

SPECIFICHE DEL PROGETTO v1.01

Metodologie di Programmazione A.A. 2011-2012

Corso di Laurea Triennale in Informatica

Il progetto consiste nella realizzazione di una piccola avventura in Java. Per progetti individuali, si richiede solo una versione grafica dell'avventura (in cui l'interazione, input/output ecc. avvengono mediante GUI). **Per progetti di due o tre studenti si richiede anche la realizzazione di una versione testuale (ulteriori specifiche richieste solo a tali gruppi sono evidenziate in giallo nel testo).**

Si deve costruire e gestire un mondo virtuale che contiene le locazioni del gioco (ad esempio, stanze, ambienti esterni, ecc.) e i personaggi del mondo. All'inizio di una partita, il mondo è costruito leggendone la composizione da file. In qualsiasi momento è possibile salvare su file lo stato di gioco (ovvero una fotografia del mondo virtuale, incluse locazioni, stato dei personaggi, degli oggetti ecc.) e caricare uno stato di gioco precedentemente salvato. Il file deve poter essere salvato in un qualsiasi percorso (pathname) specificato dall'utente. Il formato in cui memorizzare lo stato del mondo virtuale sarà rilasciato successivamente.

Locazioni

Una locazione possiede un nome (ad esempio, "porto", "cucina di casa Rossi", ecc.) e tiene traccia dei personaggi e degli oggetti ivi presenti in un dato momento.

Oggetti

Un oggetto è caratterizzato da un nome (ad esempio, "chiave di casa Rossi"). Ogni oggetto può esporre uno o più comportamenti, ovvero gli oggetti possono essere:

- **aperti/chiusi:** ad esempio, una porta, un baule, un armadio, ecc.
- **letti:** ad esempio, un libro o una lettera
- **spinti/tirati:** un bottone, una leva, ecc.
- **esaminati:** applicabile a tutti gli oggetti
- **cercati:** oggetti che possono contenere altri oggetti (ad esempio, un cofanetto, un cassetto, ecc.)
- **accesi/spenti:** una lampada, un computer, ecc.
- **usati:** da soli (per esempio, un letto, per dormire) e con altri oggetti (ad esempio, una chiave usata con una porta)

Realizzare i seguenti oggetti: porta, finestra, **armadio, libreria, cassetto, scrivania, sedia, lampada, lampadario, cestino, letto, vaso**, tappeto, botola, libro, lettera, baule, bottone, leva, cofanetto, computer, **spago**, mappa, denaro, **quadro**, cassaforte, **coltello, bastone**, divano, televisore, biglietto, dischetto per computer, telefono, cabina telefonica, gettone telefonico, **slot machine**, chiave, **boccale, grog, matrioska**, passaggio segreto, pillola.

Un oggetto può eventualmente essere di proprietà di un determinato personaggio (anche se

non presente nel suo inventario). Ad esempio, un dischetto può essere di proprietà di Mario, ma può essere preso (“rubato”) da Luigi.

Personaggi

Ogni personaggio è caratterizzato dalle seguenti informazioni: nome, locazione in cui si trova, inventario degli oggetti che possiede con sé, memoria degli eventi che conosce e legami che il personaggio ha nei confronti di altri personaggi.

Un personaggio può eseguire le seguenti azioni:

- **dai** oggetto a personaggio
- **prendi** oggetto
- **cerca** in un oggetto (ad esempio in un baule)
- **spingi, tira, apri, chiudi, esamina, usa** oggetto
- **parla con** un personaggio, il quale rivelerà eventualmente informazioni rilevanti
- **domanda** qualcosa a un personaggio
- **entra** in un passaggio che porta a un'altra locazione (es. attraverso una porta aperta)
- **perdi la memoria**

I legami che un personaggio può avere nei confronti di un altro personaggio sono: conoscenza, amicizia, inimicizia, amore (mutuamente esclusivi tra loro). Questi legami vengono usati nell'implementazione delle azioni verbali (“parla con” e “domanda”) per determinare la reazione del personaggio con cui si parla. I legami possono variare in negativo o in positivo nel corso del gioco sulla base di determinate azioni o sequenze di azioni effettuate dal giocatore (ad esempio, dando del denaro o una lettera d'amore, rubando, ecc.). Se i legami non sono implementati (progetto realizzato da 1 studente), le azioni “parla con” e “domanda” possono avere un comportamento deterministico o casuale.

