

GDC
“Giochi del Colosso”

Montanari Nicola, Federico Muccioli, Valeria Verde
23 giugno 2022

Indice

1. Analisi	3
1.1. Requisiti	3
1.2. Analisi e modello del dominio	4
2. Design	5
2.1. Architettura	5
2.2. Design dettagliato	6
3. Sviluppo	24
3.1. Testing automatizzato	24
3.2. Metodologia di lavoro	27
3.3. Note di sviluppo	30
4. Commenti Finali	31
4.1. Autovalutazione e lavori futuri	31
A Guida utente	33

Capitolo 1

Analisi

L'applicativo proposto consiste nella realizzazione di un videogioco che contenga tre differenti giochi da casinò, tutti e tre contenuti in un unico Hub chiamato “**Giochi del Colosso**”. Questo Hub consentirà agli utenti di ottenere la Membership registrandosi tramite la piattaforma stessa. Una volta effettuato l'accesso, l'utente potrà effettuare operazioni di modifica del proprio saldo e partecipare ai vari giochi in totale libertà.

L'utente sarà anche abilitato alla richiesta di eseguire la modifica del proprio Username e Password. Se necessario, il giocatore potrà richiedere l'eliminazione del proprio account, nel caso questo accadesse, l'utente sarà riportato alla schermata di accesso.

1.1 Requisiti

Requisiti funzionali obbligatorie

- Gestione logica e grafica di ogni singolo gioco:
 - BlackJack.
 - Roulette.
 - Baccarat.
- Interfaccia di accesso.
- Interfaccia principale dell'Hub.
- Gestione e creazione di account.
- Memorizzazione dati di account su file (info personali e saldo).
- Possibilità di uscire da un gioco e partecipare a un altro a piacimento.
- Guida al gioco con tutte le regole necessarie e relative quote di vincita.

Requisiti non funzionali

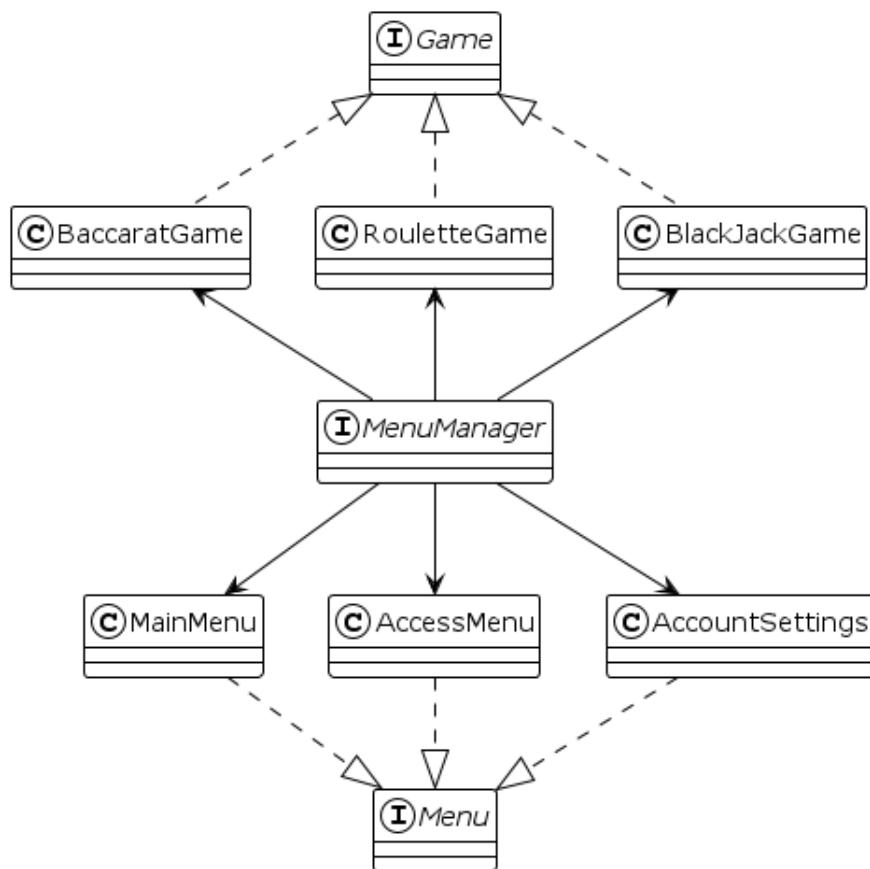
GDC dovrà essere utilizzabile su macchine Windows, Mac OS e Unix/Linux.

1.2 Analisi e modello del dominio

GDC richiederà all'utente di effettuare l'accesso al proprio account, e solo dopo aver verificato che i dati inseriti siano corretti, porterà l'utente al menù principale dell'applicativo, così da fornirgli la possibilità di scegliere a quale gioco partecipare in totale autonomia.

Ogni gioco sarà caratterizzato da un'interfaccia visiva generale dove sarà possibile visualizzare il proprio saldo, i soldi messi in palio e la vincita di ogni partita. Inoltre sarà possibile selezionare una tra le varie fiches disponibili a schermo per modificare la quantità di incremento di valuta messa in gioco ad ogni pressione del pulsante relativo alla scommessa. Un giocatore sarà libero di svolgere quante partite voglia in qualsiasi gioco esso desidera partecipare, a patto che abbia un saldo disponibile maggiore di 0.

Tramite il menù principale dell'applicativo inoltre si potrà accedere ad una zona separata di sola gestione account in cui verranno effettuate le operazioni più importanti, tra cui il deposito o prelievo di denaro, modifica dei dati dell'account oppure l'eliminazione di tale profilo.



Schema UML delle entità del dominio.

Capitolo 2

Design

2.1 Architettura

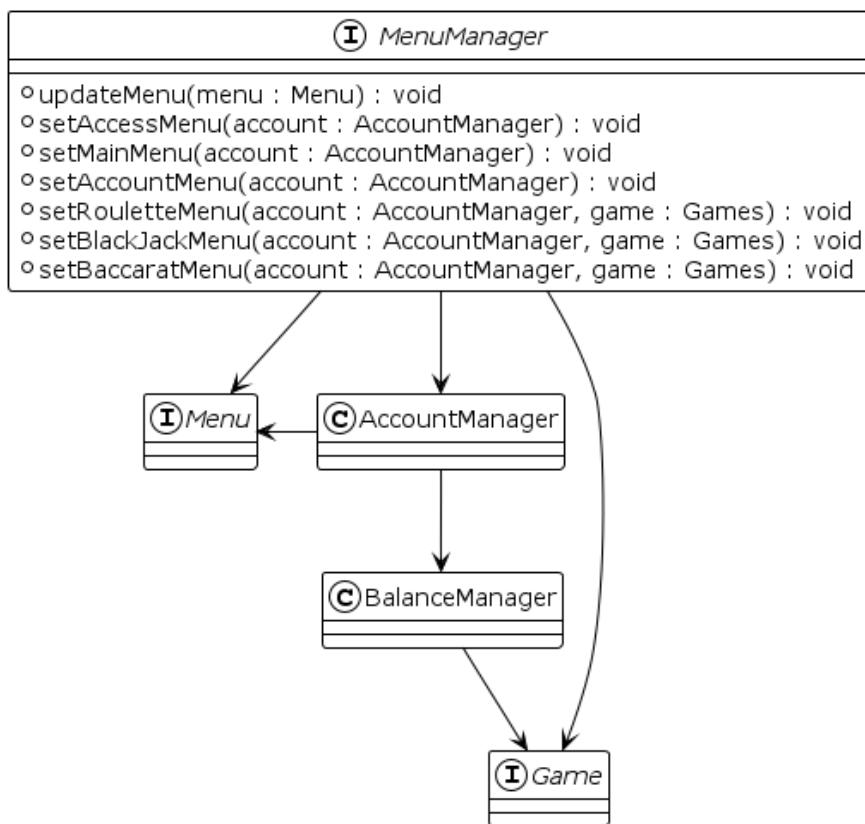
L'architettura di GDC è basata sull'implementazione e utilizzo dell'interfaccia MenuManager che con la sua struttura rende disponibile all'applicazione la gestione dei vari Menu presenti.

Questa interfaccia è caratterizzata da metodi utili allo scambio di Pannelli all'interno della finestra.

Nel momento in cui un pannello specifico viene richiesto, esso verrà impostato nella finestra tramite un metodo all'interno di MenuManager. Ad ogni pannello viene passata la stessa istanza di AccountManager così da mantenere costante e in sicurezza la gestione dell'account tra tutti i pannelli senza incorrere in alcun rischio di perdita o sovrascrittura di dati dell'utente.

Nel caso in cui il pannello richiesto sia quello di un gioco, al posto di AccountManager verrà passato un BalanceManager (ottenibile tramite un AccountManager) in quanto ogni gioco dovrà andare a modificare solamente il saldo dell'utente e non i dati personali.

