 2000 davanti a *Flamel* (pregevolissima anch'essa, fra l'altro).

Forse non siamo riusciti a coglierne il fascino ma, purtroppo, è stata per diversi aspetti una grossa delusione. Il gioco a nostro parere non soddisfa nessuno dei requisiti standard condivisi per definire un'AT di 'qualità'.



Anzitutto, l'avventura è realizzata con una cura assolutamente non ineccepibile. Molti oggetti che sono nominati nelle descrizioni delle stanze, ad esempio, non risultano essere in realtà esistenti. Ancora, l'autore non ha personalizzato i messaggi standard di Inform, per cui se canti "sei stonatissimo" e volendo "salti sul posto, senza risultato"; questo porta anche ad involontari effetti semi-comici per cui guardandoti "hai sempre lo stesso bell'aspetto" ma guardandoti allo specchio "non hai una bella cera".

I personaggi non giocanti poi, sono monodimensionali, funzionali solo al loro compito di “ostacolo” e niente li differenzia da altri oggetti inanimati; esistono per uno scopo, non fanno altro e non reagiscono minimamente a qualsiasi stimolo del giocatore che non sia quello necessario per procedere nel corso del gioco. Uno di loro, addirittura, “non sembra interessato” quando gli viene mostrato l’oggetto col quale andrà corrotto poche mosse dopo.

Dal punto di vista “letterario” *Enigma*, fedelmente al suo nome, offre poco o nulla. La trama è estremamente esile ed è chiaramente soltanto un espediente per proporre al giocatore una serie di problemi e, appunto, enigmi. Detto questo, possiamo anche perdonare il fatto che la storia abbia un *ché* di “visto e rivisto, trito e ritrito”.

Ma se poi si analizza *Enigma* sotto i parametri dei “crimini contro la mimesi” (dall’omonimo articolo di Roger Giner-Sorolla) mi sembra evidente che siamo in presenza di “enigmi fuori dal contesto”, “oggetti fuori dal contesto” e “contesti fuori dal contesto”. Quello che ne risulta è una mescolanza di elementi diversi malamente amalgamati che non può non riuscire indigesta. Se poi si analizza il gioco dal punto di vista della lista dei diritti del giocatore (articolo di Graham Nelson che trovate in questo stesso numero) si vede anche che molti dei punti che Graham suggerisce sono violati.

E dunque? Cosa resta?

Resta che si potrebbe dire: “E’ un’AT che *non ha* trama e contesto per una scelta precisa da parte dell’autore” (anche se alla fine del gioco si dovrebbe capire perché abbiamo incontrato quei PNG e perché ci siamo mossi in quelle locazioni).

Ma a questo punto vengono spontanee due considerazioni.

La prima è che creare un’AT *senza* contesto è un facile alibi per poter giustificare qualunque cosa e questo a nostro modo di vedere è più una debolezza che una giustificazione.

La seconda è che è possibile creare AT (nel senso di avventure testuali ‘old style’) con un contesto ben preciso senza che questo possa costituire un limite alla fantasia, alla giocabilità e al design degli enigmi. Riguardo questo aspetto, invito tutti a giocare al meraviglioso “The Isle of the Cult” rilasciato l’anno scorso.

Lasciando da parte anche queste considerazioni, permane un ultimo problema: *Enigma* delude parzialmente anche secondo quello che dovrebbe essere il suo punto di forza. Gli enigmi che si incontrano risolvendo il gioco sono abbastanza semplici e tutto sommato scadenti e banali. Anzitutto, la maggior parte di essi segue lo schema “get x - use x”: bisogna trovare un oggetto che servirà (unicamente, poi potrà essere buttato via) a risolvere un enigma, che a sua volta ci fornirà un nuovo oggetto per risolvere un altro enigma...

Anche i pochi puzzle che escono da questo stile non brillano certo per originalità: un banale crittogramma, un paio di indovinelli, il solito labirinto. Si salvano soltanto il problema del foglio da leggere (ma chi conosce il latino è avvantaggiato) e l’enigma numerico finale; ma è comunque poco, si sente la mancanza di enigmi più articolati, forse più difficili ma capaci di dare ben più soddisfazione al giocatore che finalmente li risolve (il pesce babele vi ricorda qualcosa?).

In conclusione, rispetto agli standard “internazionali” *Enigma* è abbastanza

deludente.

Rispetto agli standard “italiani” è un’AT che vorrebbe ricalcare gli stili e il fascino delle vecchie AT della scuola italiana ma che non riesce a farlo (mi sembra, tra l’altro, che le avventure di BdB fossero estremamente attente al “contesto”).

Pro: breve

Contro: trama assente, fuori contesto, enigmi fiacchi

Little Falls

di Roberto Grassi e Alessandro Schillaci

recensita da Francesco Cordella

Little Falls è, innanzitutto, un ottimo esempio di Glulx non solo per l'Italia ma anche per il resto del mondo (esperimento finora provato da noi solo da Tommaso Caldarola con il suo *Skizo*). *Little Falls* ha una buona grafica, fatta di bei disegni e di quelle che sembrano rielaborazioni di foto con un interessante ammanto sfocato, e un contorno di musiche ed effetti sonori che contribuiscono a “fare atmosfera”. *Little Falls*, dunque, in qualche modo consacra l'utilizzo di Glulx nel panorama italiano: e si vede che gli autori, dal disegnatore, al programmatore e agli ideatori dei testi hanno fatto un grosso sforzo. Indubbiamente, la sequenza iniziale è molto bella, crea suspense e sembra l'inizio di un telefilm. Complimenti. Lo stesso effetto suspense lo dà il sonoro, che contribuisce a rendere coinvolgente il gioco.



Per certi versi, e da un certo punto di vista tutto è un pregio, *Little Falls* rievoca l'atmosfera delle avventure grafico-testuali anni Ottanta: immagini, descrizioni brevi ed enigmi a ripetizione. Avrà di sicuro successo perché ha un impianto classico, molto amato in Italia, con tutti gli ingredienti della “vecchia scuola”. Con un ulteriore valore aggiunto: enigmi ben legati all'intreccio. E non è poco, davvero.

