



DESCRIZIONE GENERALE DEL GIOCO

Il gioco realizzato è un'avventura testuale che tratta di un personaggio che deve affrontare le varie avversità di un maestoso castello con l'obiettivo finale di salvare un suo compagno Erik che è stato rinchiuso nella prigione di esso.

Per far orientare il giocatore è stata creata una mappa che raffigura le varie stanze del castello in cui l'utente deve avventurarsi per poter ottenere tutti gli oggetti e la chiave che aprirà la prigione.

Però dovrà fare attenzione a quale porta del castello varcherà perché alcune potrebbero portare alla riduzione della sua vita e alla fine del gioco.

Inoltre il giocatore non ha a disposizione tutto il tempo che vuole, ha i minuti contati per portare in salvo il suo compagno e dovrà impegnarsi a scegliere la strada giusta senza nessuna perdita di tempo, perché rischierebbe che la partita finisca prima che riesca a salvare in tempo il suo amico.

All'avvio del gioco viene data all'utente la possibilità di attivare o meno una musica di sottofondo che serve all'utente per immergersi a pieno nell'avventura che deve affrontare.

Subito dopo l'introduzione viene mostrato l'elenco dei comandi principali, per rendere il gioco accessibile anche a chi non conosce questa tipologia di giochi. L'utente si ritroverà all'ingresso del castello, da cui si potrà spostare attraverso i comandi ("nord", "sud", "ovest")

o “est”), raggiungendo così altre stanze con oggetti da raccogliere (attraverso il comando “prendi”) e utilizzare/indossare (con il comando “usa/indossa”).

È stato inoltre aggiunto un meccanismo che simulerà la vita del giocatore attraverso una percentuale “%” che potrà aumentare o diminuire, in base a determinati eventi compiuti dal giocatore (per esempio entrare in determinate stanze o prendere oggetti), per cui bisogna fare molta attenzione perché il gioco potrebbe concludersi prima del previsto se la percentuale di vita diminuisce troppo.

MAPPA

Ecco la rappresentazione della mappa dell'avventura testuale presentata:



Dettagli Implementativi

File

La classe “Files” fornisce i metodi per salvare e caricare lo stato di una partita su un file. Questa classe è essenziale per permettere ai giocatori di continuare il loro gioco daò punto in cui lo avevano salvato, conservando i progressi effettuati, come gli oggetti presi o utilizzati, la stanza in cui si trovavano nel momento del salvataggio e la percentuale di vita del giocatore.

I dati verranno salvati all’interno di un file chiamato “castle_save”.

Al richiamo del comando carica, i dati di gioco contenuti nel file sostituiranno i dati della partita in corso. In questo modo sarà possibile continuare la partita precedentemente salvata.

```
--->---> vite 70 %  
Sei nella Sala dei Segreti!!  
--->SALVA  
  
Salvataggio in corso...  
Partita salvata correttamente.  
|
```

```
CARICA  
  
--->---> vite 70 %  
Sei arrivato nella Sala delle Armature  
--->INVENTARIO  
  
Nel tuo inventario hai:  
- Tuta protettiva  
- Spada
```

Lambda Expression

Le lambda expression sono state utilizzate per in login per eseguire la creazione e la visualizzazione dell'interfaccia grafica.

Inoltre servono per collegare gli action listener ai vari bottoni della javaSwing, in modo da assegnare un determinato comportamento ad ogni bottone, in questo caso nel metodo MostraDialogoSceitaMusica per mostrare un dialogo di conferma per l’attivazione della musica.

Swing

La JavaSwing è il framework di java che permette la creazione di interfacce grafiche, ed è stata utilizzata nella parte iniziale del nostro codice per consentire di creare un'interfaccia grafica per il login del gioco in modo semplice e strutturato. Swing fornisce tutti gli strumenti necessari per gestire la GUI e le interazioni dell'utente.

Database

Abbiamo utilizzato il database per memorizzare l’username e il tempo del giocatore nel momento in cui termina la partita.

Con queste informazioni memorizzate nel database abbiamo creato una classifica dei giocatori che riporta l’username del giocatore e il tempo di gioco, in ordine crescente, da chi è

stato più veloce a concludere la partita a chi lo è stato meno. Se lo stesso giocatore vorrà nuovamente giocare all'avventura, se esso entra con lo stesso username e il tempo di gioco della partita sarà minore di quella precedente, il tempo di quel giocatore verrà aggiornato col nuovo tempo, altrimenti se il tempo sarà maggiore lascerà il suo tempo di gioco migliore.

```
<*****>
      lista dei giocatori:
<*****>
Elisabeth Picard 19
Mary 32
VINCENZA 109
```

```
<*****>
      lista dei giocatori:
<*****>
VINCENZA 13
Elisabeth Picard 19
Mary 32
```

Thread

L'uso di «EventQueue.invokeLater» e «SwingUtilities.invokeLater» è una pratica per assicurarsi che il codice che interagisce con l'interfaccia grafica venga eseguito nel thread.