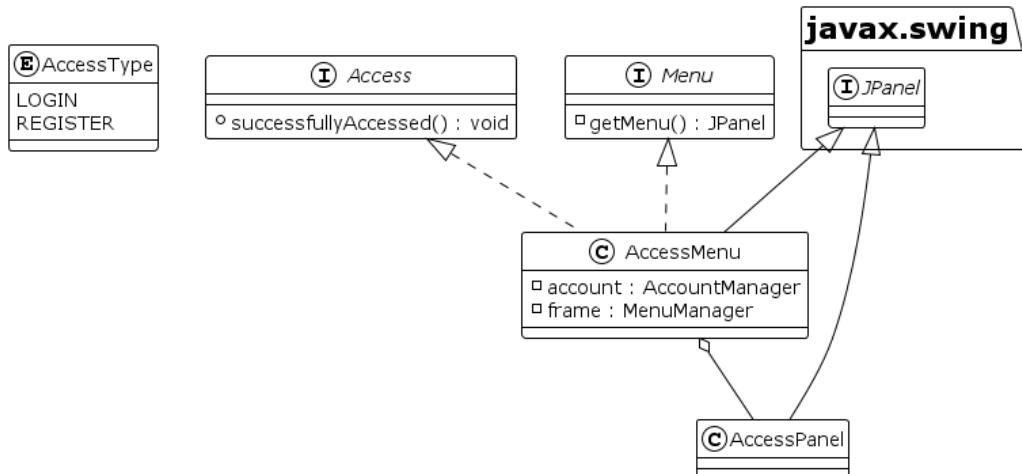
Questi due Manager sono stati implementati con lo scopo di separare totalmente il funzionamento dei vari giochi con la gestione dei vari file di ogni utente.



2.2 Design dettagliato

Montanari Nicola

Gui di accesso alla piattaforma



Affinché ogni utente possa gestire il saldo a proprio piacimento, è stato fondamentale implementare una schermata di accesso ad ogni avvio dell'applicativo, così da consentire ad ogni giocatore di accedere con le proprie credenziali.

In questa interfaccia di accesso, un utente può inserire i propri dati per effettuare l'accesso, oppure registrarsi nel caso fosse la prima volta che utilizza questo programma. In questo modo il software ogni volta che verranno effettuate richieste di modifica saldo (sia per richiesta dell'utente nel caso volesse effettuare un deposito o una ricarica, sia per richiesta di un gioco da casinò nel caso di vittoria o sconfitta) sarà in grado di identificare il giocatore a cui effettuare tali modifiche e di tenere traccia del saldo generale di quel determinato utente. I pulsanti della GUI vengono gestiti tramite degli ActionListener, che in base alla situazione ci permettono di gestire l'intero processo di accesso o registrazione.

User Manager

Ad ogni utente registrato è associato un file JSON contenente tutti i dati necessari al funzionamento di questo applicativo.

- **USERNAME**
 - Nome utente con cui viene identificato il giocatore, utilizzato da quest'ultimo in fase di accesso per entrare nella piattaforma.
- **PASSWORD**
 - Parola di accesso con cui l'utente protegge il proprio account da accessi indesiderati.
- **AGE**
 - Età del giocatore, un giocatore deve avere un'età compresa tra i 18 e i 99 anni per potersi registrare all'interno del software.
- **BALANCE**
 - Valore dell'attuale saldo del giocatore. Questo valore viene modificato ad ogni mano di qualsiasi gioco e a discrezione dell'utente nel caso volesse effettuare depositi o prelievi.

Per la gestione dei dati di ogni singolo utente sono stati creati due diversi manager, il primo chiamato "Account Manager" che avrà il compito di gestire tutte le operazioni necessarie alla creazione o modifica di tutti i dati dell'utente, escluso il saldo.

Il secondo chiamato "Balance Manager" che sarà il gestore di tutte le operazioni di modifiche del saldo quali possono essere deposito e prelievo.

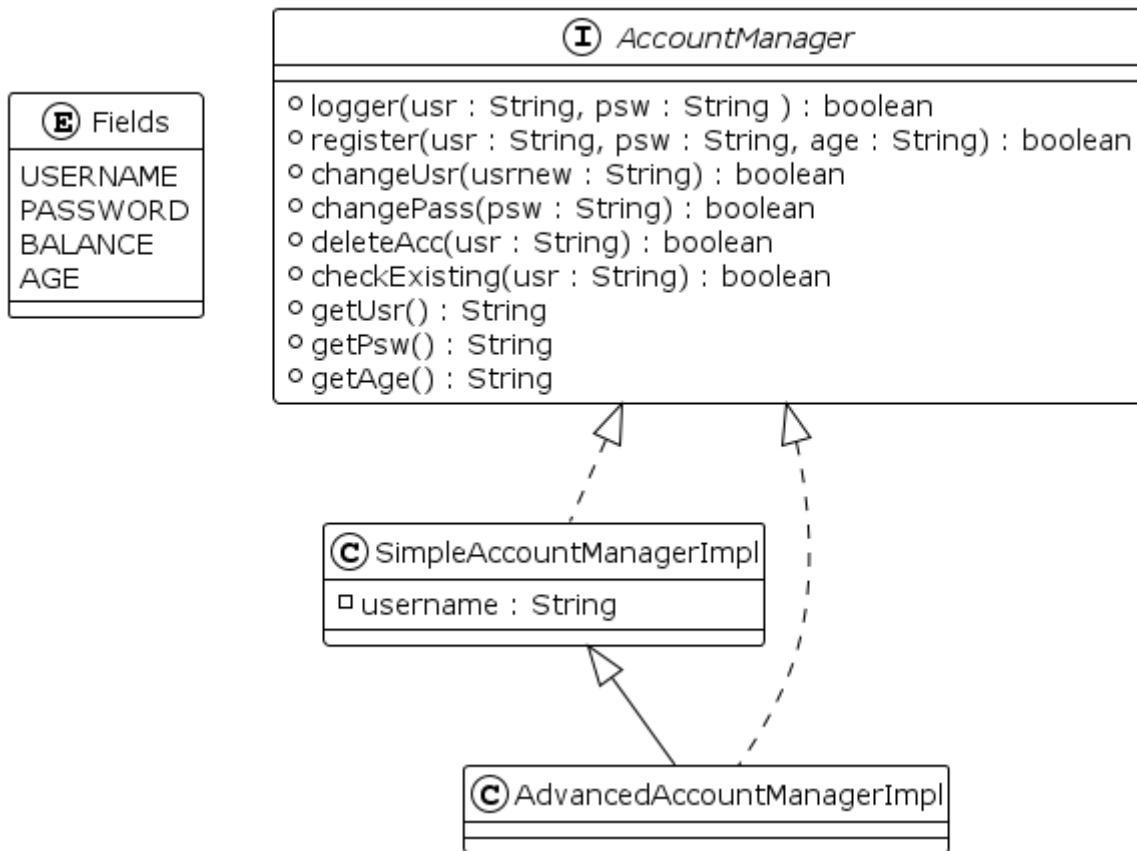
Questa distinzione è stata fatta sia per scorporare i dati dell'utente dal proprio saldo, sia in una visione futura nel caso sia necessaria l'aggiunta di uno o più campi all'interno di ogni utente, così da mantenere invariata l'implementazione e la gestione del saldo e focalizzarsi esclusivamente sui nuovi campi richiesti.

Questa distinzione torna utile anche nell'utilizzo dei vari giochi, in quanto ogni singola partita avrà la necessità di modificare il saldo dell'utente ma non dovrà assolutamente toccare i dati di quest'ultimo.

Ho deciso inoltre di implementare due differenti classi per ogni Manager, una classe "semplice" che andrà a implementare solo le funzionalità richieste e una seconda classe "avanzata" che estende la classe "semplice". La classe avanzata ha il compito di effettuare verifiche preventive prima di richiamare la funzione presente all'interno di quella semplice. Questo da separare l'implementazione dell'effettiva funzionalità dalle verifiche necessarie, anche in un caso futuro in cui si vogliano aggiungere o rimuovere verifiche per una determinata operazione.

Seguono nel dettaglio i due Manager sviluppati:

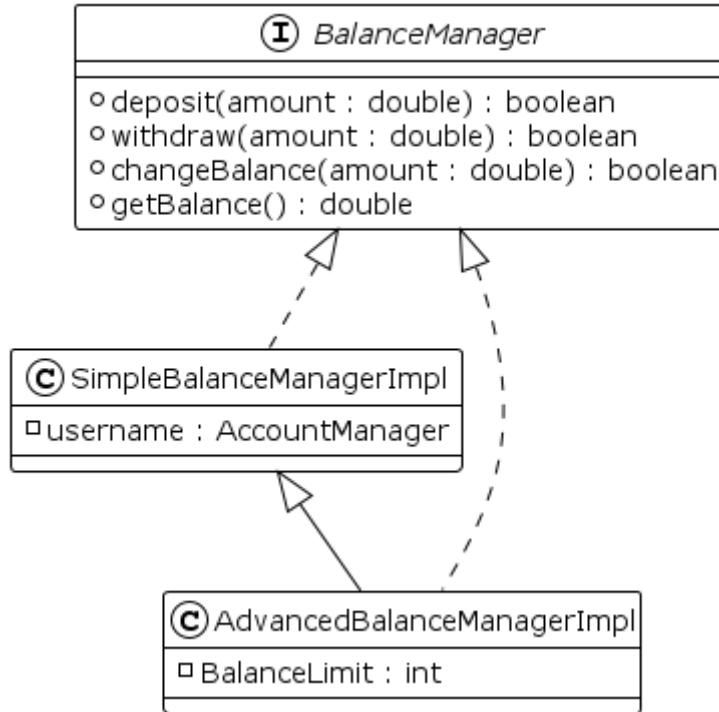
Account Manager



Classe per gestire i dati di ogni utente. Queste classi sono in grado di registrare nuovi utenti, verificare la correttezza dei dati in fase di login, modificare campi come Username e Password a richiesta dell'utente e infine cancellare un account nel caso il proprietario lo richiedesse. La versione Advanced contiene verifiche come scritto precedentemente, alcune di queste verifiche sono:

- Nome utente e password non devono essere più lunghe di 20 caratteri
- L'età deve essere compresa tra 18 e 99 anni.
- Non si può cambiare l'Username con un altro se quel determinato Username è già utilizzato da un altro utente.

