

Note esplicative del progetto

Formato esplicito del file

Il file **mini_monkey_island.txt**, fornito come esempio di input, ha semplicemente lo scopo di creare un mondo virtuale tra i tanti possibili. Si nota che questo non contiene tutti gli oggetti che un mondo virtuale potenzialmente può contenere. A fronte di una corretta progettazione, la comparsa di nuovi oggetti, simili ai precedenti per caratteristiche, non deve provocare alterazioni. Ad esempio, una finestra ha caratteristiche simili a una porta, per cui il caricamento di una finestra da file dovrebbe essere trattato allo stesso modo. Se si svolge il progetto da soli, non essendo richiesta l'implementazione di alcuni oggetti, è sufficiente saltare le linee che si riferiscono a oggetti non implementati.

Per facilitare il parsing del file si è preferito dividere il file in **sezioni**, ciascuna caratterizzata da una stringa in maiuscolo che ne denota la classe di riferimento (LOCAZIONI, APERTURE, PERSONAGGI, SENTIMENTI, OGGETTI, INVENTARIO_PERSONAGGI, LOCAZIONE_OGGETTI, PROPRIETARI).

Ciascuna sezione è composta da un numero (variabile) di righe; ciascuna riga rappresenta un oggetto della sezione corrispondente (ad esempio dentro la sezione PERSONAGGI sono elencati i personaggi da inserire all'interno del mondo virtuale; all'interno della sezione APERTURE vi è la lista di oggetti apertura che fungono da collegamento tra locazioni, ecc.)

Una riga del file può:

- essere vuota
- essere il nome di una sezione (tra l'insieme finito elencato sopra)
- contenere un numero variabile di campi, delimitati dal carattere di tabulazione '\t'

Ogni riga termina con il carattere '\n'.

Commenti: una riga può contenere, come ultimo campo, un commento che inizia con il simbolo '#'. Tutto ciò che segue il simbolo può essere scartato.

Il formato di ciascuna riga, in generale, dipende da sezione a sezione; una caratteristica comune a più sezioni è che, nel momento in cui si introducono nuove entità (siano essi oggetti, personaggi o locazioni), ad esse viene associato un *identificativo stringa* (ID) che permette di riferirsi all'oggetto più avanti nel file. Ad esempio se nella sezione PERSONAGGI una riga presenta come primo campo "PERS_1" e come secondo campo "Guybrush", questo permetterà di riferirsi a "Guybrush" in modo univoco tramite l'ID "PERS_1".

Per tutti i personaggi, gli oggetti e le locazioni è presente un ulteriore campo stringa che fornisce una descrizione verbale dell'oggetto, ovvero un nome che è possibile mostrare al giocatore. Tale stringa non identifica univocamente l'entità, tuttavia può essere utilizzato per scopi di visualizzazione o nel parser testuale o anche nell'azione esamina.

Di seguito il dettaglio del formato per ciascuna sezione:

- LOCAZIONI
ID_LOCAZIONE \t Nome

- APERTURE

ID_APERTURA \t NomeClasse \t Nome \t ID_LOCAZIONE1 \t ID_LOCAZIONE2
Un'apertura collega due locazioni, ID_LOCAZIONE1 e ID_LOCAZIONE2.

- PERSONAGGI

ID_PERSONAGGIO \t Nome \t ID_LOCAZIONE

L'ID_LOCAZIONE specifica la posizione iniziale del personaggio.

Nota bene: Il primo personaggio specificato (ad esempio, PERS_1) può essere considerato il personaggio giocante.

- SENTIMENTI

ID_PERSONAGGIO1 \t NomeSentimento \t ID_PERSONAGGIO2

Ogni riga di sentimento specifica il sentimento del personaggio ID_PERSONAGGIO1 nei confronti del personaggio ID_PERSONAGGIO2. **Nota bene:** la relazione non è simmetrica.

- OGGETTI: questa sezione contiene righe dal formato differente, in funzione del tipo di oggetti coinvolti:

ID_OGGETTO \t NomeClasse \t Nome

Questo formato si applica ai seguenti oggetti: Sedia, lampada, lampadario, letto, grog, coltello bastone, divano, televisore, gettone telefonico, slot machine, denaro, quadro, spago, computer.

ID_OGGETTO \t NomeClasse \t Nome \t Testo

Questo formato si applica agli oggetti che possono essere "letti": libro, lettera, biglietto, dischetto per computer, mappa.

ID_OGGETTO \t NomeClasse \t Nome \t ID_OGGETTO1 \t ID_OGGETTO2 \t ... \t ID_OGGETTON

Questo formato si applica agli oggetti che possono contenere altri oggetti: armadio, libreria, cassetto, scrivania, cestino, vaso, baule, cofanetto, cassaforte, boccale, matrioska, tappeto, cabina telefonica. **Nota bene:** la lista di oggetti può essere anche vuota (cioè, inizialmente l'oggetto non contiene nulla).

