

Rubamazzo [C++]

Analisi Tecnica

Indice

<u>1</u>	<u>PREMESSA</u>	3
<u>2</u>	<u>OBIETTIVO</u>	3
<u>3</u>	<u>DOCUMENTI DI RIFERIMENTO</u>	3 - 4
<u>4</u>	<u>TERMINOLOGIA E ACRONIMI</u>	4
<u>6</u>	<u>DIAGRAMMA DELLE CLASSI (UML)</u>	5
<u>7</u>	<u>ANALISI ARCHITETTURALE</u>	5 - 6
	<u>7.1</u> <u>ATTRIBUTI CLASSE RUBAMAZZO</u>	7
	<u>7.1</u> <u>METODI CLASSE RUBAMAZZO</u>	8 - 9
<u>8</u>	<u>DIAGRAMMA DI GANTT</u>	10
<u>9</u>	<u>REGOLAMENTO</u>	10

PREMESSA

Il presente documento illustra le specifiche tecniche relative al sistema di simulazione del gioco di Rubamazzo. Rubamazzo è un gioco di carte popolare molto diffuso in Italia. Per far sì che si possa comprendere a pieno il funzionamento del software proposto, è riportata di seguito una dettagliata analisi tecnica grazie alla quale sarà possibile comprendere nel dettaglio la fase di progettazione e sviluppo del software

OBIETTIVO

Obiettivo di questo documento è descrivere dettagliatamente i componenti necessari alla realizzazione del sistema di simulazione del gioco di carte Rubamazzo. Inoltre, il sw proposto, permette anche la visualizzazione del regolamento del gioco di carte Rubamazzo, andando a chiarire eventuali dubbi riguardanti il regolamento del gioco.

DOCUMENTI DI RIFERIMENTO

Per la realizzazione del sistema di simulazione del gioco di carte Rubamazzo si è fatto riferimento a diversi documenti utilizzati per documentarsi sul linguaggio di programmazione utilizzato e sulla realizzazione della gerarchia delle classi.

Di seguito sono elencati i documenti ai quali si è fatto riferimento:

- Object Oriented Programming - Linguaggio C++ (cap 1-21)
a cura delle proff. Nunziata Chiara, Ferrari Fiorenza.
- Object Oriented Programming - Appendice A - UML
a cura delle proff. Nunziata Chiara, Ferrari Fiorenza.