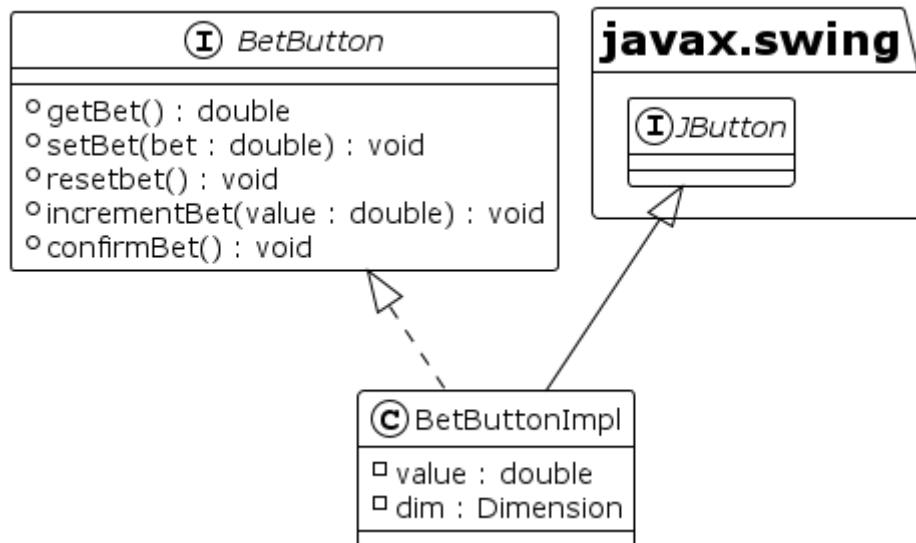
Balance Manager



Classe per gestire il saldo di ogni utente. Queste classi sono in grado di effettuare Depositi e prelievi a richiesta dell'utente o di un qualsiasi gioco. La versione Advanced contiene verifiche come scritto precedentemente, alcune di queste verifiche sono:

- Non si possono depositare o prelevare valori negativi.
- Limite massimo di Saldo (1 Milione).
- Non si possono ritirare una quantità maggiore di soldi rispetto al saldo disponibile.

Pulsante per le scommesse

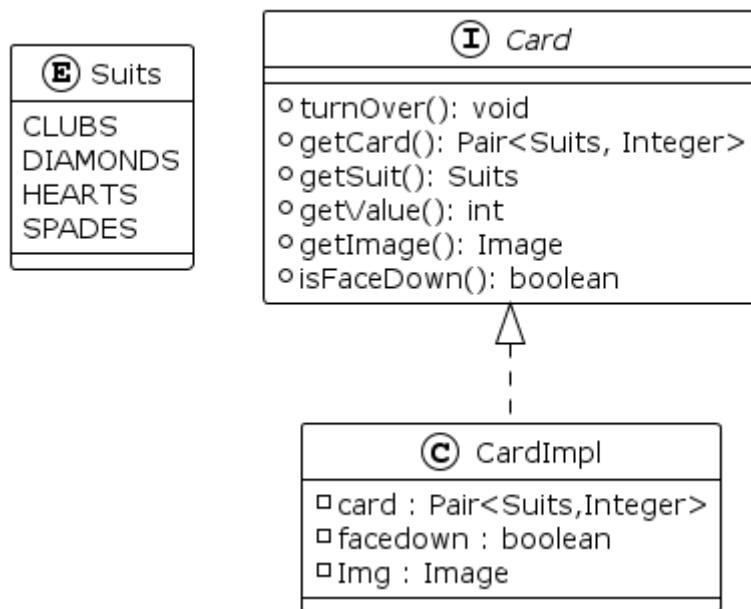


Al fine di ottenere dall'utente il valore dei soldi scommessi ad ogni partita, abbiamo voluto implementare un unico componente che sarà condiviso con ogni gioco d'azzardo, siccome ogni singolo gioco ha necessità di avere almeno un bottone in cui l'utente posizionerà le sue fiches. Ad ogni pressione il pulsante definirà il valore della scommessa in base al valore delle fiches selezionata nell'interfaccia di gioco. Questo componente risulterà invisibile se non è presente nessuna scommessa, altrimenti sarà caratterizzato da un'immagine di una fiches con sopra di essa il valore della puntata stessa, così da simulare il posizionamento di un'effettiva fiches sul tavolo di gioco una volta definita la puntata.

Elementi di gioco

Per la progettazione e lo sviluppo di giochi come BlackJack e Baccarat è stata ritenuta necessaria l'implementazione di classi che modellano oggetti reali quali singole carte e mazzi di carte. Oltre a queste è stata sviluppata anche una classe che modella e gestisce una mano di gioco, utilizzabile sia dal giocatore che dal *dealer*.

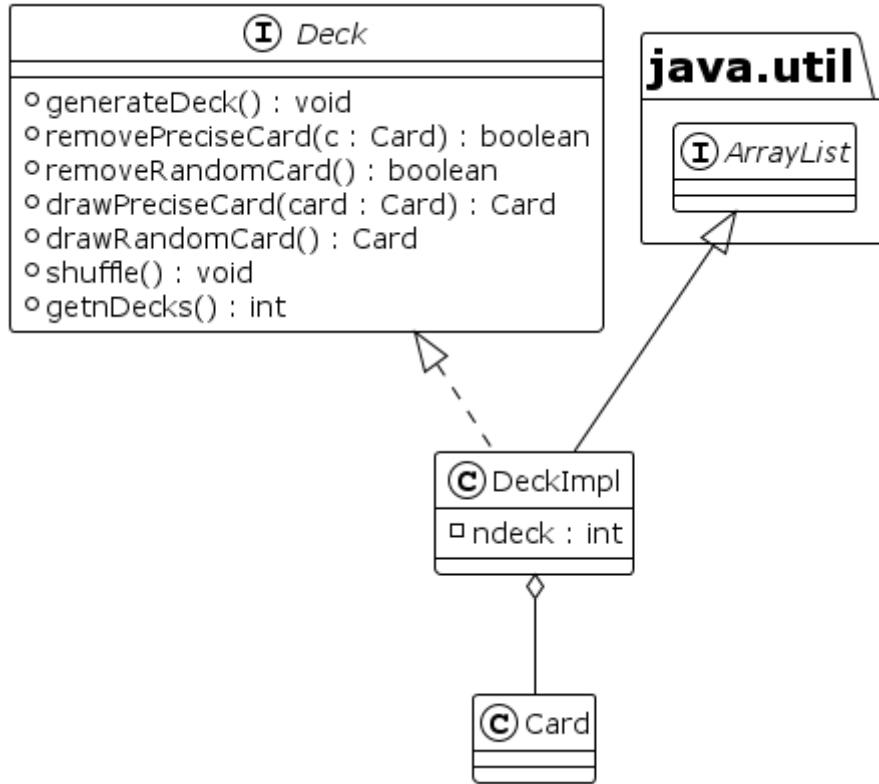
Carte



Per implementare la classe carte ho fatto affidamento alla classe “Pair” vista durante lo svolgimento di questo corso, in quanto ho ritenuto molto utile il suo utilizzo siccome ogni carta può essere vista come un’associazione di seme e valore. Questa classe ha la caratteristica di poter generare carte random o a piacimento specificando seme e/o valore desiderato. Inoltre ogni carta può essere girata a piacimento, richiamando il relativo metodo che andrà a modificare l’immagine di tale carta in base alla sua precedente posizione.

I semi implementati sono 4 e i valori 13, pari a quelli presenti in un mazzo standard da 52 carte. I semi presenti sono: QUADRI, CUORI, PICCHE e FIORI. I valori sono i numeri nell’intervallo 1-13. Nulla toglie a questa implementazione di poter aggiungere, rimuovere o modificare Semi e/o valori differenti da quelli standard, in quanto questa classe è stata progettata per ottimizzare al massimo la riusabilità.