A un giocatore abituato alla narrativa interattiva nata negli ultimi anni in America, tuttavia, non basta un bellissimo salto nel passato: un giocatore abituato alla narrativa interattiva vuole una storia solida, meglio ancora se costruita o montata in modo originale. E a mio avviso quella di *Little Falls* è una storia debole che nel finale si ingarbuglia e diventa piuttosto confusa. Ho apprezzato il gusto dei dettagli (come i codici di emergenza della polizia), ma non ho trovato originalità. Prima di tutto: il poliziotto con un incubo del passato, scatenato da un errore durante un'azione, è un classico che più classico non si può. Uno stereotipo. Da questo punto di vista, dunque, *Little Falls* ha poco di “innovativo” (anche se per essere bello un gioco non deve necessariamente essere “innovativo”, sia chiaro).

Ma quello che davvero non mi convince è la storia.

L'esperimento provato in *Little Falls* consiste nell'inserire due intrecci “incastrati” in una sola storia. Primo intreccio: Sonja vuole vendicarsi del

poliziotto Gene e studia un piano per farlo finire in galera. Secondo intreccio: Tom, amico del poliziotto Gene, è un serial killer, mentre Sonja è “solo” una vittima di questo serial killer.



Per ognuno di questi due intrecci sono previsti diversi finali (in tutto i finali sono sette). A mio avviso, il problema è che i due intrecci non funzionano autonomamente. Mi spiego: nella storia di Little Falls i due intrecci procedono paralleli fino a un certo punto: l'arrivo di Tom nella casa. E' questo il punto focale: quando arriva Tom nella casa, la storia può prendere strade diverse e sfociare nell'uno o nell'altro intreccio. Ma analizzati *autonomamente* i due intrecci hanno diversi problemucci. Alla fine, insomma, io lettore-giocatore mi faccio troppe domande, davvero troppe. Infatti, solo dopo un fitto scambio di mail con Roberto Grassi mi sono chiarito alcuni passaggi. Mai ci sarei riuscito giocando e rigiocando a meno di non mettermi a fare fumare il cervello di congetture. Ma analizziamo i due intrecci:

SONJA E' IL FANTASMA DEL PASSATO DI GENE E VUOLE VENDICARSI. Nel finale, si dice che “Sonja ha studiato il piano nei “minimi particolari” eppure il suo piano prevede troppe variabili rischiosissime. Mettiamoci nei suoi panni quando studia il piano. Allora: innanzi tutto, devo attirare e uccidere Kelly, la moglie di Gene, in quella casa, sperando che né Gene né Tom in quel lasso di tempo si preoccupino per lei. Poi, devo attirare Gene nella casa. E che faccio? Anziché, magari, farlo chiamare dalla moglie messa sotto minaccia o attirarlo con un altro stratagemma, per farlo venire fino a lì faccio una cosa rischiosa: chiamo il 911 e spero che sia Gene ad arrivare nella casa e non un altro poliziotto che potrebbe mandare un piano studiato in cinque anni (cosa *probabile* benché Sonja, si presume, abbia studiato i movimenti di Gene per cinque anni). Poi, mando un sms a Tom, lo invito in questa casa per lui sconosciuta dicendogli che troverà le chiavi fuori dalla porta (questo delle chiavi è un elemento sottinteso che mi ha palesato Roberto ma che dalla storia non si evince), spero che Tom non si insospettisca con un sms del genere, spero che si trovi non troppo distante dalla casa, spero che non gli sembri strano prendere le chiavi e aprire la porta da sé (certo, l'ho invitato io, dovrebbe suonargli strano che gli dica di prendere le chiavi, ma speriamo che lo faccia), spero che Gene si liberi e che Tom, entrando, non lo veda lì legato e chieda “che succede, Gene?”, spero che, una volta liberatosi, Gene lo ammazzi e non provi a parlargli dicendogli “Tom, che ci fai qui, bastardo?”. Ricordiamoci, poi, che Gene è un poliziotto, è abituato a pensarci due volte prima di ammazzare uno e, vedendo un sospetto di spalle, è più portato a stordirlo che ad ammazzarlo.

Francamente: chi impiegherebbe cinque anni a progettare un piano così

strampalato, con così tante variabili? Roberto dice che Sonja “ci prova”, e che se fallisce è pronta a riattare un piano così. Ma francamente se il piano fallisce è complicatissimo, se non impossibile, riattuarlo. Insomma, *potrebbe* andare tutto bene, d'accordo, ma mi sembra onestamente molto difficile che nessuna di queste variabili impazzisca mandando all'aria tutto. Per un piano studiato in cinque anni, sono molte di più le cose che possono andare male di quelle che possono andare bene, anche perché il piano è strettamente “psicologico”: presuppone che ben due persone si comportino in un certo modo in varie situazioni.

Se dovessi ideare io il piano, non mi comporterei mai come Sonja, ma farei così: attirerei lì Gene magari facendolo chiamare dalla moglie sotto minaccia (o con un altro stratagemma, ma non certo chiamando la polizia e sperando che affidino la chiamata a lui). Lo guarderei arrivare dalla finestra e, sempre tenendola sotto minaccia, obbligherei Kelly a chiamare Tom per attirarlo lì. A questo punto però... manderei da sola il piano all'aria chiedendomi: “Come diavolo faccio a fare in modo che Gene ammazzi Tom?”... Questo è davvero il nodo più difficile, addirittura irrisolvibile direi: devo sperare che Gene si liberi e, poi, liberatosi, c'è una probabilità su cento che Gene ammazzi Tom...

E non basta che l'assunto sia: Sonja è pazza e ha ideato un piano folle. Perché, comunque sia, il gioco dà per scontato che il piano funzioni, incastrando troppe variabili improbabili. Soprattutto Tom: solo un uomo su dieci si comporterebbe come lui senza insospettirsi, spinto unicamente da un sms. Io, dopo un sms del genere, quanto meno chiamo il mio amico, e se non lo trovo chiamo Kelly. E se non rintraccio nessuno dei due mi metto sul chi vive. E se proprio voglio andare nella casa di uno sconosciuto a incontrare un “amico” che so essere furioso perché ha scoperto il tradimento, mi faccio accompagnare.

SECONDO INTRECCIO: TOM E' IL SERIAL KILLER E SONJA UNA VITTIMA. Ma allora perché il serial killer mi lega e non mi ammazza subito? Cos'ha di tanto urgente da fare? Vuole torturarmi? D'accordo: ma allora sarebbe più logico legare la donna, Sonja, l'altra vittima, e chiudere me, un uomo, per di più un poliziotto, in cantina: il poliziotto, infatti, ha più probabilità di liberarsi se legato a una sedia.

