

Programmiamo avventure testuali!

Iniziamo con lo spiegare cos'è un'avventura testuale.

Un'avventura testuale è un tipo di videogioco che si basa principalmente su enigmi (con alcune eccezioni) da risolvere nato negli anni '70 e basato su un'interfaccia, come dice il nome, puramente testuale, ossia il giocatore interagisce leggendo testo e digitando parole, che d'ora in poi chiameremo *comandi*. Potrebbe sembrare un'esperienza di tipo arcaico e ormai obsoleta, visti i recenti sviluppi in ambito grafico (anche se, come ben ci ricorda Sheldon Cooper in *The Big Bang Theory*, le avventure testuali utilizzano il più potente chip grafico tra tutti: l'immaginazione), ma basti pensare che sono pochi i videogiocatori che non conoscono, almeno di nome, i celebri titoli che ne hanno fatto la storia, uno fra tutti *Zork*, segno che questo tipo di giochi è stato d'impatto una volta rilasciati sul mercato, al punto che anche oggi, a distanza di quarant'anni, esiste una comunità, ristretta ma mai estinta e quantomai tenace, dedicata allo sviluppo di avventure testuali, molte delle quali elogiate per la loro pregevolezza anche in un ambito videoludico più generale; ad esempio è recente la notizia della pubblicazione di un'avventura testuale finanziata con metodo di crowdfunding e rilasciata nientemeno che su Steam, annualmente si svolgono tutt'ora competizioni internazionali per stabilire quale sia la migliore avventura testuale rilasciata e, anche qui in Italia, esistono persone che si danno ancora da fare per diffondere questo genere di videogiochi.

Io mi chiamo Mariano Sassi, il mio pseudonimo in campo cibernetico è Di avolo dei crocicchi, sono da qualche anno appassionato e fervente sostenitore di avventure testuali, di cui ho avuto esperienza dapprima come giocatore e, passando attraverso una (breve) fase di beta tester, sono infine approdato alla programmazione e alla recensione di queste ultime.

Per programmare utilizzo il linguaggio Inform 6, ormai "superato" dalla versione 7 di cui Leonardo Boselli, responsabile degli adattamenti in italiano, ha già caricato numerosi tutorial su youtube, tuttavia reputo che anche la versione "meno evoluta" abbia numerose possibilità di utilizzo e, forse, di comprensione, risultando molto vicina, ad esempio, al C++ che la maggior parte dei programmatori senz'altro conosce.

Scopo di questi miei interventi, comunque, è fornire una panoramica più o meno approfondita di Inform 6 spiegandone i meccanismi partendo da zero affinché vi si possa avvicinare chiunque. Per fare ciò mi avvarrò dell'*Inform Designer's Manual* scritto da Graham Nelson, lo sviluppatore di questo linguaggio di programmazione, e del *Manuale Inform* di Vincenzo Scarpa (entrambi ottimi e comunque, a mio parere, imprescindibili), cercando tuttavia di essere il più possibile originale negli esempi, con l'idea in futuro, magari, di raccogliere questi articoli in un nuovo manuale di cui altrove si è fatta "richiesta". Probabilmente non risulterà sempre esauriente per non appesantire gli articoli, ma rimango comunque a disposizione per chi volesse approfondire.

Ogni intervento vedrà la presentazione di un codice sorgente che, volta per volta, analizzerà un aspetto caratteristico di Inform 6 partendo dalle basi fino ad arrivare a codici sorgenti più complicati per permettere ai futuri programmatori di esprimere al meglio la loro creatività.

La fine dell'inizio è quindi per spiegare dove reperire il materiale necessario, per poi partire col prossimo articolo direttamente alla spiegazione di alcuni codici sorgente.

Digitate nella barra url di un browser <http://www.vincenzoscarpa.it/> poi cliccate a sinistra il link **Inform e Glulx**: troverete un elenco di file scaricabili, ma a noi per il momento interessano solo

LIBRERIE DI SISTEMA (v6.11)

LIBRERIE IN ITALIANO (INFIT v2.5)

COMPILATORE INFORM (v6.30)

INTERPRETE INFORM (WINDOWS FROTZ 2002 v1.08)

WIDE, quest'ultimo si trova in penultima posizione nella lista, per chi non riuscisse a scaricarlo basta digitare nella barra url <http://vincenzoscarpa.it/inform/manuale/wide099beta.zip>

Una volta scaricati tutti questi file, bisogna recarsi nella cartella C: del proprio computer per creare in essa una cartella denominata *inform*, in quest'ultima se ne dovranno creare altre tre:

- una chiamata *libraries* all'interno della quale vanno scompattati i file che si trovano in *inform_library611.zip* e *infit25.zip* precedentemente scaricati; a questo punto si troveranno in questa cartella le librerie, nell'ordine: *English*, *Grammar*, *infix*, *linklpa*, *linklv*, *Parser*, *parsepm*, *VerbLib*, *verblibm* che vanno rinominate aggiungendo *.h* al termine di ciascuno dei loro nomi (*English.h*, *Grammar.h* ecc...)
- una chiamata *interpreter* all'interno della quale vanno scompattati i file che si trovano in *WindowsFrotz2002.zip* precedentemente scaricato
- una chiamata *wide* all'interno della quale vanno scompattati i file che si trovano in *wide099beta.zip* precedentemente scaricato

Per scompattare i file ci si può avvalere di WinZip o WinRar.

Infine basta tornare nella cartella *inform* in cui vanno scompattati i file che si trovano in *inform_compiler630.zip* precedentemente scaricato.

A questo punto per programmare l'avventura testuale basta lanciare l'ambiente di sviluppo integrato WIDE che ci permetterà di scrivere, compilare ed eseguire il nostro codice sorgente. Verrà dato per noto come utilizzare WIDE, essendo estremamente intuitivo, così come verrà dato per scontato che si utilizzi questo ambiente di lavoro, o uno analogo (Scarpa, ad esempio, spiega nel Manuale come utilizzare JIF).

Per questa volta è tutto: per chi vuole iniziare a curiosare, tra i vari file che sono stati scaricati è presente il suddetto Manuale scritto da Scarpa, per chi è interessato a continuare l'esplorazione di questo linguaggio di programmazione e delle avventure testuali, ci leggiamo al prossimo articolo!