Mazzi



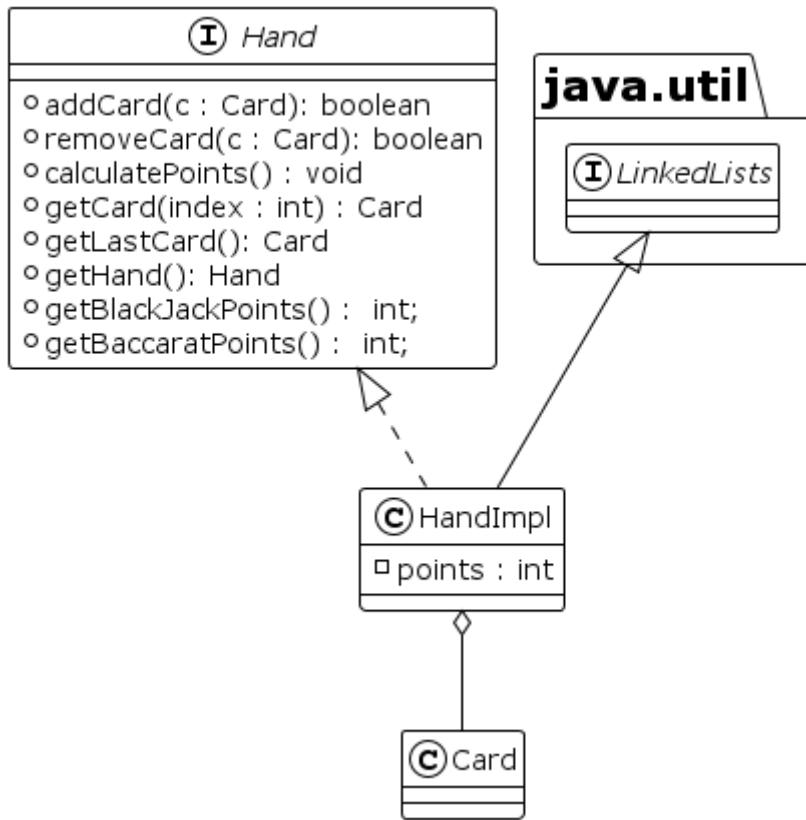
I mazzi non sono altro che liste di carte in cui possono essere effettuate delle specifiche operazioni quali pescaggio di una carta o il conteggio del numero di carte presenti in un determinato mazzo.

Per produrre una struttura il più possibile simile alla realtà, nella generazione di un mazzo si possono definire il numero di mazzi standard che questo andrà a contenere. Questa scelta progettuale è stata utilizzata perchè nella maggior parte dei giochi di casinò, per andare incontro alle tecniche di baro quali il conteggio delle carte da parte dei giocatori, vengono utilizzati una molteplicità di mazzi proprio per annullare le possibilità di utilizzo di queste tecniche illegali.

Il numero di mazzi che viene utilizzato più spesso è pari a 6 mazzi standard da 52 carte l'uno (4 semi per 13 valori).

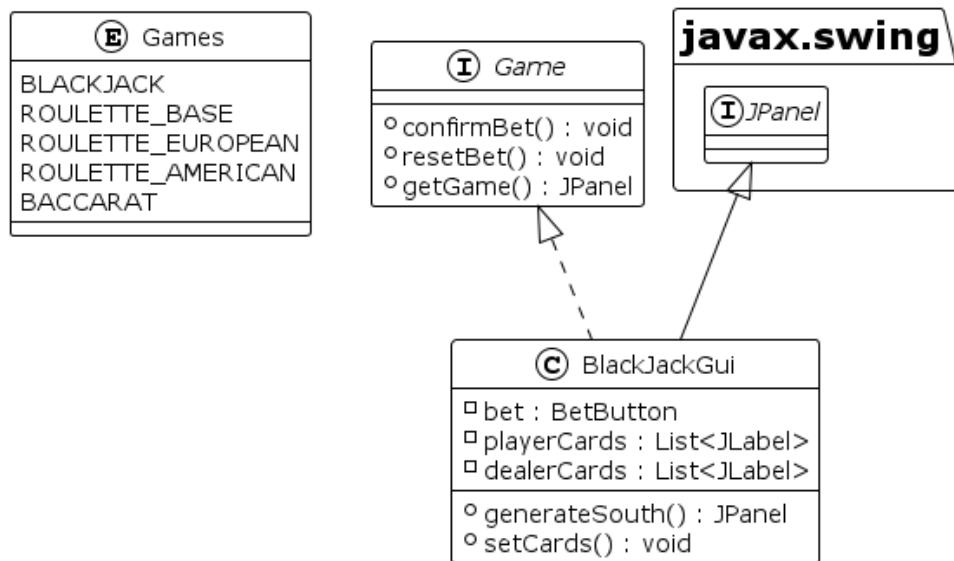
Nel momento in cui il dealer avrà pescato più della metà delle carte presenti nel mazzo originale, prima dell'inizio della prossima partita, l'addetto del casinò avrà il compito di mischiare il mazzo generandone uno nuovo.

Mani



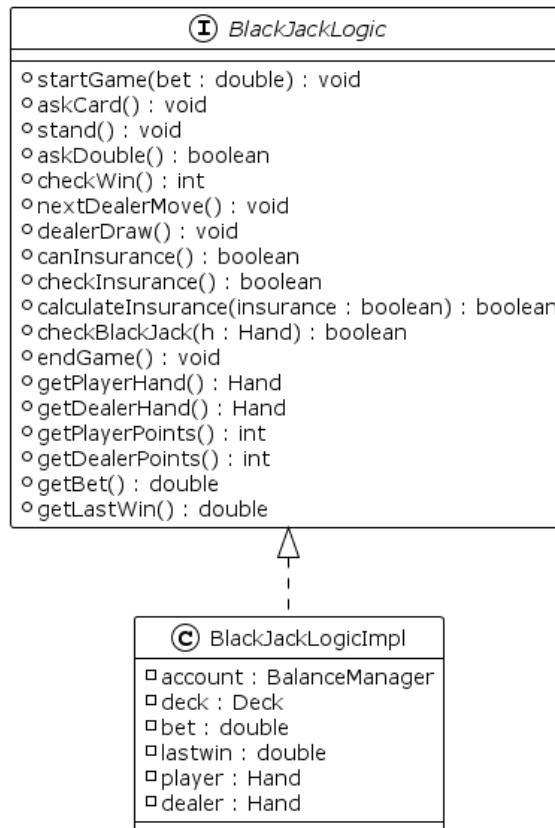
Una Mano è una lista di carte in cui si possono effettuare specifiche operazioni come il calcolo del punteggio. Sia il giocatore che il dealer sono rappresentati da queste mani. Ogni volta che una carta verrà pescata dal mazzo verrà aggiunta alla relativa mano. Alla fine di una partita, per decretare il vincitore e quindi calcolare la vincita o meno di una scommessa, si confronteranno i punteggi delle mani del dealer e del giocatore.

Gioco "BlackJack"



L'interfaccia di gioco del Blackjack è caratterizzata dall'unione di un pannello contenente gli elementi del gioco stesso e di una Gui generale condivisa tra i 3 giochi d'azzardo. La GUI che ho sviluppato contiene i componenti necessari a svolgere tutte le mosse e strategie possibili del gioco "BlackJack" quali la richiesta della carta, il raddoppio, oppure la decisione di non pescare più carte.

Per ogni partita è presente un relativo punteggio che indica i punti di ogni mano (nel caso del dealer con una carta coperta indica solo il punteggio della carta scoperta). Una volta finita la partita, all'utente sarà lasciato il tempo di visualizzare a schermo l'esito della scommessa fino a quando non deciderà di iniziare un'altra utilizzando il tasto azzurro "ricomincia".



Per gestire l'intera logica del gioco “BlackJack” rispettando regole e strategie possibili ho sviluppato una classe di nome `BlackJackLogicImpl` e la sua relativa interfaccia.

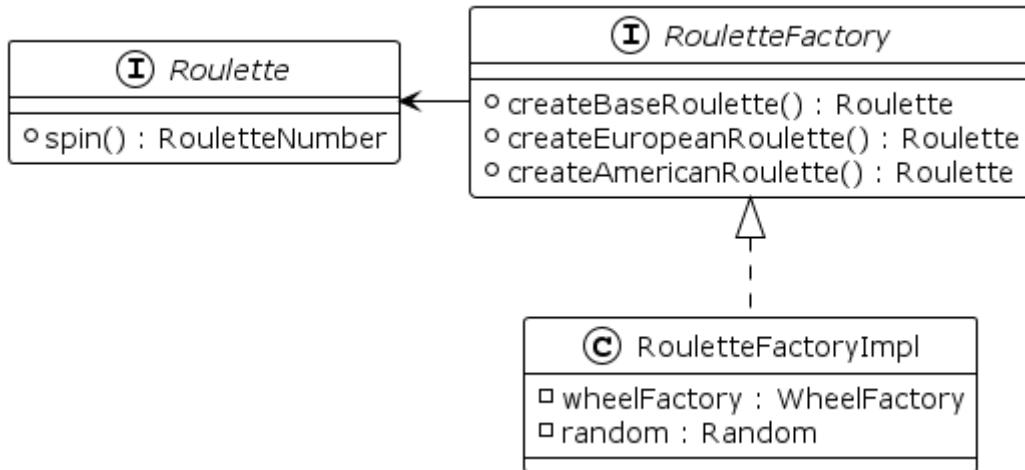
Questa classe contiene tutti i metodi necessari per gestire tutte le mosse sia da parte dell’utente che da parte del Dealer.

All'avvio del gioco verranno distribuite due carte al giocatore e al dealer e spetterà al giocatore iniziare il turno decidendo le proprie mosse.