Roberto infine mi ha fatto notare che ci sarebbe anche un terzo intreccio: IL SERIAL KILLER E' SONJA. Ma questa ipotesi non mi è mai minimamente balenata nella mente giocando. Nessun passaggio dell'avventura me l'ha fatto credere. Ma se lei lo fosse sarei assalito da miriadi di domande: è la stessa Sonja del mio passato? Perché ci attira tutti lì dopo cinque anni? perché non ci ammazza separatamente? E se lei non è la Sonja del mio passato, perché attira proprio me, Tom e Kelly? E' forse un serial killer dedito all'eliminazione dei “triangoli” sentimentali?

DIALOGHI. Chiuso il capitolo storia, voglio dire due parole sull'interazione con i personaggi. Ho trovato deboli i dialoghi che, proprio perché ci sono pochissimi personaggi, un giocatore di narrativa interattiva si aspetterebbe più approfonditi: le scelte sono poche, non condizionano la storia, e fanno approdare sempre allo stesso punto. Sembrano, insomma, inseriti tanto per essere inseriti, visto che una paginata di testo con i dialoghi *senza interazione* avrebbe dato lo stesso risultato.

CONCLUSIONE. Little Falls, a prescindere dalle osservazioni sulla trama, resta un gioco divertente, coinvolgente con una bella grafica e un bel sonoro, un ottimo esperimento nel panorama italiano. Ma secondo me gli autori, avendo tra le mani una storia non troppo solida, hanno un po' voluto strafare proponendo intrecci diversi, incastri e finali a go-go, finendo con il creare un po' troppa confusione.

Little Falls

di Roberto Grassi e Alessandro Schillaci

recensita da Francesco Sircana

Una casa da incubo sulle rive del Mississippi

Senso di colpa, angoscia, pericolo imminente: questi gli ingredienti di *Little Falls*, la nuova avventura testuale illustrata realizzata da Alessandro Schillaci e Roberto Grassi, con immagini di Enrico Simonato e musiche originali di Roberto Grassi.

Un agente di polizia ligio al dovere, un incidente inaspettato che cambia la vita, la convinzione di aver perduto per sempre qualcosa (il passato). Uno psicopatico omicida in libertà, una casa da incubo sulle rive del Mississippi, sangue che cola dalle pareti (il presente).



In *Little Falls* l'atmosfera è palpabile; il profilo del protagonista adeguatamente caratterizzato, luoghi e azioni coerenti, trama al cardiopalma, con frequenti colpi di scena: mentre esploravo la tetra abitazione, ho sperimentato tutte le emozioni del thriller: mi sentivo braccato e in pericolo, percepivo l'oscura presenza del killer attorno a me, avevo sentore di orrori dietro l'angolo.

L'accompagnamento sonoro (non solo musica, ma anche effetti) merita

menzione: gradevole e appropriato, contribuisce alla peculiarità del gioco, non disturba ma arricchisce; la grafica gioca un ruolo importante nel contribuire a rendere unica l'atmosfera di *Little Falls* (i toni scuri, l'uso delle tinte seppia, il taglio spettrale di alcune immagini): il mistero che avvolge la villa è notevole.

Nel complesso, un'avventura curata e intrigante, narrata con impegno, innovativa per via dell'uso esperto delle risorse Glulx, soprattutto coinvolgente e ricca di atmosfera e realismo (due ingredienti a mio giudizio fondamentali), sebbene di modeste dimensioni e abbastanza semplice da risolvere.

Insomma, provatela; soprattutto se vi piace il genere: è un pezzo unico nel panorama dell'IF italiana.

Little Falls

di Roberto Grassi e Alessandro Schillaci

recensita da Roberto Bertoni

Le avventure testuali per un fan delle avventure grafiche come me rappresentano un po' il big bang, quello da cui tutto inizia; devo ammettere che, seppur dovendo concettualmente molto a questo tipo di gioco, non sono mai stato attratto dal genere: niente grafica (ma stiamo scherzando? Nel 2005?), il bisogno di scrivere esattamente la frase in italiano (o ancor peggio in inglese), la difficoltà di orientarmi all'interno di stanze o labirinti, sono sempre stati ostacoli insormontabili per me povero videogiocatore.

Questo fino all'arrivo di *Little Falls*: all'inizio devo ammettere che un po' di scetticismo, nonostante le promesse degli sviluppatori, c'era. Ma dopo aver lanciato il gioco mi sono dovuto ricredere e non di poco. Prima di tutto, però, mi sembra doveroso un breve accenno alla storia: in quest'avventura impersoniamo un poliziotto che nel suo giro di ricognizione deve fermarsi davanti a una casa misteriosa. Di più non posso dirvi e questo è il primo punto di forza di questa produzione: la storia è un susseguirsi di colpi di scena, il ritmo è incalzante, non è permesso un attimo di tregua e questo per un'avventura testuale (almeno per la mia esperienza) è qualcosa di sicuramente nuovo.



Le novità però non si fermano qui: a fianco della componente testuale sono presenti due elementi che contribuiscono a creare l'atmosfera: l'audio e la grafica; iniziamo da quello più "strano", la grafica: a fianco dello spazio per scrivere sono presenti infatti, nella parte alta dello schermo, alcuni screenshot che accompagnano quella che è la vostra avventura, aiutando così a focalizzare la realtà che investe (nel vero senso della parola) il giocatore. Lo stile di questi disegni è graffiante, sporco, contribuendo ad aumentare l'atmosfera di tensione che circonda il gioco. E qui entra in gioco il sonoro: grida, cigolii, musiche "devianti", tutto rende l'esperienza del giocatore viva, angosciante, nel terrore che la prossima mossa possa essere l'ultima.

La struttura del gioco beneficia poi di un'ulteriore chicca, con la presenza di finali multipli che invogliano il giocatore a riprendere in mano il gioco anche una volta finito, nel tentativo di trovare la mossa giusta per sopravvivere o anche solo per vedere un altro finale.

Se dovessi trovare una parola per definire *Little Falls*, la più adatta sarebbe senza dubbio "coinvolgente", mentre se volete trovare un difetto, forse è l'eccessiva brevità, anche se da una parte la longevità aumenta proprio in luce dei numerosi finali mentre dall'altra la brevità è frutto della tensione che spinge il giocatore a giocare in maniera forsennata :-).

Se la mia recensione vi può sembrare troppo ottimistica o entusiastica, beh... ne riparleremo dopo che avrete provato il gioco!

