

Relazione progetto JUno

Annarosa Montanari, matricola 1838742, presenza M-Z.

Per la progettazione di JUno sono stati utilizzati i seguenti design patterns:

- **MVC:**
Il model-view-controller è un pattern che permette di separare le varie componenti del progetto consentendo una più semplice modifica del codice ed un'eventuale aggiunta di funzionalità.
Il package "view" comprende tutte le classi usate per costruire l'interfaccia utente.
Il package "model" comprende tutta la logica del gioco, le classi che rappresentano le varie carte, la classe che rappresenta il mazzo e la classe che rappresenta il gioco vero e proprio e che quindi definisce le regole di gioco.
Il package "controller" contiene invece tutte le classi che fanno da collegamento tra il model e la view, e che permettono di gestire le interazioni utente e si occupano di scrivere/leggere sul file.
Per ogni tipologia di interazione utente esiste una classe (che implementa l'interfaccia ActionListener) che permette, in seguito al click di un bottone, di eseguire le istruzioni specifiche.
Le varie classi di EventListener si occupano di eseguire le istruzioni presenti nel model, verificando quindi che siano rispettate le regole del gioco e poi si occupa di comunicare alla view i dati aggiornati in modo tale che possano essere mostrati visivamente anche all'utente sullo schermo.
- **Observer-Observable:**
grazie a questo pattern è possibile rendere la view un'osservatrice del modello e far sì che ad ogni cambiamento di quest'ultimo venga aggiornata anche la view.
In questo modo è possibile gestire le azioni utente, e in caso esse cambino la logica del gioco, viene aggiornata in automatico anche la view.

Specifiche:

- **File:**
I dati dei giocatori (nickname, avatar, punteggi delle partite) sono salvati in un file("fileData.txt") che funge da database.
Questo file viene aperto prima di ogni partita per verificare se il giocatore è già presente in archivio o se è necessario creare un nuovo account.
In quest'ultimo caso chiede ulteriori dati all'utente, tra cui un'immagine e poi salva tutto sul file.
- **Partita completa:**
Nel progetto viene implementata una partita tra l'utente e tre giocatori artificiali.
I giocatori artificiali scelgono la carta grazie a un numero random che decide quale strategia giocare.

- **Interfaccia:**

Il codice dell'interfaccia utente è stato scritto utilizzando Java Swing.

All'interno di un frame principale sono contenuti vari pannelli, ognuno dei quali definisce gli elementi a disposizione dei vari giocatori.

Il pannello utente contiene sia le carte disponibili da giocare sia i bottoni per chiamare uno o per passare il turno.

Una volta cliccato uno di questi bottoni, tramite i vari ActionListener, viene cambiato lo stato del sistema ed eventualmente aggiornata la view.

- **Stream:**

La lettura dei dati avviene tramite stream. Essi permettono sia di verificare che un giocatore abbia già un account registrato cercando il nickname nel file, sia recuperando i dati di un giocatore filtrando le varie righe in base ad una parola chiave.