Viene utilizzato il thread per gestire tutti gli eventi e le operazioni dell'interfaccia utente per evitare comportamenti inaspettati e garantisce che venga aggiornata in modo sicuro.

I thread vengono utilizzati anche per gestione l'esecuzione della musica di sottofondo senza bloccare l'applicazione, garantendo che l'interfaccia utente rimanga reattiva e l'esperienza utente sia fluida.

API REST

Nel nostro progetto, abbiamo implementato una soluzione per generare un nome casuale utilizzando un'API REST pubblica. Questo approccio viene utilizzato nel caso in cui l'utente non inserisca il proprio nome utente o entri nel gioco senza fornire alcun dato. In tali situazioni, un nome casuale verrà automaticamente recuperato dall'API REST e assegnato all'utente. Questo consente all'utente di iniziare a giocare e proseguire con la propria avventura senza interruzioni, anche in assenza di un nome utente specificato. Dunque il nome casuale è utilizzato come soluzione alternativa per permettere all'utente di continuare a giocare senza dover inserire un nome. Le immagini riportano il caso in cui l'username viene inserito dall'utente e il caso in cui viene generato un nome casuale random.

```
*****
Nome utente inserito: vincenza
***** Sei entrato nel gioco *****
Benvenuto nel Castello

Ti trovi di fronte ad un maestoso castello che si erge fiero e misterioso nella valle.
All'interno delle sue mura, il castello si rivela un labirinto di stanze e corridoi,
ognuna con i propri segreti e pericoli; sarà il palcoscenico della tua epica impresa.
Dovrai affrontare le sfide che ogni stanza presenta per salvare il tuo caro amico ERIK.
Il tempo è essenziale!! Il destino del tuo amico dipende solo da te.

TROVA LA STRADA GIUSTA PER AIUTARE ERIK

*****
Digita 'h' per visualizzare i comandi del gioco
```

```
*****
Nome casuale ottenuto: Kadir Karaböcek
***** Sei entrato nel gioco *****
Benvenuto nel Castello

Ti trovi di fronte ad un maestoso castello che si erge fiero e misterioso nella valle.
All'interno delle sue mura, il castello si rivela un labirinto di stanze e corridoi,
ognuna con i propri segreti e pericoli; sarà il palcoscenico della tua epica impresa.
Dovrai affrontare le sfide che ogni stanza presenta per salvare il tuo caro amico ERIK.
Il tempo è essenziale!! Il destino del tuo amico dipende solo da te.

TROVA LA STRADA GIUSTA PER AIUTARE ERIK

*****
Digita 'h' per visualizzare i comandi del gioco
```


Diagramma delle Classi



Classi Principali

Login

La classe “login” contiene l’interfaccia GUI creata con le JavaSwing che permette di gestire l’interfaccia grafica per il login del giocatore ed è responsabile di raccogliere il nome utente del giocatore; l’utente inserisce il proprio nome nel campo di testo e preme il bottone di login per avviare il gioco. Se l’utente non inserisce un nome, viene generato un nome casuale. Durante il login, l’utente ha la possibilità di scegliere se ascoltare la musica di sottofondo. Se il login ha successo, la finestra di login viene minimizzata e il gioco viene avviato.

Castello

La classe “Castello” rappresenta il cuore del gioco, gestisce tutte le operazioni principali e le interazioni con l’ambiente di gioco. La sua implementazione dettagliata e strutturata permette di creare un’esperienza di gioco coinvolgente e dinamica.

Descrizione gioco

La classe “descrizioneGioco” fornisce la struttura completa del gioco; definisce e inizializza le istanze del gioco, creando e collegando le varie stanze in una mappa e definendo i vari comandi e gli oggetti. Gestisce l’inizializzazione del gioco, inclusa la creazione delle stanze, l’associazione dei comandi e l’allocazione degli oggetti all’interno delle varie stanze.

Engine

La classe “Engine” gestisce l’avvio e l’esecuzione principale del gioco. Essa coordina le interazioni tra il giocatore e il sistema, interpretando i comandi inseriti e aggiornando lo stato del gioco di conseguenza. Inoltre questa classe contiene un metodo per ottenere un nome casuale da un’API rest esterna.

Oggetto

Questa classe è progettata per gestire vari attributi e stati che un oggetto può avere, rendendo possibile la creazione di oggetti con proprietà e comportamenti diversi. Inoltre questa classe implementa l’interfaccia Serializable per permettere la serializzazione degli oggetti.

Stanza

La classe “Stanza” fornisce una struttura per definire le proprietà di una stanza, come un valore booleano che indica se la stanza è accessibile o meno, gli oggetti in essa contenuti e le connessioni con altre stanze adiacenti. Implementa l’interfaccia Serializable, permettendo la serializzazione.

.