Radicofani

di Roberto Marcarini

recensita da **Andrea Rezzonico**

Ancora una volta Amelia, la tua ragazza, non è accanto a te: tu sei lì, nel suo appartamento, che aspetti il suo arrivo e lei invece è a chilometri di distanza, a svolgere con passione il suo lavoro di restauratrice; la sua professione - che ha impedito alla vostra relazione di crescere - ora l'ha portata a Radicofani, un piccolo paesino sito fra le colline senesi. Qui, però, sta accadendo qualcosa: qualcosa di terrificante e di maligno che è in procinto di spezzare la quiete del paesino, e Amelia deve averlo scoperto... E nel momento in cui anche tu intuirai in quale oscura storia sia stata coinvolta, capirai che la sua assenza, forse, è giustificata...

Iniziamo col dire che il punto di forza di *Radicofani*, avventura scritta da Roberto Marcarini, è sicuramente la trama: una trama che non annoia, una trama che viene rinvigorita nel gioco grazie ad azzeccati sipari che interrompono le nostre azioni fornendoci particolari e sviluppi decisamente interessanti. Un altro merito che deve essere riconosciuto all'autore, inoltre, è l'aver sfruttato abilmente in talune occasioni anche la grafica statica (come ad esempio con l'enigma del cameo o con i dischi di Amelia - a proposito, questa ragazza ha dei gusti inattaccabili: Beatles e Pink Floyd!).

A questo punto vorrei passare ad affrontare i lati negativi di quest'avventura, ed è per questo che vi invito a mettervi comodi: ne avremo per molto.

Se vogliamo congedare questa pratica con una sola ma esaustiva frase, possiamo affermare che la programmazione è assolutamente inadeguata: provate a immaginare un difetto di programmazione e con tutta probabilità lo troverete in *Radicofani*.

Volendo sorvolare sulla circostanza che le immagini di Amelia che lentamente si trasforma possono anche infastidire, abbiamo comunque i nostri motivi per lamentarci...

Innanzitutto il parser è limitatissimo: quest'avventura ci costringerà ad una fastidiosissima "caccia alla parola", e questo secondo me è uno dei peccati capitali degli scrittori di avventure (se poi consideriamo che anche esaminare le persone che abbiamo di fronte a volte è impossibile, allora possiamo davvero aspettarci di tutto); oltre a ciò, in alcune locazioni mancano o sono totalmente inadeguate le descrizioni, causandoci anche qualche piccolo problema di orientamento: pensiamo alla casa di Amelia, dove a causa della mancanza di una qualsiasi indicazione (es. "a nord vedi la cucina") nei primi minuti di gioco ho rischiato di perdermi, sentendomi come un simpatico personaggio di un vecchio manga giapponese...

Inoltre, anche l'interfaccia non è molto soddisfacente e su questo punto possiamo limitarci a segnalare che se il nord è davanti e il sud è alle nostre spalle, allora l'est deve essere alla nostra destra e l'ovest a sinistra e non il contrario (guardate i pulsanti delle direzioni cardinali). Per concludere posso affermare che la sensazione che si ha giocando a *Radicofani* è che l'autore,

dopo essersi dedicato con profitto alla trama e agli enigmi (che se non brillano per originalità almeno sono ben inseriti nel contesto nel quale si presentano), abbia deliberatamente evitato ogni minimo sforzo nella programmazione. Una volta posta in essere questa considerazione, di conseguenza non posso che augurarmi che Roberto Marcarini si rimetta nuovamente al lavoro e — prestando l'attenzione dovuta alla programmazione — rilasci una seconda versione di quest'avventura: sarebbe certamente un peccato se non lo facesse.

PRO: buona trama, intelligente utilizzo della grafica statica

CONTRO: parser limitatissimo, descrizioni più che inadeguate, interfaccia insoddisfacente.

Gumshoe

di Roberto Grassi

Porte da aprire e chiudere

In un recente thread di discussione si è parlato molto di porte da aprire e chiudere, come mezzo (enigma?) per precludere fisicamente alcune locazioni al giocatore.

In Gumshoe, un gioco del 1996, ho trovato proprio un bell'esempio di porta per discutere di questo argomento. Colgo l'occasione per scusarmi con i lettori del fatto che non ho proseguito con l'analisi del gioco *Fear* iniziata nello scorso numero, ma ho ritenuto più interessante affrontare questo argomento. Tornerò sul gioco *Fear* nel prossimo numero.

Gumshoe inizia con il nostro protagonista che si risveglia, dopo una notte alcolica, sul pianerottolo dell'edificio davanti alla porta del suo ufficio d'investigazione. Dopo le righe introduttive, ci troviamo di fronte alla seguente descrizione:

Hallway

You are in the second floor hallway of the McMurtry building. To the east, a stairway leads down to the first floor. Your office is to the south. The friendly-looking welcome mat that you purchased sits expectantly in front of the door.

The door leading into the reception area is closed.

You can also see a copy of the Los Angeles Times here.

Come vedete, c'è una porta che conduce allo studio. Fate attenzione a quest'ultima e cerchiamo di capire se la sua implementazione è soddisfacente.

>x door

You swell with pride as you read on the door:

Joe Gumshoe

Private Investigator

Cerchiamo di aprire la porta.

>open door

The doorknob frustratingly evades your grasp.

Non riusciamo perché siamo alticci e non riusciamo a prendere la maniglia. Dopo aver aspettato un po' di tempo, arriva la nostra zia Marge ad aprire lo studio e ci vede ubriachi sul pianerottolo.

>ask marge about door

“It’s locked, but it’s your office. You should have a key.” She looks at you as if you were a child. You’ve always hated that.

La zia Marge ci dice che la porta è chiusa ma che dovremmo avere le chiavi. (Naturalmente, non le abbiamo).

Riproviamoci.

>open door

It seems to be locked.

You feel almost human again.

Aunt Marge unlocks the door and opens it.

Ok, la zia apre la porta dell’ufficio ed entra.

La seguiamo.

Guardiamo ora la descrizione della camera

Reception area

This is the reception area outside of your office. A door to the north leads out into the hallway. In one corner of the room, next to the opening that leads east into your office, is the small desk that Marge uses. A coat rack stands beside the door.

The door leading out to the hallway is open.

Hanging on the coat rack is your hat.

Aunt Marge walks into view.

Aunt Marge sits down at her desk and begins working (on what, you have absolutely no idea).