ID_OGGETTO \t NomeClasse \t Nome \t ID_APERTURA

Questo formato si applica a oggetti che provocano l'apertura o la chiusura di un'apertura (ovvero di porte, finestre, passaggi segreti o botole): bottone, chiave, leva.

ID_OGGETTO \t NomeClasse \t Nome \t ID_LOCAZIONE

Questo formato si applica a oggetti che permettono il "trasporto" a una certa locazione: pillola e telefono (stile Matrix :-)).

- INVENTARIO_PERSONAGGI

ID_OGGETTO ID_PERSONAGGIO

Ogni riga specifica l'appartenenza di un oggetto a un personaggio.

- LOCAZIONE_OGGETTI

ID_OGGETTO ID_LOCAZIONE

Ogni riga specifica la locazione in cui si trova l'oggetto specificato.

- PROPRIETARI
ID_OGGETTO ID_PERSONAGGIO

Ogni riga specifica il proprietario di un oggetto.

Note durante la creazione degli oggetti:

- Una porta, se non specificato diversamente, è sempre chiusa. L'associazione tra la porta e la chiave viene specificata nella sezione OGGETTI, in cui ciascuna chiave viene associata alla porta che questa apre. Se alla porta è associata una chiave mediante la specifica nella sezione OGGETTI (per esempio, vedi la Chiave OGG_15 nel file di esempio), la porta corrispondente è **chiusa A CHIAVE per default**.
- Una finestra si differenzia da una porta per il semplice fatto che, pur consentendo il passaggio da una locazione ad un'altra, non è possibile chiuderla a chiave.

Caricamento/salvataggio dei mondi

La **modalità** di caricamento e salvataggio dello stato del mondo virtuale rimane aperta: ciascun gruppo può decidere un formato (testuale o binario) atto a contenere un'istantanea dello stato di gioco (ad esempio, mediante serializzazione). In qualunque momento dev'essere possibile salvare lo stato di gioco ed effettuare il ripristino di uno stato precedentemente salvato.

Interfaccia testuale: l'interfaccia testuale può essere implementata seguendo uno tra i seguenti schemi:

1. **comandi:** la singola interazione avviene tramite una linea di testo contenente un comando espresso secondo un certo formato. Esempi: "prendi chiave dell'agente Smith" o "prendi chiave" o "entra nella porta". Attenzione: non è necessario accordare preposizioni e articoli né coniugare verbi. Ad esempio, potete accettare comandi quali "entra in porta" piuttosto che "entra nella porta" o "entra nella studio" o "entra studio".
2. **menù testuale:** tramite stampa a video di un menù testuale (una lista di tutte le scelte che è possibile eseguire in un dato momento) e successiva cattura della scelta dell'utente (ad esempio tramite codice numerico). Esempio:

Puoi eseguire le seguenti azioni:

1. Prendi
2. Parla
3. Usa
4. Apri
5. ...
6. Esamina

Inserisci la tua scelta: 2

Presentando, a seguito della scelta dell'azione, l'elenco degli oggetti compatibili con tale azione e così via.

Separazione logica da GUI: **NOTATE MOLTO BENE** che nella valutazione del progetto verrà considerato come **fondamentale** l'aspetto di separazione tra 1) logica del progetto (il motore sottostante), 2) interfaccia grafica, 3) interfaccia testuale (quando richiesta), 4) I/O.

Quando necessario, dovrebbe essere considerato anche un insieme di **eccezioni**, gestendo eventuali situazioni anomale ad esempio in fase di I/O o di inserimento comandi. Di seguito alcuni esempi:

- comandi con sintassi errata: "parla con porta" o "prendi" (senza argomenti)
- comandi impossibili da eseguire: prendere un oggetto inesistente, parlare con un personaggio non presente nella stanza, etc.

Le eccezioni possono essere stampate a video (es. `e.printStackTrace()`) e gestite in modo da proseguire l'esecuzione del programma.

Modalità di consegna del progetto

Per consegnare correttamente il progetto è necessario consegnare un file **.zip** (o **tar.gz**) contenente i seguenti file:

- **.jar**: questo file contiene i sorgenti `.java`, i file compilati `.class` e le librerie necessarie all'esecuzione del progetto mediante l'istruzione:

```
java -jar nome_vostro_progetto.jar
```

- **.doc** o **.pdf**: relazione del progetto in cui vengono esplicitate e giustificate le scelte progettuali e implementative. Tale relazione deve contenere inoltre un diagramma delle classi, senza la specifica di metodi e attributi;
- **javadoc**: la javadoc generate per le classi del progetto.