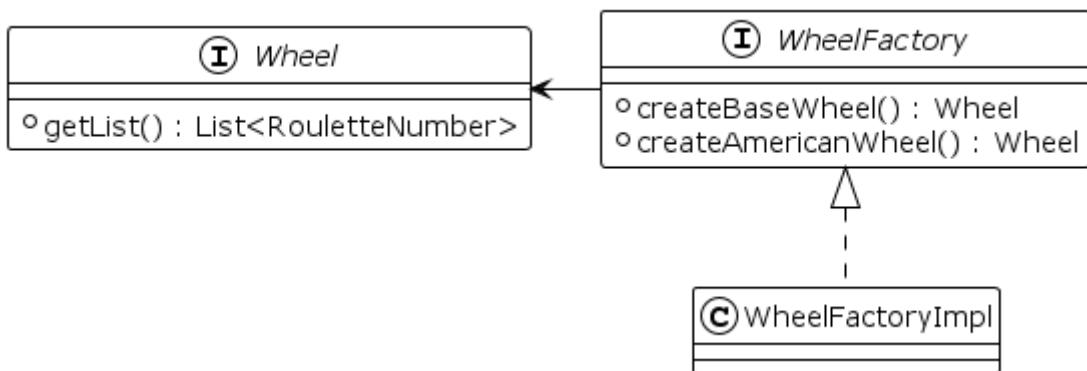
Una volta terminato il turno del giocatore, l'intera logica andrà a calcolare le azioni del dealer con relativi richiami a metodi rispettando le regole del gioco per poi andare a decretare il vincitore della partita.

Federico Muccioli

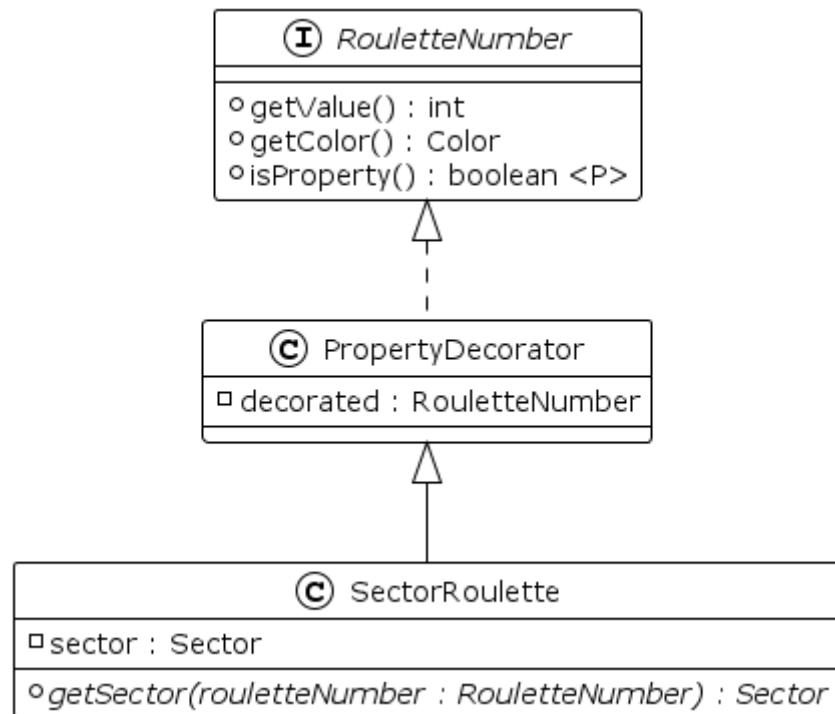
Per gestire la creazione delle roulette ho utilizzato il pattern Abstract Factory definendo l'interfaccia RouletteFactory e implementandola in RouletteFactoryImpl.



Per gestire la creazione delle wheel(collezione che contiene i numeri della roulette) ho sempre utilizzato il pattern Abstract Factory definendo l'interfaccia WheelFactory e implementandola in WheelFactoryImpl.



La European roulette funziona come la Base roulette di differenza ha solo che anzichè ritornare un numero della roulette base aggiunge a quest'ultimo una proprietà aggiuntiva ovvero il settore. Per questo ho utilizzato il pattern Decorator che permette di aggiungere a un qualsiasi RouletteNumber una proprietà aggiuntiva.



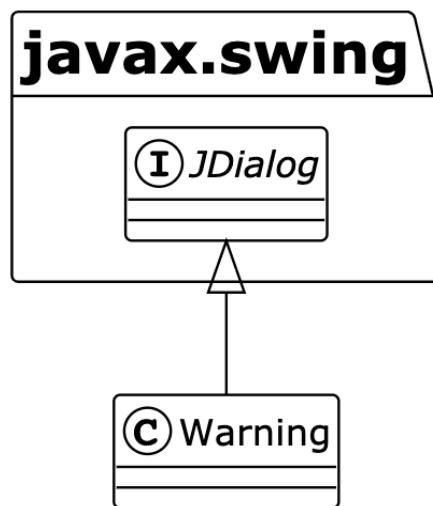
Valeria Verde

Gui Warning all'accesso della piattaforma



Warning gui che avvisa dei pericoli del gioco d'azzardo. La Warning è una JDialog che appare dopo aver effettuato il login nell'applicazione.
E' possibile procedere oltre solo selezionando la checkbox. Dopo averla selezionata il pulsante accept sarà disponibile e si potrà procedere nell'applicazione.

Warning UML



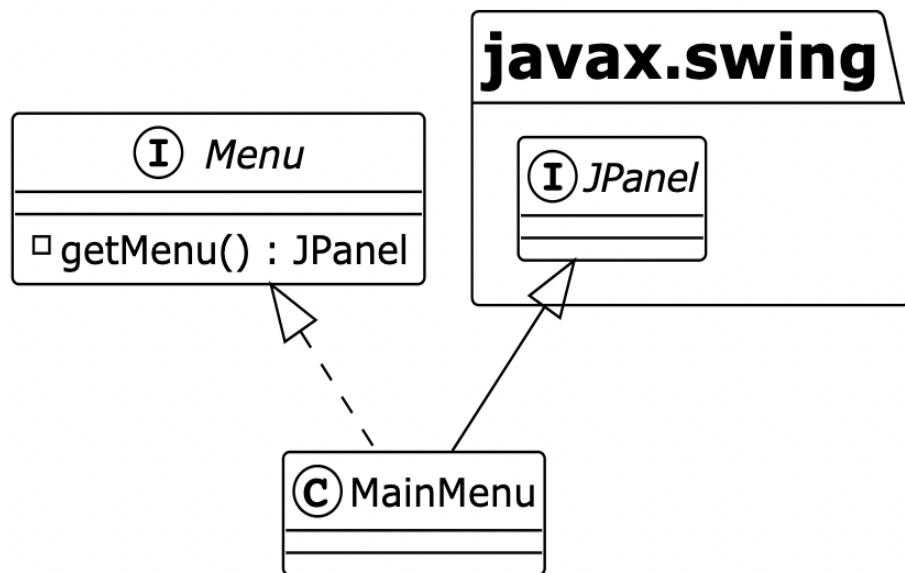
Menu principale



Il menu principale del gioco contiene 3 pulsanti, uno per ogni gioco, che permette di accedere al gioco desiderato.

Troviamo poi in basso il pulsante dove l'utente può visualizzare e gestire le proprie impostazioni dell'account.

Main menu UML

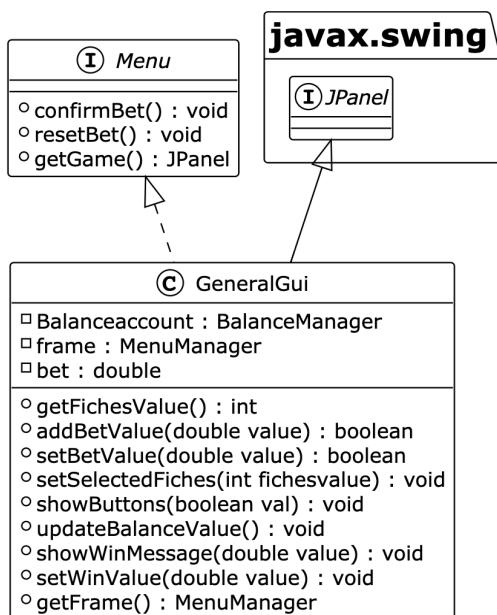


General gui



La General gui viene implementata da tutti e tre i giochi. In alto troviamo il pulsante torna al menu, per permettere al giocatore di tornare indietro in qualsiasi momento e di selezionare un altro gioco. Troviamo poi una Label contenente la vincita del giocatore e infine un pulsante di help che ci rimanderà alla guida specifica del gioco.
Nella parte sud troviamo le label relative alla possibile vincita in base alla puntata del giocatore, la scommessa effettuata dal giocatore e il bilancio attuale in base al deposito effettuato e vincite/perdite. Sulla destra troviamo i pulsanti delle fiches, ognuno con il proprio valore, necessarie per fare una scommessa nel gioco.