La porta è sempre nella descrizione e probabilmente c’è un motivo particolare per cui debba esserci. Tuttavia, esaurita la parentesi iniziale, non c’è alcun motivo per cui questa porta debba esistere in questo momento. In altre parole, non dovrebbe più essere gestita come ‘ostacolo’ e bisognerebbe assecondare i movimenti del giocatore.

Invece...

>close door
You close the office door.

“I’ll be leaving around 2:00 today,” Marge says. “I’ve got some shopping to do.”

>out
You can’t, since the office door is in the way.

>open door
You open the office door.

E questo mi consente di andare fuori. Poi rientro nella stanza e succede qualcosa di strano...

>z
Time passes.

Marge shakes her head disapprovingly and goes to shut the door.

>marge, open the door

Marge returns to her desk, muttering something about nephews that would lose their heads if they weren’t attached.

>marge, open the door

Marge gets up and opens the door.

“Lost your keys again?” she scolds.

La zia Marge va spontaneamente a chiudere la porta, se questa rimane aperta. In questo modo, se volessi uscire, dovrei continuare a digitare “open door” oppure chiedere alla Zia Marge di aprirla. Da notare che l’autore non sente alcuna esigenza di mettere un’altra porta nel gioco, quella che separa la reception dal nostro ufficio vero e proprio. Infatti...

>in

Office
Your office is amazingly messy considering you only just moved in a month ago. At one end of the room is a cluttered desk and, behind

it, a large, comfy looking chair. On the other side of the desk is a small chair, presumably meant for clients (although you haven't had any yet). The reception area lies to the west, and a window looks out onto the street to the east.

>out

Reception area

This is the reception area outside of your office. A door to the north leads out into the hallway. In one corner of the room, next to the opening that leads east into your office, is the small desk that Marge uses. A coat rack stands beside the door.

Come vedete, in questo caso si suppone che una porta ci sia ma non è necessario gestirla.

Dunque? Quali conclusioni ne possiamo trarre?

Implementare in maniera soddisfacente una "porta" è difficile. Bisogna sempre trovare un equilibrio fra il tipo di gioco che stiamo scrivendo e la giocabilità che intendiamo dargli.

Nel caso specifico, la porta in questione serviva come 'ostacolo' fisico per il nostro accesso alle locazioni, ma dopo aver assolto questo ruolo, diventa solo un ostacolo senza scopo (anzi, la zia Marge continua a chiuderla in continuazione).

Giusto?

Sbagliato!

L'ostinazione con cui la zia Marge continua a chiudere la porta è sospetta. C'è infatti un enigma collegato a questo evento e non vi dirò di più per non rovinarvi il gusto del gioco.

Quest'ultima considerazione fa quindi pensare al fatto che l'autore di questo gioco abbia pensato molto all'effettiva implementazione della "porta" nel gioco, tenendo dunque presente le considerazioni che vi ho esposto in precedenza, e che l'abbia adattata alle sue esigenze di game design.

Questo mi sembra un indice di accuratezza e credo che si debba considerare come un buon esempio.

INFORM: La libreria `sgw.h` per Glulx

di Alessandro Schillaci

Per la comprensione di questo articolo sono richieste conoscenze minime di Inform. Non sono richieste, invece, conoscenze di Glulx.

INTRODUZIONE

La libreria **Simple Glulx Wrapper** (SGW) è una piccola e semplice libreria che estende le principali funzioni glulx per la gestione di elementi multimediali (immagini e suoni).

È sufficiente includere la libreria ed è possibile inserire immagini, suoni o musica, sfruttando le funzioni già predisposte per visualizzare un'immagine oppure riprodurre un suono (o una musica).

In questo modo è anche possibile prendere un'avventura scritta in Z-Code e convertirla in formato Glulx (con immagini e suoni) semplicemente importando la libreria nel file INF principale e utilizzando le funzioni disponibili, senza curarsi di aspetti tecnici quali ad esempio la sincronizzazione degli elementi multimediali nel caso di un `RESTORE` della partita in corso o di un `RESTART`.

Questa libreria deriva direttamente dal codice utilizzato per *Little Falls*, riassembleto da poter essere utilizzato a tutti gli effetti come una libreria.

Questa libreria definisce e prevede anche 3 canali audio da utilizzare nelle vostre avventure:

- **music** : è il canale dedicato alla musica
- **chan1** : è il canale 1 per i suoni (effetti)
- **chan2** : è il canale 2 per i suoni (effetti)

Con questa libreria è possibile anche creare un'avventura testuale con suoni (senza immagini).

PRIMI PASSI

L'utilizzo della SGW è semplicissimo. Basta seguire questi punti:

1. Aggiungere nel vostro file INF principale, questa direttiva di inclusione
Include "sgw.h";
Prima di includere il parser (Include "Parser";)
2. Aggiungere la funzione
initializeSGW(x);
nella funzione `Initialize()` del vostro gioco. Questa funzione inzializza gli oggetti della libreria e crea una finestra grafica per le

vostre immagini con altezza uguale a “x” pixels. Ovviamente è possibile utilizzare un altro valore per cambiare le dimensioni delle immagini.

3. Creare un file di risorse con l’elenco delle immagini e dei suoni che intendete utilizzare nella vostra avventura.
4. Ora è possibile riprodurre un suono o visualizzare un’immagine in qualsiasi punto del codice Inform semplicemente utilizzando le funzioni:
 - ***viewImageLeft(image)*** : visualizza l’immagine allineata a sinistra
 - ***viewImageCenter(image, image_width)*** : visualizza l’immagine centrata nello schermo
 - ***viewImageRight(image, image_width)*** : visualizza l’immagine allineata a destra

(dove *image* è l’identificativo indicato nel file delle risorse)

Da notare che per centrare o per allineare a destra un’immagine, è necessario passare un ulteriore parametro alla funzione (*image_width*). Questo parametro è la larghezza (in pixel) dell’immagine da centrare (allineare a destra).