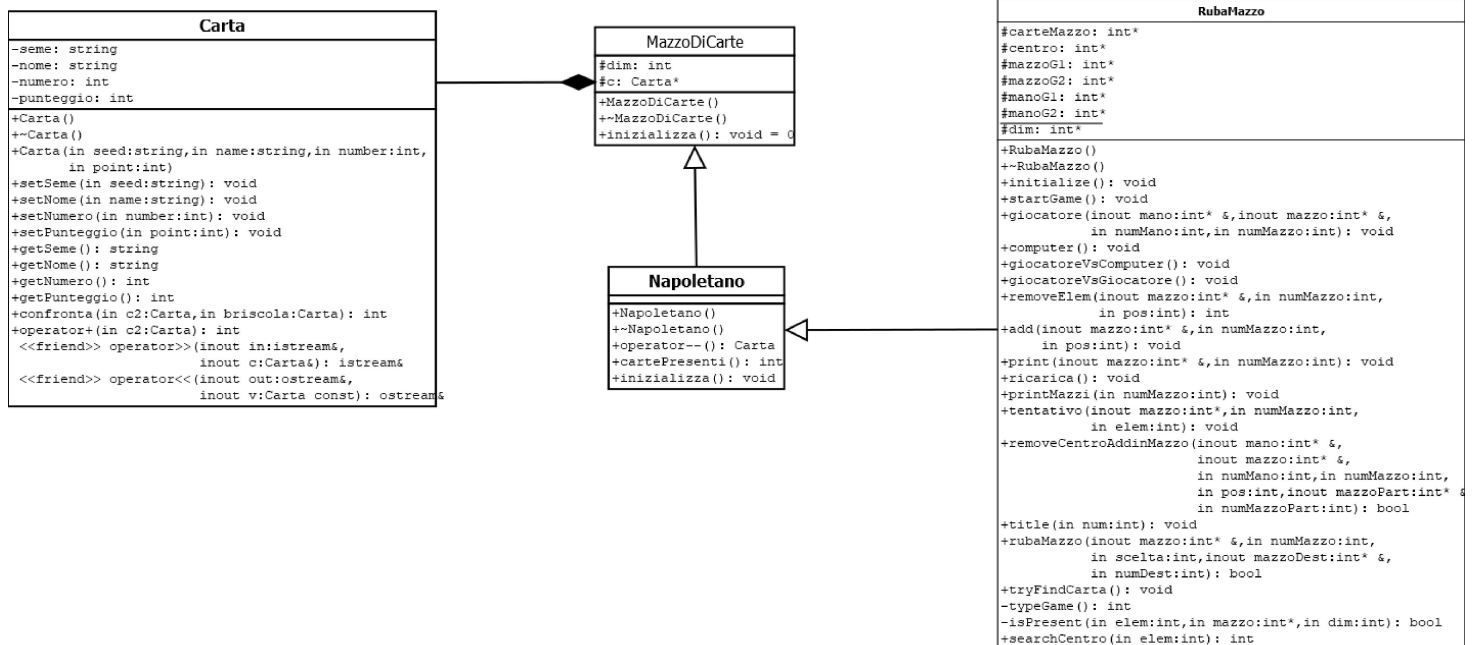
TERMINOLOGIA E ACRONIMI

Di seguito è riportata una tabella nella quale sono presenti acronimi ed abbreviazioni utilizzate all'interno della documentazione:

<i>Abbreviazione / Acronimo</i>	<i>Definizione</i>
UML	Unified Modeling Language: definisce un insieme di concetti per la descrizione e la progettazione di sistemi software di qualsiasi tipo,
SW	Software
CAP	Capitolo
AV	Ambito di Visibilità

DIAGRAMMA DELLE CLASSI

In questo paragrafo si riporta la descrizione delle classi necessarie per la gestione dell'applicazione nel suo complesso:



[Link uml](#)

ANALISI ARCHITETTURALE

Prima di descrivere nel dettaglio la classe Rubamazzo, si riporta una breve descrizione delle classi Carta, MazzoDiCarte, Napoletano:

- La classe **Carta** ha come attributi privati due stringhe che rappresentano il seme della carta e il nome, un numero intero che identifica la carta e il punteggio intero associato alla carta.

- La classe **MazzoDiCarte** è astratta. Essa contiene due attributi: un array dinamico di carte e la dimensione effettiva dell'array. Ha un costruttore di default, un distruttore e un metodo che inizializza un generico mazzo di carte.
- La classe **Napoletano** è una classe che rappresenta un mazzo di carte napoletane, derivata dalla classe base MazzoDiCarte. Questa classe eredita tutte le caratteristiche e le funzionalità di un mazzo di carte generico e ne aggiunge alcune specifiche per gestire le carte napoletane, come per esempio l'implementazione del metodo *inizializza()* che ha il compito di creare un mazzo di 40 carte e inizializzarlo con le carte napoletane. Nelle carte napoletane i semi sono 4: spade, bastoni, denari, coppe. Per ogni seme ci sono 10 carte (numerate da 1 a 10) e ogni carta ha un nome e un punteggio. La carta numero 1 si chiama Asso, la numero 8 si chiama Fante, la numero nove è il Cavallo, la numero 10 è il Re, per le altre carte il nome coincide con il numero. I punteggi sono così distribuiti: Asso = 11 punti, Fante = 2 punti, Cavallo = 3 punti, Re = 4 punti, Tre = 10 punti, tutte le altre carte valgono 0 punti.

Di seguito è riportato una descrizione dettagliata della classe **RubaMazzo**:

La classe RubaMazzo è una classe che rappresenta un gioco basato su un mazzo di carte napoletane ed estende le funzionalità della classe Napoletano. Quella che segue è una descrizione degli attributi e dei metodi della classe:

- **Attributi:**

*Si è pensato di non realizzare degli array contenenti delle carte vere e proprie ma si è optato per una soluzione differente, ossia di realizzare degli array che al posto di essere di tipo **Carta** sono array di interi, i quali elementi stanno a rappresentare l'indice della carta presente all'interno dell'array di Carte **c** presente nella classe **MazzoDiCarte**.*