Specifica Sintattica:

OPERATORI:

- | | |
|--------------------------------------------------------|--------------------------------------------------|
| - <i>newList</i> : () → lista | Crea una nuova istanza di inventario nella lista |
| - <i>isEmpty</i> : (lista) → boolean | Verifica se l'inventario è vuoto |
| - <i>getLista</i> : (posizione, lista) → lista | Restituisce la lista nella posizione specificata |
| - <i>setLista</i> : (tipoelem,posizione,lista) → lista | Imposta un elemento alla posizione specificata. |
| - <i>addLista</i> : (tipoelem,posizione,lista) → lista | Aggiunge un elemento alla posizione specificata |

Specifica Semantica:

Tipi :

- inventario: insieme delle liste $I = \langle a_1, a_2, \dots, a_n \rangle$, $n \geq 0$, di elementi di tipo `tipoelem` dove l'elemento i -esimo ha valore $a(i)$ e posizione $pos(i)$
- boolean: insieme dei valori di verità

1.newLista

Precondizioni: Nessuna

Postcondizioni: crea un oggetto `Inventario` con una lista vuota

2. isEmpty

Precondizioni: Nessuna

Postcondizioni: Restituisce true se la lista è vuota, altrimenti false

3. getLista

Precondizioni: La posizione deve essere un indice valido nella lista ($0 \leq \text{posizione} < \text{lista.size}()$).

Postcondizioni: Restituisce l'oggetto alla posizione specificata.

4. setLista

Precondizioni: La posizione deve essere un indice valido nella lista ($0 \leq \text{posizione} < \text{lista.size}()$).

Postcondizioni: L'oggetto alla posizione specificata è sostituito con il nuovo oggetto.

5. addLista

Precondizioni: La posizione deve essere un indice valido ($0 \leq \text{posizione} \leq \text{lista.size()}+1$)

Postcondizioni: Inserisce l'oggetto alla posizione specificata, spostando gli elementi successivi.

OPERAZIONI	COSTRUTTORE I'	
	newList()	addLista(o, p, i)
isEmpty(i)	If (isEmpty(newList(i)) != <> Then true else false	If(isEmpty(addLista(i,p,o),o')=<> Then false else true
getLista(p, i)	getLista(newList(i,p) = error	getLista(addList(i,p,o),o' = if o=o' then addLista(p,o') else error
setLista(o, p, i)	setLista(newList(i),p,o)=error	setLista(addLista(o,p,i),o')=if o=o' then o' else setLista(i,p,o')

Processo Di Sviluppo e Organizzazione del Lavoro

Il processo di sviluppo di questa avventura è avvenuto secondo il modello Scrum, ovvero in modo iterativo e man mano incrementale.

Abbiamo utilizzato la piattaforma Microsoft Teams, già a noi nota e pertanto scelta in quanto ci è risultata più comoda avendo già dimestichezza con essa; durante le riunioni, abbiamo sfruttato la funzionalità di condivisione schermo e audio in modo tale sia di lavorare insieme e poter interagire al meglio tra noi.

Giornalmente sono stati stabiliti quali task portare a termine prima e quali in un secondo momento in base alla loro difficoltà e quantità di lavoro da svolgere; per organizzarci ancora meglio, ci siamo dati delle scadenze interne e quindi un tempo massimo da rispettare per portare a termine i vari problemi in tempo utile.

I problemi più impegnativi sono stati affrontati in riunione così da lavorare insieme scambiandoci feedback e aiutandoci in tempo reale.

Soluzione del Gioco

Il giocatore si trova all'ingresso del castello da cui dovrà necessariamente spostarsi attraverso i comandi a disposizione.

Andando verso sud, si ritroverà nel corridoio 1, da cui sarà possibile accedere alla sala degli elfi con il comando ovest o alla sala delle armature con il comando est. All'interno della sala delle armature l'utente dovrà utilizzare il comando guarda, prendere e utilizzare/indossare gli oggetti che ci sono (spada e tuta protettiva) che serviranno per aumentare la percentuale di vita del giocatore.

Successivamente il giocatore dovrà proseguire nel corridoio 2 attraverso il comando sud, e andare ancora verso sud per ritrovarsi nella sala dei segreti in cui ci sarà la chiave che servirà successivamente per aprire la prigione.

Il giocatore dovrà proseguire verso sud nel corridoio 3 e ancora verso sud si troverà di fronte alla porta della prigione, qui attraverso il comando usa chiave, si sbloccherà l'accesso alla prigione. Il giocatore così sarà riuscito ad aprire la prigione e portare in salvo il suo amico Erik così da concludere con successo l'avventura testuale.

Dovrà fare attenzione al tempo in quanto è limitato, non avrà tutto il tempo a sua disposizione!

La sala degli elfi e la sala delle vipere invece sono stanze nelle quali non è necessario entrare per poter poi varcare la porta della prigione. Entrando in queste stanze la vita del giocatore verrà danneggiata e ridotta, dovrà fermarsi in tempo e trovare la strada giusta perché in questo modo il gioco potrebbe concludersi per l'esaurimento della percentuale di vita. Queste stanze condurranno all'uscita sul retro del castello, dalla quale il giocatore può decidere spontaneamente di uscire dal gioco o tornare indietro e trovare la strada giusta che condurrà dal suo amico Erik.