- ***playSound(channel,sound,lenght,volume)*** : dove *channel*=music oppure *chan1* oppure *chan2*, *sound*=identificativo del suono nel file delle risorse, *lenght* è il numero di volte che il suono verrà riprodotto [-1 = all’infinito] e *volume*=VOLUME_HIGH oppure VOLUME_NORMAL oppure VOLUME_LOW)

5. Questa libreria definisce i colori dei vari tipi di font (quelli usati per *Little Falls*). Ma è possibile cambiarli nel caso non siano di gradimento. Basta definire queste Costanti prima di includere la libreria *sgw* (cioè prima della direttiva `Include "sgw.h"; :`

Esempio:

```
Constant SCBACK $110101;
Constant SCTEXT $DDDB99;
Constant SCSoft $665544;
Constant SCEMPH $FFFFDD;
Constant SCHEAD $EEDDAA;
Constant SCINPU $DDEEAA;
```

6. Se volete utilizzare la libreria solo con il supporto audio, è possibile “inibire” l’inizializzazione grafica impostando questa costante prima di includere la libreria:

```
Constant NOGRAPHICS;
```

Ecco nel dettaglio tutte le funzioni che la SGW rende disponibili

Funzioni grafiche

- ***clearMainWindow()***: ripulisce la finestra principale
- ***closeAllWindows()***: chiude tutte le finestre grafiche
- ***initializeSGW(h)***: imposta l'altezza della finestra grafica a h
- ***viewImageLeft(image)***: visualizza l'immagine allineata a sinistra
- ***viewImageCenter(image, image_width)***: visualizza l'immagine centrata nello schermo
- ***viewImageRight(image, image_width)***: visualizza l'immagine allineata a destra

Funzioni audio

- ***playSound(channel,sound,lenght,volume)***: riproduce un suono o musica
- ***silenceAll()***: azzera tutti i canali audio
- ***silenceChannel(channel)***: azzera il canale audio "channel"
- ***setVolume(val, channel)***: imposta il valore "val" per il volume

ESEMPIO

Ricreiamo un esempio completo (da zero) e vedrete che è più semplice a farsi che a dirsi. L'esempio in questione è compreso nella distribuzione della sgw, ma in queste pagine vedremo come costruirlo da zero.

Si presume che il lettore abbia installato sul suo sistema un compilatore INFORM e i tool Blorb (blb.exe e bres.exe).

Il file principale

Creiamo una nuova directory C:\test\ (per esempio). Creiamo un nuovo file con questo nome: sgw_test_it.inf (e salviamolo nella directory di test).

Ecco il contenuto del file:

```
Constant Story "Test per sgw";
Constant Headline "^Simple Glulxe Wrapper (sgw) 1.5 ^di
                  Alessandro Schillaci^^";
Include "sgw.h";      ! Inclusionione della libreria SGW
Include "Parser";
Include "Infglk";

Object LibraryMessages
  with before [;
    LMODE1 : lookmode =2; print "^"; rtrue;
    LMODE2 : lookmode =2; print "^"; rtrue;
    LMODE3 : lookmode =2; print "^"; rtrue;
  ];

Include "VerbLib";
Include "Replace";
Include "sgw_test_it.bli";

Object room_1 "room 1"
  with description [;
    print "Una semplice stanza di esempio. Stanza 1. L'immagine è
          allineata a sinistra.^";
```

```

        viewImageLeft(room1);
    ],
    cant_go "Puoi muoverti solo a nord, verso la stanza 2.",
    n_to room_2,
    has light;

Object room_2 "room 2"
    with description [;
        print "Una semplice stanza di esempio. Stanza 2. L'immagine è
            centrata.^";
        viewImageCenter(room2,620);
    ],
    cant_go "Puoi muoverti a nord verso la stanza 3 oppure a sud,
verso la
            stanza 1.",
    n_to room_3,
    s_to room_1,
    has light;

Object room_3 "room 3"
    with description [;
        print "Una semplice stanza di esempio. Stanza 3. L'immagine è
            allineata a destra.^ In questa stanza puoi vedere un
            esempio di utilizzo degli stili della SGW.^";
        print "Testo con lo stile ", (s_emph) "Emphasized", "^";
        print "Testo con lo stile ", (s_bold) "Bold", " (come in
Inform)^";
        print "Testo con lo stile ", (s_pref) "Preformatted", "^";
        print "Testo con lo stile ", (s_fixed) "Fixed", " (come in
Inform)^";
        print "Testo con lo stile ", (s_head) "Header", "^";
        print "Testo con lo stile ", (s_subhead) "Subheader", "^";
        print "Testo con lo stile ", (s_alert) "Alert", "^";
        print "Testo con lo stile ", (s_reverse) "Reverse", " (come
in
            Inform)^";
        print "Testo con lo stile ", (s_note) "Note", "^";
        print "Testo con lo stile ", (s_underline)
"Underline/Italic", "
            (come in Inform)^";
        print "Testo con lo stile ", (s_block) "BlockQuote", "^";
        print "Testo con lo stile ", (s_input) "Input", "^";
        viewImageRight(room3, 620);
    ],
    cant_go "Puoi muoverti solo a sud, verso la stanza 2.",
    s_to room_2,
    has light;

[ Initialise;
    lookmode = 2;
    initializeSGW(240); ! Imposta la finestra grafica alta 240 pixel
    location = room_1;
    playSound(chan1,rumble,1,VOLUME_HIGH);
    playSound(chan2,heart,-1,VOLUME_NORMAL);
];

Include "ItalianG";

```

Come possiamo notare, è stata inclusa la libreria `sgw` (`Include "sgw.h"`) prima di includere `Parser`. Nella funzione `Initialise()` ho inserito

l'inizializzazione della libreria `sgw` (`initializeSGW(240)`) e impostato il lookmode a 2 (vedremo in seguito il motivo di questa operazione). In questo modo ho incluso e inizializzato la libreria SGW, possiamo riprodurre un suono e visualizzare un'immagine in qualsiasi punto del codice Inform.

Il file delle risorse

Ovviamente serve un altro file, che gestisce la “mappatura” delle risorse (i file multimediali JPF o AIFF). Creiamo dunque un nuovo file (`sgw_test_it.res`) sempre nella directory di “test”, con questo contenuto (questo file viene utilizzato per la creazione del file `sgw_test_it.bli` grazie all'utility `bres.exe` — in seguito vedremo come utilizzarlo a linea di comando):

```
CODE C:\test\sgw_test_it.ulx

PICTURE room1 C:\test\room1.jpg
PICTURE room2 C:\test\room2.jpg
PICTURE room3 C:\test\room3.jpg

SOUND rumble C:\test\rumble.aif
SOUND heart C:\test\heart.aif
```

In questo caso dovremo avere a disposizione tre immagini (di dimensioni 620×240 pixel) e due suoni in formato AIF (`rumble.aif` e `heart.aif`). Se non avete a disposizione tali file, non preoccupatevi. Potete scaricare il file di esempio della libreria SGW: include già un esempio da compilare con le risorse già pronte. Il link di partenza è sempre lo stesso:

http://www.slade.altervista.org/?Inform:Simple_Glulx_Wrapper

In questo caso, supponiamo di avere nella directory `C:\test\` i file JPG e AIF appena descritti.

