MANUALE DEL GIOCO DEL MUSEUM TEAM

Alessio Marchetti
Federico Cosma
Sirio Trentin
Edoardo Cavaliere

DESCRIZIONE AD ALTO LIVELLO

DINAMICHE DI GIOCO

Il team si è concentrato sullo sviluppo di un gioco ambientato in un museo.
L'obiettivo è quello di raggiungere un determinato profitto rubando più opere possibili.
Il giocatore ha a disposizione un tempo limitato (rappresentato da un timer) e deve fare attenzione a non incrociare la guardia che può, in modo casuale, muoversi per tutte le stanze del museo (se il giocatore e la guardia si trovano nella stessa stanza il giocatore perde 5 secondi). Se il timer scade la partita è persa e si può decidere se avviarne una nuova o caricarne una precedentemente salvata.
Esistono tre diversi tipi di opere (quadri, sculture e gioielleria), e ogni tipo di opera richiede uno specifico strumento per essere rubato:
1. Coltello, per i quadri.
2. Martello, per le sculture.
3. Telecomando (per disattivare i laser), per la gioielleria.

Il giocatore può portare nell'inventario un solo strumento alla volta. Se si desidera rubare un tipo diverso di opera bisogna trovare e raccogliere lo strumento apposito (la sostituzione avviene automaticamente nell'inventario).
Ogni opera ha un suo peso associato, e nel caso in cui si voglia trasportarne più di una contemporaneamente non si può superare il limite di 10kg.
Se si è vicini al limite, ci si può dirigere all'entrata ed è possibile svuotare l'inventario con un apposito comando (verrà automaticamente salvato e visualizzato il profitto corrente). Depositando un profitto di 12.500.000 all'entrata entro il tempo limite la partita è vinta.

COMANDI DEL GIOCO

1. North, south, east, west: movimento nella mappa (sempre che il movimento sia consentito dalla posizione attuale).
2. Back: ritorno alla stanza precedente.
3. Look: visualizzazione della lista di opere e strumenti all'interno di una stanza.
4. Take: raccogliere opere e strumenti all'interno di una stanza.
5. Take partial profit: serve a svuotare l'inventario, per calcolare e visualizzare il profitto corrente (può essere eseguito solo nella stanza di entrata).
6. Save: serve a salvare l'attuale configurazione della partita (se si sceglie di salvare la partita, si deve dare un nome al salvataggio. Utilizzando un nome già usato si sovrascrive il salvataggio precedente).
7. Load: Caricare una configurazione precedentemente salvata (utilizzabile soltanto se sono disponibili salvataggi precedenti).
8. Exit: per uscire dalla partita.
9. Use: comando non implementato, dato che nel momento in cui viene avviato il comando Take automaticamente viene utilizzato uno strumento.

N.B: per i comandi Load e Save l'utente deve predisporre un Google Cloud Bucket e caricare la chiave privata nella radice del progetto.

DESIGN PATTERNS

1. Controller: questa classe si occupa di agire da interfaccia tra grafica e sistema. Gli input dell'utente vengono gestiti da questa classe che successivamente chiama i metodi corretti delle altre classi.
2. Singleton: la classe Controller è stata realizzata anche tramite il design pattern Singleton. Questo significa che la classe viene istanziata una sola volta, e viene riutilizzata durante tutta l'esecuzione del codice (esempio nella classe Timer).
3. High Cohesion: tutte le classi sono state implementate per essere più atomiche possibili.
4. Low Coupling: la modifica di una classe deve impattare il meno possibile il resto del codice.

COME INSTALLARE E LANCIARE IL SOFTWARE

Clonare la respository da GitHub e aprire il progetto tramite Visual Studio Code. È necessario modificare la variabile vmArgs all'interno del file launch.json inserendo il percorso di installazione di javafx.

VERSIONI SOFTWARE UTILIZZATE

Per la realizzazione del progetto è stato utilizzato l'ide Visual Studio Code. Il linguaggio di programmazione utilizzato per il codice è java 23.
Per poter creare l'interfaccia grafica dell'applicazione client è stato utilizzato javafx 23.0.1. Abbiamo usato SceneBuilder, un software usato per realizzare la GUI Javafx più comodamente e per la generazione del file XML.

LIBRERIE UTILIZZATE

1. Jackson (versione 2.12.3), utilizzato per la serializzazione e deserializzazione di file Json.
2. Google Cloud Storage (versione 2.47), utilizzata per interfacciarsi con il bucket di Google.
3. Junit (versione 5.9.2), framework per il test automatico delle classi.