DOCUMENTAZIONE

Gruppo GG-EZ, partecipanti:

1) Christian Novara

2) Antonio Martiradonna

3) Giuseppe Miglionico

**INTRODUZIONE:**

Lo scopo del progetto è stato quello di creare non solo un software solido che rispecchi i principi della programmazione ad oggetti e gli argomenti trattati nel corso di M.A.P , ma un gioco vero e proprio, utilizzando una storia e disegni originali(le ambientazioni e i personaggi sono frutto del lavoro di amici grafici), in modo da creare un'avventura che unisca fantasy e mistero nel fascino delle avventure grafiche anni 90. Per la realizzazione di questa impresa è stato scelto di utilizzare la libreria LibGDX:

Il gioco consente di affrontare più livelli, ogni livello ha un’ambientazione esplorabile, il giocatore può cliccare sullo sfondo per interagire in vari modi, inoltre ha a sua disposizione un inventario.

Il suo scopo sarà superare i livelli a seconda degli enigmi proposti dai personaggi incontrati nell’avventura.

Per spiegare le scelte progettuali è utile prima fare un’introduzione della potenza di LibGdx:

**INTRODUZIONE ALLA LIBRERIA GRAFICA LIBGDX:**

Abbiamo impostato il nostro progetto basandosi sulla libreria grafica per Java: LibGDX(<https://libgdx.badlogicgames.com/>)

La libreria LibGDX permette di sviluppare giochi open source e multipiattaforma (Desktop, android ,ios e browser web) ed inoltre  permette di creare delle scene sia in 2D ma anche in  3D (abbiamo scelto la grafica in 2D).

Nonostante la libreria metta a disposizione numerose funzioni, quelle che abbiamo reputato essenziali per la creazione di un gioco punto e clicca sono:

1) L'interfaccia Screen per l'implementazione dei vari metodi per la visualizzazione della parte grafica su schermo, nel nostro caso le classi del **package “Schermate”** implementano tutte l’interfaccia Screen, questo perchè sono usate per il solo scopo di caricare le texture dei livelli e mostrarli su schermo (la funzione **render()** si aggiorna ogni frame e si occupa di renderizzare su schermo

2) L'interfaccia InputProcessor che ci permette di prendere come input il click del mouse così da poter ricreare lo stile dei giochi punta e clicca. Nel nostro caso la classe **Menu** utilizza questa interfaccia per prendere il click del mouse.

Le classi principali che la libreria mette a disposizione, delle quali noi abbiamo usufruito sono: **Actor(Image) e Stage**

-La classe Actor rappresenta il componente primitivo della classe Stage il quale  può essere un’immagine,una  texture, un font per la scrittura ecc.. Ogni elemento è associabile ad una parte grafica ed è definito come Actor, nel nostro caso abbiamo usato una classe più specifica che ci aiuta nella definizione delle texture, la classe **Image** (estensione della classe Actor)

-La classe Stage implementa una struttura dati che contiene gli Actor e permette di manipolarli. In questo modo è possibile impostare la loro posizione(dando come input delle coordinate ben precise), grandezza, priorità (quale immagine deve essere posizionata in primo piano e quali in secondo). Gli **oggetti stage** sono utilizzati in ogni classe del package ‘**Schermate’**  e nella classe ‘Menu’ del package **Azioni**

-La classe **Group** è un altro tipo di contenitore degli Actor che permette la loro organizzazione in gruppo separati,così è possibile gestire in modo più semplice una parte dell’interfaccia grafica senza modificarne il resto. Il group è usato nelle classi del package Schermate e nella classe Menu del package Azioni.

Infine esiste un’altra forma di stage chiamata **Table**, questa struttura è anch’essa un contenitore grafico come uno stage con la differenza che è possibile aggiungere attori **secondo un determinato ordine**, è usata nella classe Menu per la visualizzazione grafica dell’ inventario.

**DETTAGLI SCELTE PROGETTUALI EFFETTUATE**

 Abbiamo deciso di suddividere il nostro progetto in 3 package così da associare ad ogni singolo package un compito ben preciso.

-Il package Azioni presenta al suo interno tutte le classi che permettono di gestire le azioni di gioco: 1)Classe Inventario: Implementa una **lista di oggetti**, contiene quindi i metodi per aggiungere e rimuovere gli oggetti dalla lista,inoltre implementa anche una **table** usata per la visualizzazione grafica dell’inventario

2)Classe Menu: questa classe gestisce tutta la parte grafica relativa alle azioni messe a disposizione per il gioco(parlare,osservare,raccogliere,usare), inoltre gestisce tutta la logica dietro di esse

3)Classe Puzzle: questa classe gestisce la logica dei ‘puzzle’ , un **puzzle** è la logica con il compito di rendere certe interazioni con l’ambiente inutilizzabili a meno che il giocatore non abbia un certo oggetto nell’inventario, ESEMPIO: una certa porta è utilizzabile solo se si possiede la sua chiave

Inoltre la classe gestisce lo spostamento tra i vari livelli per le ‘interazioni teletrasporto’ , ovvero **interazioni** che consentono lo spostamento da un livello all’altro liberamente senza bisogno di determinati oggetti.

-Il package Schermate ci permette di contenere le classi che riguardano la creazione dei vari livelli.

Esempio: Classe ‘Capitolo1’ ha il compito di caricare il primo livello, quindi cariare texture, Interazioni , dialoghi e oggetti per l’inventario, nello stesso package sono presenti ‘Capitolo2’,Capitolo2\_2’,’Capitolo3, ‘Capitolo3\_3’, ‘SchermataIniziale’(chiede il nome del giocatore) e ‘SchermataFinale’(crediti e ringraziamenti)

-Il package WorldElement contiene le classi che riguardano gli elementi che popolano il mondo di gioco. È Composto da

1)Classe Combinazioni: contiene le istanze degli oggetti frutti di combinazione e gli oggetti ‘speciali’, ovvero usati per lo spostamento da un livello all’altro appartenenti allo stesso capitolo

2)Classe Interazione: questa classe serve a istanziare le entità interagibili dell’ambiente, ad ognuna è associata una texture e degli stati per riconoscere il tipo di interazione (osservabile, raccoglibile,utilizzabile,personaggio non giocante) insieme ad altri stati (più dettagli nella Presentazione)

3)Classe Oggetto: questa classe serve a istanziare gli oggetti dell’inventario, ad ogni oggetto è assegnata una texture e gli stati utili alla loro combinazione(più dettagli nella Presentazione)

4)Classe Protagonista: istanzia la texture del personaggio protagonista.

5)Classi interazioniCap2 e interazioniCp3: sono utilizzate per l’istanziazione delle interazioni per i capitolo multi livelli