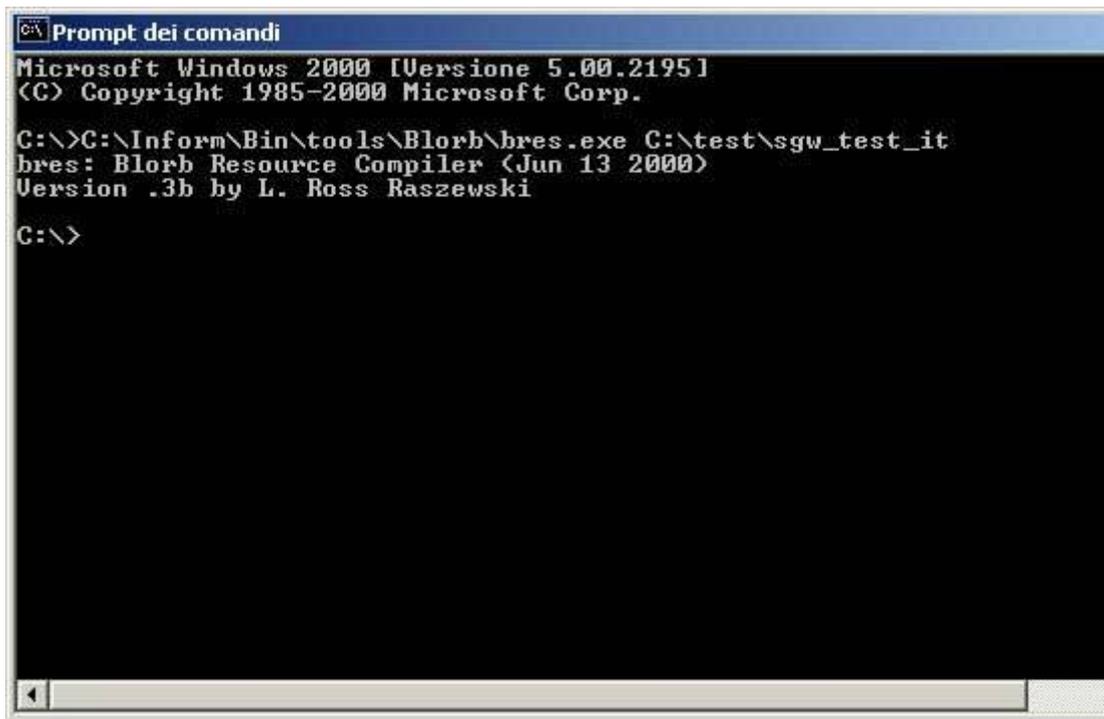
Creazione del file bli e blc

Il primo passo consiste nell'esecuzione del tool `bres.exe` utilizzando come parametro il file di risorse `sgw_test_it.res` creato in precedenza.

Supponendo che il percorso assoluto del tool `bres.exe` sia `C:\Inform\Bin\tools\Blorb\`, possiamo lanciare il seguente comando (in una finestra DOS):

```
C:\Inform\Bin\tools\Blorb\bres.exe C:\test\sgw_test_it
```

Se tutto è andato correttamente, dovrete vedere un messaggio simile a questo:



```
C:\> Prompt dei comandi
Microsoft Windows 2000 [Versione 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>C:\Inform\Bin\tools\Blorb\bres.exe C:\test\sgw_test_it
bres: Blorb Resource Compiler (Jun 13 2000)
Version .3b by L. Ross Raszewski

C:\>
```

Vedrete che il tool *bres.exe* ha creato due nuovi file: uno con estensione BLI (*sgw_test.bli*) e il secondo con estensione BLC (*sgw_test.blc*). Provando ad aprire questi file generati vedrete che si tratta di file che associano una risorsa (immagine o suono) ad un codice univoco. Questo servirà poi per assemblare il tutto in un file con estensione BLB (eseguibile direttamente da un interprete Glulx).

Compilazione

Questo passo consiste nella compilazione del file principale (quello che abbiamo chiamato *sgw_test_it.inf*). Supponendo che:

- il compilatore *inform* abbia come percorso
C:\Inform\Bin\compiler\
- le librerie *Inform* (e *INFIT*) abbiano come percorso
C:\Inform\Lib\Base
- che la libreria *sgw.h* abbia come percorso C:\Inform\Lib\Contrib

il comando sarà:

```
C:\Inform\Bin\compiler\inform.exe +language_name=italian -G
+include_path=C:\test,C:\Inform\Lib\Base,C:\Inform\Lib\Contrib
C:\test\sgw_test_it.inf C:\test\sgw_test_it.ulx
```

Il messaggio di compilazione dovrebbe essere simile a questo:

```
Microsoft Windows [Versione 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>C:\Inform\Bin\compiler\inform.exe +language_name=italian -G +inc
Base.C:\Inform\Lib\Contrib C:\test\sgw_test_it.inf C:\test\sgw_test_
Inform 6.30 for Win32 (27th Feb 2004)
Using the SGW 1.5 (Simple Glulxe Wrapper) library by Alessandro Schi
Using Infit v2.5 / Serial: 841014 / (c) 2004 by Giovanni Riccardi

[Including <infglk>]
Including bres resource declarations generated on Jun 13 2000
C:\Inform\Lib\Contrib\sgw.h(135): Warning: Defined constant "VOLUME
C:\Inform\Lib\Contrib\sgw.h(152): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(155): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(158): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(161): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(164): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(167): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(170): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(173): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(176): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(179): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(182): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(185): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(188): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(191): Warning: Defined constant "CLR_GG
C:\Inform\Lib\Contrib\sgw.h(397): Warning: Routine "clearMainWindow
C:\Inform\Lib\Contrib\sgw.h(478): Warning: Routine "silenceChannel"
Compiled with 17 warnings

C:\>
```

Andando a controllare la directory `C:\test`, vedrete che è stato creato un nuovo file `sgw_test_it.ulx`.