General gui UML



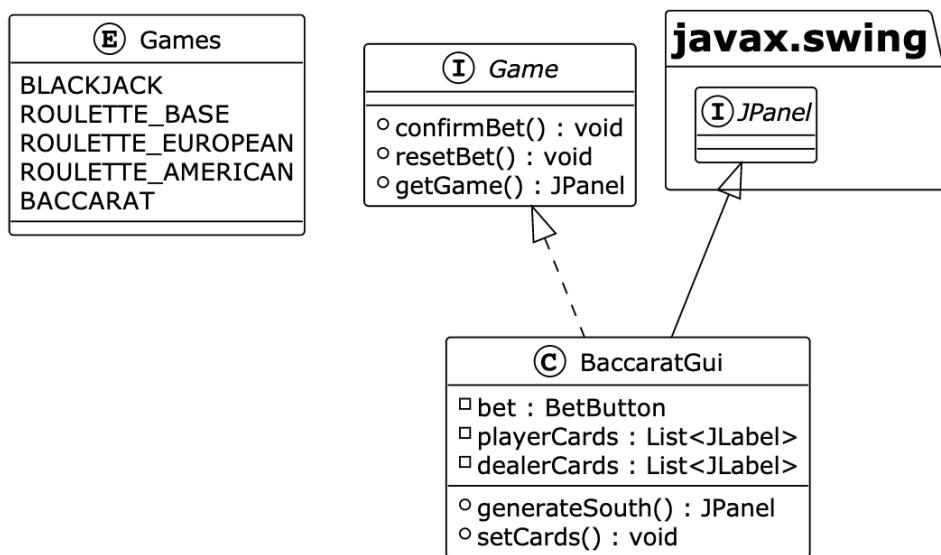
Gioco Baccarat



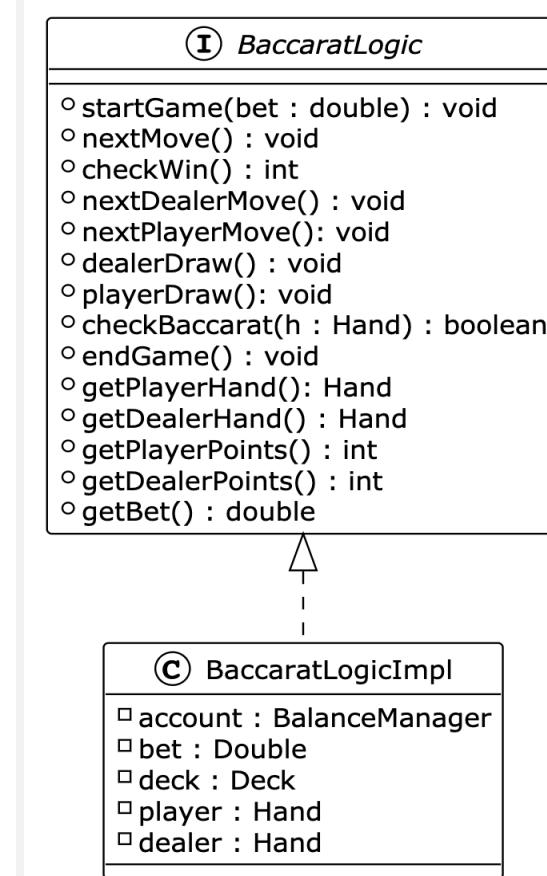
L'interfaccia di gioco del Baccarat è caratterizzata dall'unione di un pannello contenente gli elementi del gioco stesso e dalla Gui generale condivisa tra i 3 giochi d'azzardo. La GUI che ho sviluppato contiene i componenti necessari a svolgere tutte le mosse e strategie del gioco Baccarat.

Per ogni partita è presente un relativo punteggio che indica i punti di ogni mano, del dealer e del player. Una volta finita la partita, all'utente sarà lasciato il tempo di visualizzare a schermo l'esito della scommessa fino a quando non deciderà di iniziare un'altra utilizzando il tasto azzurro "ricomincia".

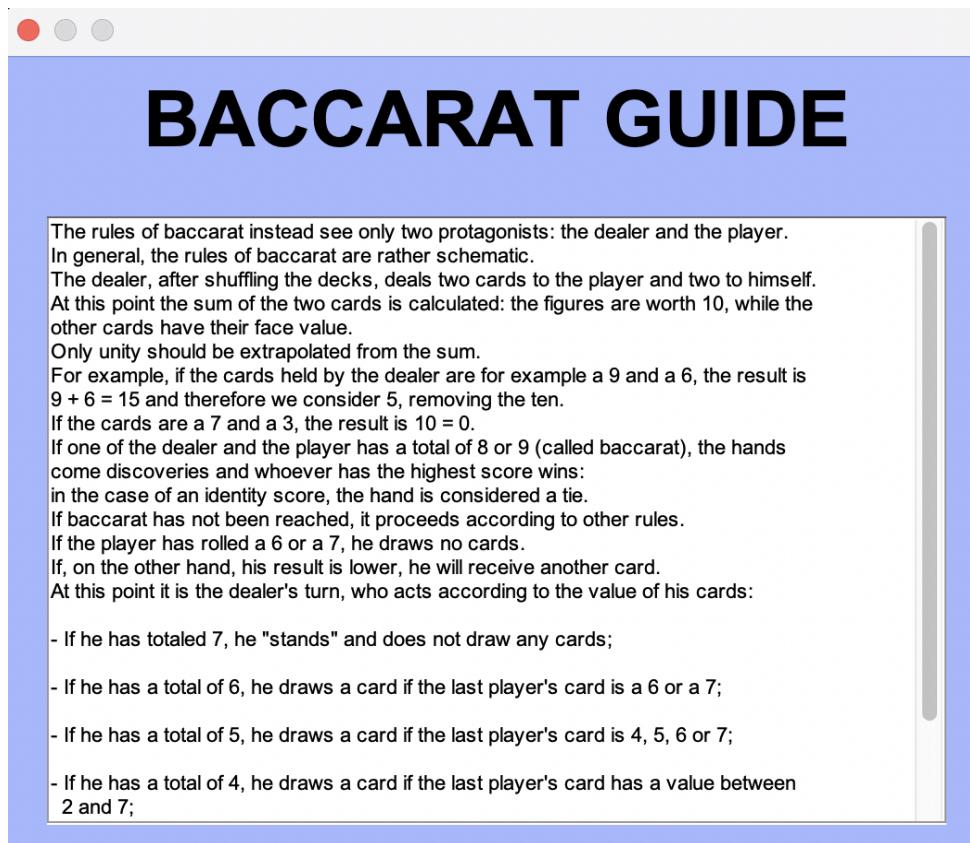
Baccarat gui UML



Baccarat logic UML



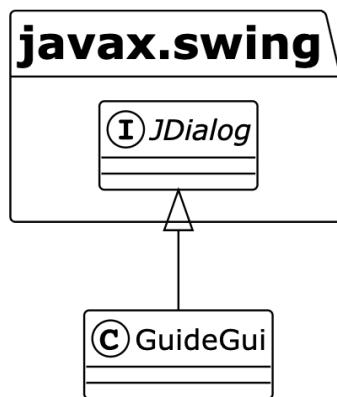
Guide gui



La guide gui, raggiungibile tramite l'help button della general gui, indica le regole per il corretto utilizzo del gioco scelto.

La gui è comune a tutti e tre i giochi, ognuno contiene un file di testo con le relative regole.

GuideGui UML



Capitolo 3

Sviluppo

3.1 Testing Automatizzato

Durante lo sviluppo di questo software sono stati creati dei test affinché tutte le funzionalità necessarie potessero essere garantite senza alcun tipo di errore. Il testing è completamente automatizzato ed è stato realizzato tramite il framework JUnit 5. Gli unici test manuali che sono stati effettuati sono stati la verifica del corretto funzionamento di questo programma all'interno di differenti Sistemi Operativi.

Segue un elenco specifico dei test automatici sviluppati da ogni elemento del gruppo:

3.1.1 Nicola Montanari

- *AccountManager Test*
 - *SimpleAccountManagerTest*
 - *AdvancedAccountManagerTest*

Verifiche riguardanti la gestione degli account come può per esempio essere la registrazione di un nuovo utente o il cambio della password di uno già esistente. La versione semplice definisce le funzionalità di base, mentre quella avanzata comprende verifiche precedenti all'effettiva funzionalità richiesta.

- *BalanceManager Test*
 - *SimpleBalanceManagerTest*
 - *AdvancedBalanceManagerTest*

Verifiche riguardanti la gestione del saldo di ogni account come può per esempio essere il prelievo di una certa quantità di denaro oppure il deposito in caso di vittoria in uno dei 3 giochi. La versione semplice definisce le funzionalità di base, mentre quella avanzata comprende verifiche precedenti all'effettiva funzionalità richiesta.

- *BlackJackLogicTest*

Test della logica del gioco ***BlackJack***

- *CardTest*
- *DeckTest*
- *HandTest*

Test dei componenti necessari al funzionamento del gioco ***BlackJack***, ovvero le singole carte, i vari mazzi e le relative mani per giocatore e per il *Dealer*.

- *ResizerTest*

Test di una classe di utility che calcola le dimensioni in altezza e larghezza della finestra prendendo la risoluzione dello schermo per poi scalarla con un fattore precedentemente definito.

3.1.2 Federico Muccioli

- *Roulette Test*
 - *BaseRouletteTest*
 - *AmericanRouletteTest*
 - *EuropeanRouletteTest*

Per ciascuna roulette viene testato l'intervallo dei numeri che vengono emessi e che la vincita sulle relative proprietà siano corrette.