Ogni volta che un personaggio effettua un'azione, questa viene registrata nella memoria di tutti i personaggi presenti nella stessa locazione. Ad esempio, quando un personaggio p prende un oggetto x, tutti i personaggi presenti registrano il fatto che p ha preso l'oggetto x. Ancora, quando si passa da una stanza all'altra, i presenti della prima registrano l'uscita dalla stanza e i presenti della seconda registrano l'ingresso nella nuova stanza. Ogni evento è costituito da un soggetto, l'azione cui l'evento si riferisce (anche non operativa, ad esempio “possiede”) e un elenco di oggetti. Ad esempio, l'evento “il personaggio p esce dalla locazione l” è memorizzato mediante l'inserimento dell'evento (p, ESCE_DA, l) nella memoria dei presenti della locazione l e la cancellazione dagli stessi dell'evento (p, SI_TROVA_IN, l). Analogamente, l'evento “il personaggio p dà l'oggetto o al personaggio p'” genera l'inserimento di due eventi: (p, DA', o, p') e (p', POSSIEDE, o), e la cancellazione dell'evento (p, POSSIEDE, o).

L'azione **domanda** per prima cosa causa la memorizzazione dell'evento (p, DOMANDA, e, p') dove p è il parlante, p' l'ascoltatore ed e è l'evento detto da p a p'. Inoltre il personaggio p' reagisce rispondendo con un altro evento che dipende dal legame tra i due personaggi e la conoscenza che il personaggio p' ha dell'evento e. La scelta su come implementare questo

meccanismo è lasciata allo studente. Per l'azione **parla con**, invece, sarà il personaggio p' a rivelare (sulla base del legame) uno o più eventi potenzialmente interessanti per il personaggio p.

Conclusione di una partita

La partita si conclude quando il giocatore ha ottenuto uno speciale oggetto o ha raggiunto una speciale locazione del mondo virtuale.

Interazione mediante interfaccia grafica

Nella versione con interfaccia grafica, la locazione deve avere una sua visualizzazione grafica, anche spartana e/o schematica (ad esempio, a griglia vista dall'alto, mediante menù a tendina, ecc.), compresi gli oggetti e i personaggi nella stanza. I comandi del giocatore devono essere visibili a schermo, selezionabili e componibili con gli oggetti e i personaggi in modo da formare un'azione da eseguire (per un esempio, si vedano le interfacce SCUMM: <http://www.scummvm.org>).

Interazione mediante interfaccia testuale

All'inizio del gioco, e ogni volta che cambia locazione, il giocatore riceve una descrizione testuale della stanza in cui si trova, inclusi gli oggetti e i personaggi in essa contenuti. Il personaggio effettua le proprie azioni digitando il comando testuale che verrà interpretato da un apposito parser. Ad esempio, per prendere l'oggetto di tipo chiave chiamato "chiave del signor Smith", si potrà digitare:

```
Neo> prendi chiave dell'agente Smith
```

ma anche:

```
Neo> prendi chiave
```

Tuttavia, essendo il secondo comando potenzialmente ambiguo poiché denota la classe di oggetti (e non l'oggetto specifico), nel caso in cui la locazione contenga più chiavi sarà chiesto al giocatore di specificare quale chiave prendere. Analogamente per:

```
Neo> entra nella porta per la cucina
```

oppure

```
Neo> entra nella porta
```

Seguono alcuni esempi di interazione verbale tra personaggi presenti nella stessa stanza:

```
Neo> parla con Morpheus
```

```
Morpheus> l'agente Smith si trova nell'appartamento dell'Oracolo!
```

Neo> domanda a Morpheus dove si trova la cabina telefonica
Morpheus> la cabina telefonica si trova nella quinta strada

Neo> domanda a agente Smith dove si trova la cabina telefonica
Agente Smith> signor Anderson, non posso rivelarle quest'informazione!

Caricare, eseguire e salvare intere partite (per gruppi da tre studenti)

Il progetto prevede anche la possibilità di caricare (da file), salvare ed eseguire lo script di un'intera partita, ovvero la sequenza delle azioni che il giocatore effettua durante una partita. Le azioni vengono eseguite una dopo l'altra in una modalità non interattiva.

Valutazione

Il progetto sarà valutato in base ai seguenti aspetti (in ordine di importanza):

1. Correttezza e rispetto delle specifiche
2. Qualità della progettazione
3. Chiarezza e completezza della documentazione
4. Efficienza

Consegna

La documentazione richiede una breve relazione che descrive le responsabilità di tutte le classi dell'applicazione e le relazioni tra di esse, possibilmente con l'aiuto di un diagramma UML (solo classi). La relazione deve contenere i nomi degli autori e deve essere in uno dei seguenti formati: PDF o DOC. Inoltre, le classi e i metodi, nei file sorgenti, vanno documentati tramite javadoc e le implementazioni vanno adeguatamente commentate.

Tutti i file del progetto vanno consegnati in un unico file JAR generato seguendo scrupolosamente le indicazioni fornite nella pagina del progetto. I sorgenti del progetto **devono** essere compilabili (senza errori) ed eseguibili nella piattaforma Java SE 1.6. Questo implica che nessuna libreria esterna, oltre a quelle di Java SE 1.6, può essere usata.