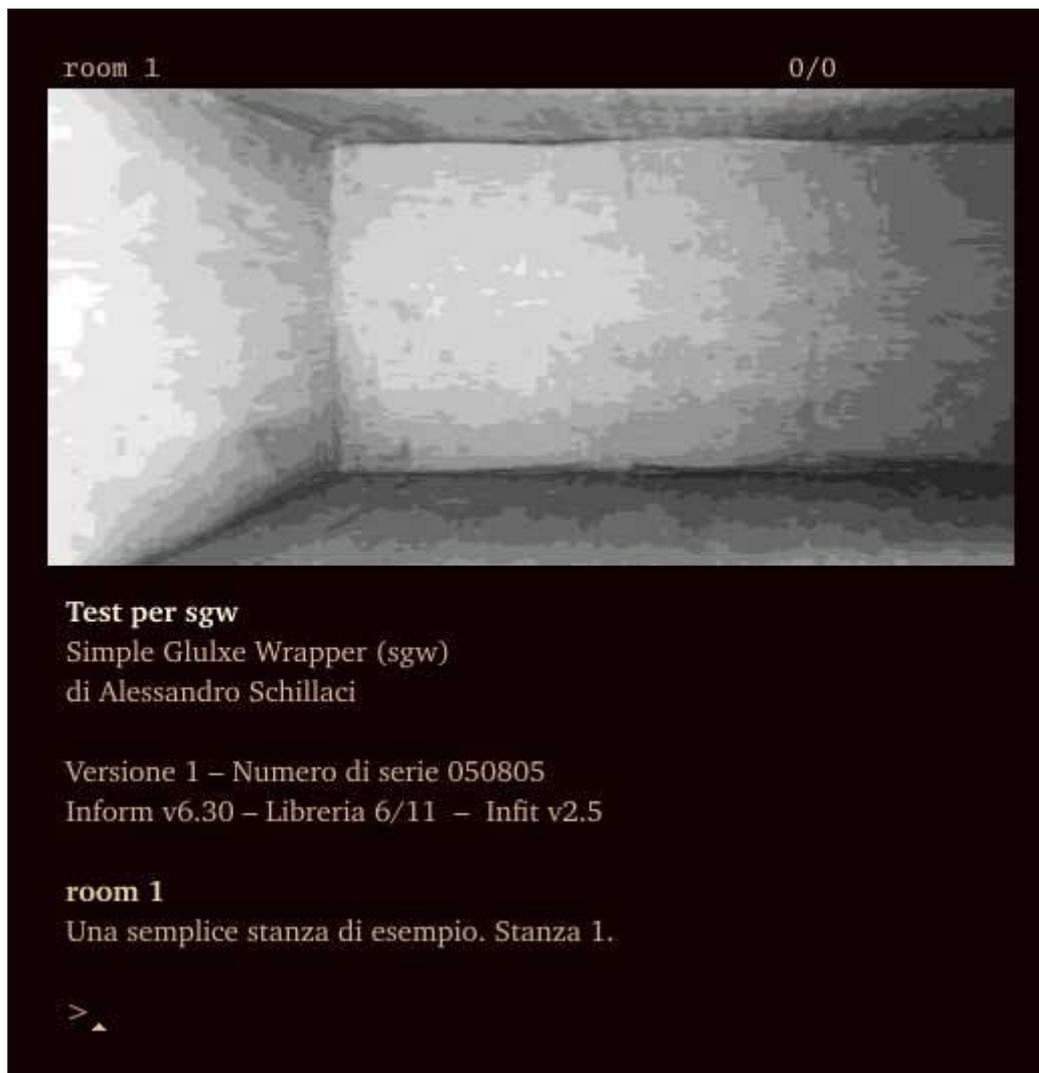
Creazione del file BLB

L'obiettivo è quello di creare un file BLB, che contiene tutte le risorse (grafiche e sonore) e l'avventura testuale stessa.

Per la creazione di tale file, si utilizza il tool `blc.exe`. Supponendo che tale file risieda nella directory `C:\Inform\Bin\tools\Blorb\`, il comando allora sarà:

```
C:\Inform\Bin\tools\Blorb\blc.exe C:\test\sgw_test_it.blc
C:\test\sgw_test_it.blb
```

Se non vi sono errori, avete appena creato il vostro file `sgw_test_it.blb` che utilizza la libreria SGW.H



Compilare l'esempio con JIF

È possibile scaricare la libreria SGW con un esempio (che è molto simile a quello appena effettuato), solo che tutti i file sono pronti per la compilazione.

Infatti, l'esempio contiene anche il file blb già compilato per vedere come funziona senza dover ricompilarlo. Ad ogni modo, è ovviamente possibile ricompilare il codice usando JIF. (È possibile ricompilarlo anche con altri tool, ma in questa pagina vedremo come compilarlo con JIF)

Sistemi win32

1. Scaricare e scompattare l'esempio nella directory `C:\sgw_it`
2. Aprire JIF (utilizzare se possibile la versione 2.0)
3. Aprire un progetto: Aprire il file `C:\sgw_it\sgw_test_it.jpj`
4. Aprire il file `sgw_test_it.inf` (sceglierlo dalla finestra di progetto, in basso a sinistra) con un doppio click sul nome del file
5. Compilare il progetto
6. Eseguire il progetto (l'icona del "razzo")
7. È possibile anche ricreare il file di risorse e il file blb da zero. In questo

caso cancellare i file *.bli *.blc *.blb ed eseguire i menu:

- Glulx -> Crea File di risorse
- Glulx -> Compila il file inf
- Glulx -> Crea Blb

***nix systems**

1. Scaricare e scompattare l'esempio in una directory XXX
2. Aprire il file XXX\sgw_test_it.jpj con un editor di testo e modificare i path dei file, cambiandoli in e facendoli puntare alla directory XXX
3. Chiudere il file XXX\sgw_test_it.jpj
4. Aprire il file XXX\sgw_test_it.res con un editor di testo e modificare mettendo i path giusti (riferiti a XXX)
5. Chiudere il file XXX\sgw_test_it.res
6. Aprire JIF (utilizzare se possibile la versione 2.0)
7. Aprire un progetto: Aprire il file XXX\sgw_test_it.jpj
8. Aprire il file sgw_test_it.inf (sceglierlo dalla finestra di progetto, in basso a sinistra) con un doppio click sul nome del file
9. Compilare il progetto
10. Eseguire il progetto (l'icona del "razzo")
11. È possibile anche ricreare il file di risorse e il file blb da zero. In questo caso cancellare i file *.bli *.blc *.blb ed eseguire i menu:
 - Glulx -> Crea File di risorse
 - Glulx -> Compila il file inf
 - Glulx -> Crea Blb