3.1.3 Valeria verde

- *Test logica del gioco baccarat*
 - *BaccaratLogicTest*

Classe di test per verificare il corretto funzionamento della logica del gioco baccarat:

- Viene testato il corretto funzionamento delle mosse di player e dealer
- Vengono testate prossime mosse di dealer e player che possono al più pescare una carta
- Viene testato il “baccarat”
- Vengono testati i punteggi di dealer e player, che al massimo possono raggiungere il punteggio 9

3.2 Metodologia di lavoro

Uno degli obiettivi principali dello svolgimento di questo progetto era quello di organizzarlo separando il più possibile i vari compiti tra di loro, affinché tutti i componenti del gruppo potessero lavorare in modo parallelo senza incorrere in problemi di conflitto di git e per evitare che qualcuno si potesse bloccare in attesa che un altro componente finisca uno dei suoi compiti assegnati.

L'utilizzo di Git è stato fondamentale per una programmazione efficiente. Abbiamo utilizzato un branch “master” come branch principale, il branch “develop” per effettuare la maggior parte delle modifiche condivise (esempio Interfaccia di accesso) e abbiamo poi diviso il resto del lavoro su branch diversi, uno per ogni gioco da sviluppare, così da rispettare l'obiettivo predisposto all'inizio.

3.2.1 Nicola Montanari

- Interfaccia grafica del sistema di accesso al proprio account con credenziali personali. Possibilità di effettuare l'accesso nel caso si abbiano già delle credenziali, oppure di effettuare la registrazione.
- Gestione degli Account tramite Manager utilizzando scrittura di file JSON.
 - Gestione di Account, quindi registrazione (creazione account), accesso (verifica dati inseriti con dati presenti), cambio password, cambio username, eliminazione definitiva di un determinato account. È presente una versione semplice e una versione avanzata, quest'ultima presenta le verifiche necessarie affinché il programma esegui senza alcun tipo di problema.
 - Gestione del saldo di un determinato Account. Quindi possibilità di effettuare depositi e ritiri a piacimento oltre che visualizzare il proprio saldo disponibile all'interno della piattaforma. Questo manager viene totalmente implementato in ciascuno dei giochi da casinò tramite una GUI generale affinché questi ultimi possano modificare il saldo di un giocatore in base all'avanzamento del gioco. È presente una versione semplice e una versione avanzata, quest'ultima presenta le verifiche necessarie affinché il programma esegui senza alcun tipo di problema.
- Pulsante specifico per le puntate. Questo componente ha la caratteristica di essere condiviso con tutti e tre i giochi. Serve a definire la quantità della puntata che il giocatore andrà a scommettere all'inizio di ogni giocata.

- Classe di Utilità contenenti i metodi necessari a definire la grandezza della finestra del programma in base allo schermo dell’utente e per caricare correttamente le varie immagini e file di testo necessari alle varie GUI.
- Gioco **Blackjack** e relativi componenti necessari al funzionamento di quest’ultimo:
 - Classe **Carta** che modella tutte e 52 le carte di un mazzo standard (senza jolly).
 - Classe **Mazzo** che modella un insieme di carte di dimensione variabile. (Più comunemente vengono utilizzati 6 mazzi da 52 carte)
 - Classe **Mano** modella un insieme di carte con funzionalità precise quali il calcolo dei punti di tale mano.
 - Finestra di avviso in caso di **Assicurazione** contro un possibile **BlackJack** del *Dealer*.

3.2.2 Federico Muccioli

- Gioco **Roulette** che poi si differenzia in BaseRoulette, AmericanRoulette e EuropeanRoulette. Sia la parte grafica che quella implementativa. Per implementare le puntate ho dovuto estendere il bottone creato da Montanari creando il RouletteButton.
- La gestione dei vari menu, quindi della creazione dell’interfaccia Menu Manager, l’implementazione MainGui e l’interfaccia Menu.
- La creazione del Menu Account e dei relativi pannelli in view.menu.account

3.2.3 Valeria Verde

- **Warning**
JDialog warning gui che appare all’accesso dell’applicativo.
E possibile procedere oltre solo accettando “termini e condizioni” tramite la checkbox presente nella gui.
- **Menu principale**
Interfaccia grafica del sistema di accesso ai vari giochi presenti nell’applicativo, ciascuno con proprio pulsante di accesso. Dalla schermata è poi possibile accedere alle impostazioni del proprio account.
- **General gui**
Gui generale implementata da tutti e 3 giochi dell’applicativo. Indica il valore della vincita in base alla scommessa, valore della scommessa con eventuale vincita

calcolata, bilancio totale del giocatore in base ai soldi depositati e vincite/perdite, pulsante help, infine pulsanti contenenti le fiches necessarie per effettuare la scommessa in base al proprio deposito rimanente.

- **Guide gui**

Gui generica utilizzata nella general gui. Viene utilizzata in tutti e 3 i giochi e comprende le regole per il corretto utilizzo dei giochi. E' possibile visualizzarla tramite pulsante help presente nella general gui.

- **Gioco baccarat**

- **Baccarat gui**

Gui che gestisce l'interfaccia grafica del gioco baccarat, che riutilizza la classe betButton (pulsante per le scommesse) elencato in precedenza.

- **Logica gioco baccarat**

Logica del gioco baccarat che implementa il funzionamento corretto del gioco. Vengono riutilizzate le classi card, hand, e deck elencate precedentemente, seguendo le regole del baccarat.

3.3 Note di sviluppo

3.3.1 Nicola Montanari

- Lambda Expressions
- JSON-Simple-1.1.1
 - Un semplice *toolkit* per JSON, utile per scrivere e leggere file JSON.
 - Fornisce molteplici funzionalità, come può essere la scrittura di un file JSON direttamente da un oggetto JSONObject oppure la modifica del valore di un determinato campo di un file.
 - Molto flessibile e semplice da usare in quanto utilizza interfacce come Map e List.

3.3.2 Federico Muccioli

- Lambda Expressions
- Generici, la classe Pair<X,Y> è creata un metodo generico.

3.3.3 Valeria Verde

- Lambda Expressions (Action Listener)

Abbiamo poi utilizzato la classe Pair, vista durante il corso, siccome ci tornava molto utile soprattutto nell'implementazione delle carte da gioco, in quanto queste ultime sono un'associazione di Seme e Valore numerico. Questa classe viene utilizzata anche all'interno del gioco "Roulette" per associare la proprietà scommessa con l'importo scommesso dall'utente.

Capitolo 4

Commenti finali

4.1 Autovalutazione e lavori futuri

Nicola Montanari

Questa è stata la mia prima esperienza in cui ho partecipato alla progettazione e allo sviluppo di un software di medie dimensioni partendo completamente da zero. Questo progetto è anche il primo software che sviluppo all'interno di un gruppo di più persone, ciò mi ha consentito di migliorare il mio approccio di lavoro progettando e organizzando varie porzioni di codice con i vari elementi del gruppo, potendo così effettuare molteplici progressi in diverse parti del progetto e ottimizzando i propri risultati simultaneamente.

Sviluppando questo applicativo ho inoltre avuto la possibilità di approfondire l'utilizzo di git, uno strumento di cui solo ora mi rendo conto dell'elevatissima utilità e praticità che offre agli sviluppatori come noi che hanno la necessità di sviluppare codice condiviso nello stesso momento.

Sono molto grato di aver imparato ad usare questo *tool* così da poterlo utilizzare in miei progetti futuri, sia personali che Universitari, così da aumentarne la qualità e migliorare il processo di sviluppo.

Per quanto riguarda il lavoro svolto da me nello sviluppo di questo applicativo mi ritengo molto soddisfatto siccome penso di aver generato un buon risultato sia in termini di progettazione e scrittura di codice che in termini di cooperazione con i propri compagni di gruppo.

Grazie a questa esperienza mi rendo conto di essere cresciuto molto in termini di conoscenze di java e di programmazione ad oggetti nel generale, non pensavo che lo sviluppo di questo software mi potesse portare a raccogliere una così elevata quantità di benefici.

L'unico grande mio difetto e/o problema che ho incontrato è l'aver sottostimato in alcune occasioni il tempo necessario di sviluppo di determinate parti di codice.

Federico Muccioli

Prima di questo progetto non avevo mai dovuto lavorare in gruppo e tutte le applicazioni che mi ero fin'ora trovato ad implementare, avevano un

impegno di massimo una settimana. Per questi motivi il progetto ha rappresentato una sfida notevole al termine della quale posso dirmi veramente soddisfatto in quello che mi ha insegnato. Ho avuto modo di approcciare aspetti quali la collaborazione con altre persone e l'utilizzo di tool di supporto come Git. Finisco questo progetto con più consapevolezza riguardo alla progettazione di software e all'importanza della parte di analisi, di design.

Valeria Verde

Mi era già capitato in passato di lavorare a progetti di gruppo, ma è la prima volta che questo comprende l'implementazione completa di un applicativo abbastanza impegnativo. Sono abituata a lavorare da sola, quindi i problemi relativi alla gestione del lavoro in comune non sono mancati, ma è stata sicuramente una bella esperienza formativa che mi permetterà sicuramente di avere basi già solide per i prossimi lavori di gruppo in comune.

Durante lo sviluppo dell'applicativo ho anche avuto modo di aumentare le mie conoscenze del linguaggio java, che sicuramente mi sarà utile anche in futuro.

Abbiamo utilizzato molto anche github, che ci ha permesso di tenere traccia dei progressi fatti da ognuno in tempo reale, che all'inizio consideravo poco intuitivo, ma alla fine sono riuscita abbastanza a utilizzarlo correttamente. Sicuramente tornerà utile il suo utilizzo anche per progetti futuri.

Appendice A

Guida Utente

All'avvio del programma è necessario inserire le proprie credenziali per effettuare l'accesso alla piattaforma, nel caso non si dispone di tali dati è possibile registrarsi dopo aver premuto l'apposito tasto presente sempre nella schermata di avvio.

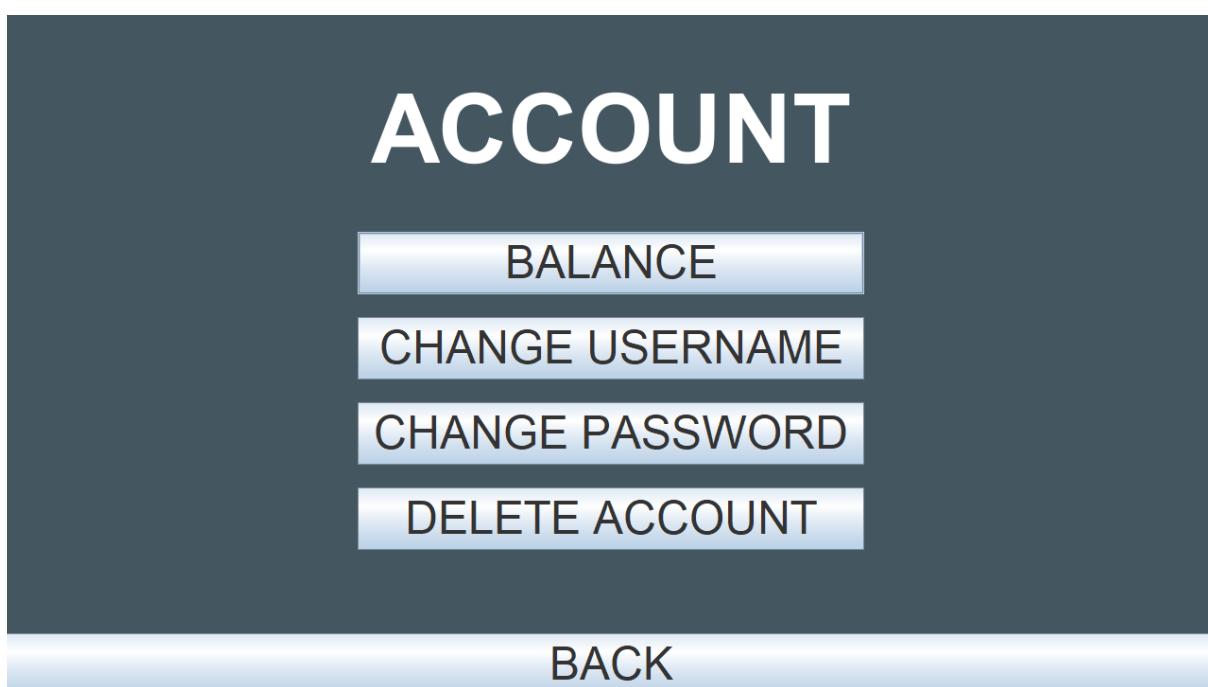
Non è possibile registrarsi utilizzando Username e Password più lunghi di 20 caratteri e sono abilitati alla registrazione solo le persone con età compresa tra i 18 e i 99 anni.



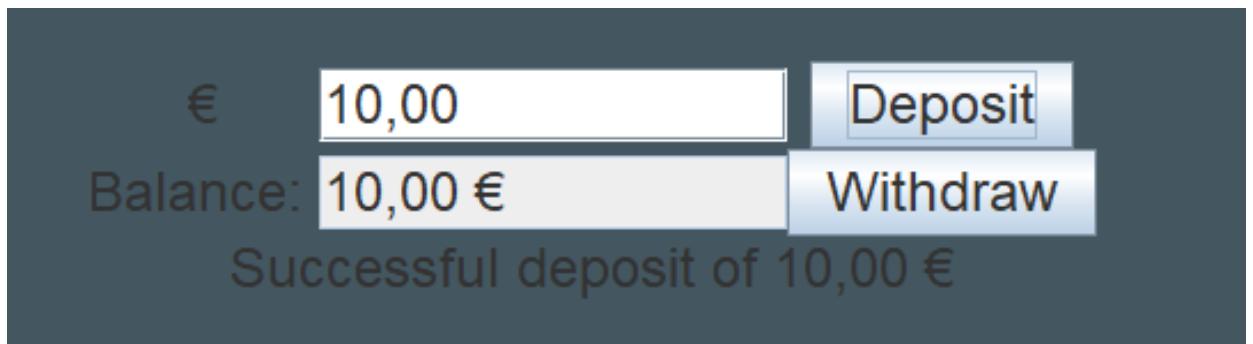
Dopo essersi registrati ed effettuato il login al proprio account si arriva al Menù principale contenente i tre differenti giochi da casinò.



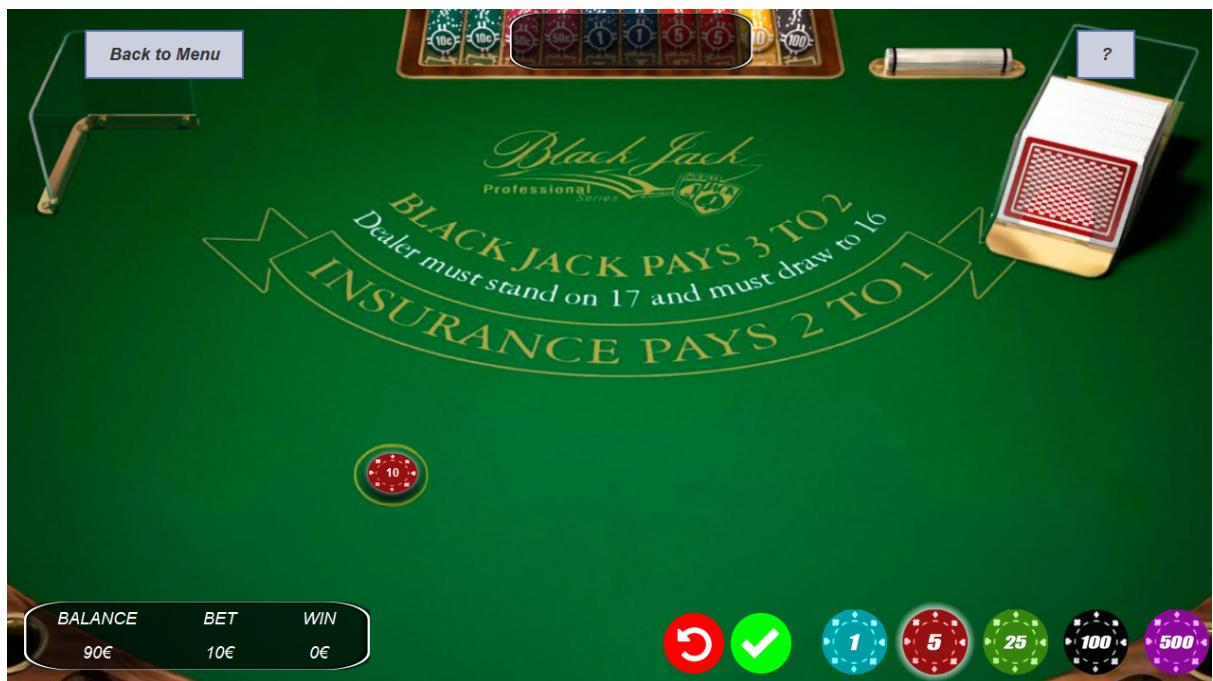
Sin da subito è possibile selezionare i giochi, ma non sarà possibile giocarci siccome alla registrazione il proprio saldo sarà pari a 0, quindi consigliamo di effettuare una ricarica tramite la sezione “BALANCE” nelle impostazioni dell’account.



In questa sezione basta inserire la quantità di soldi da trasferire e definire quale operazione eseguire, in questo caso se si vuole depositare 10 euro, basta inserire il valore 10 e cliccare il tasto "Deposit". Un messaggio verrà visualizzato contenente l'esito del trasferimento richiesto.



Ora è possibile giocare a uno dei tre giochi disponibili.
Per iniziare una partita, si selezioni il gioco a cui si vuole giocare e si punti la quantità di soldi che si vogliono scommettere.
Per farlo basta premere nello spazio prestabilito alle puntate. Ogni volta che quel tasto viene premuto, la quantità di soldi scommessi aumenta in base al valore della fiches selezionata in basso a destra dello schermo, di default la fiches selezionata ha valore 1 euro.



Una volta decisa la quantità, si prema il tasto verde di conferma per iniziare a giocare la partita.



Per conoscere tutte le regole dei vari giochi e le relative quote di vincita è possibile premere il tasto “?” in alto a destra della schermata.