- *int ***carteMazzo**: mazzo integrale che all'inizio del gioco ha 40 carte*
- *int ***centro**: insieme di carte presenti sul tavolo*
- *int ***mazzoG1**: insieme di carte raccolte nel corso del gioco dal giocatore 1*
- *int ***mazzoG2**: insieme di carte raccolte nel corso del gioco dal giocatore 2*
- *int ***manoG1**: insieme di carte presenti nella mano del giocatore 1*
- *int ***manoG2**: Giocatore o computer - insieme di carte presenti nella mano del giocatore 2 o del computer*

*L'array descritto di seguito, invece, è una struttura dati abilitata a contenere la dimensione dei singoli array dichiarati nella classe **Rubamazzo**.*

- *int ***dim***
 - *dim[0] = dimensione ***carteMazzo***
 - *dim[1] = dimensione ***centro***
 - *dim[2] = dimensione ***mazzoG1***
 - *dim[3] = dimensione ***mazzoG2***
 - *dim[4] = dimensione ***manoG1***
 - *dim[5] = dimensione ***manoG2***

- **Metodi con AV pubblico**

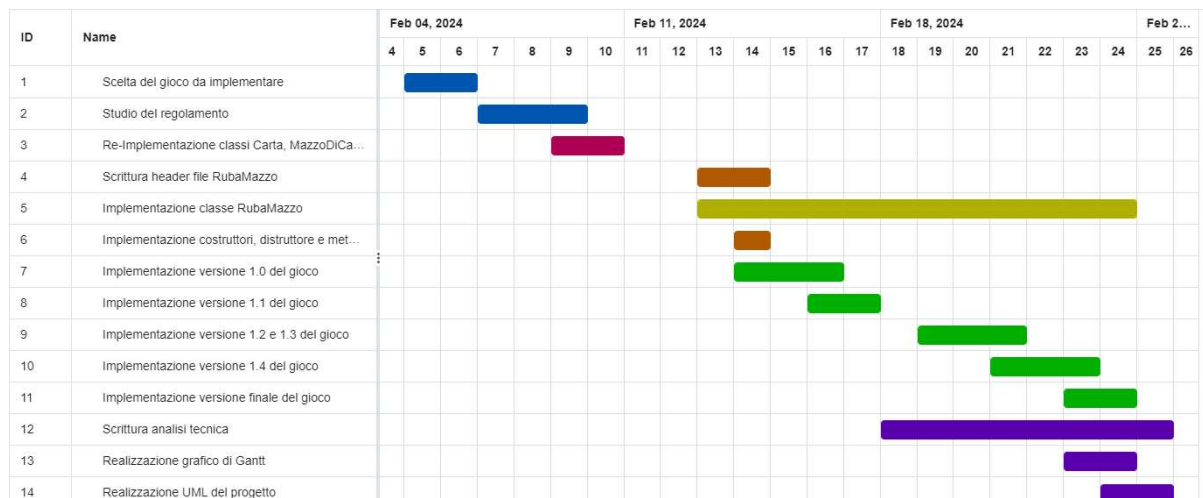
- **void initialize():** metodo abilitato all'inizializzazione del mazzo di 40 carte (metodo che simula il mescolamento del mazzo);
- **void startGame():** metodo utilizzato per iniziare una partita;
- **void giocatore(int* &mano, int* &mazzo, int numMano, int numMazzo):** metodo che fornisce l'interfaccia che permette ad un giocatore di interagire con il sistema (per esempio selezionare quale carta mettere al centro del tavolo);
- **void computer():** metodo utilizzato per eseguire il turno di gioco del computer;
- **void giocatoreVsComputer():** metodo che permette di iniziare e svolgere una partita Giocatore vs Computer;
- **void giocatoreVsGiocatore():** metodo che permette di iniziare e svolgere una partita Giocatore vs Giocatore;
- **int removeElem(int* &mazzo, int numMazzo, int pos):** metodo utilizzato per rimuovere una carta dal mazzo;
- **void add(int* &mazzo, int numMazzo, int elem):** metodo utilizzato per aggiungere una carta ad un mazzo;
- **void print(int* &mazzo, int numMazzo):** metodo utilizzato per stampare le carte di un mazzo;
- **void ricarica():** metodo che simula la ridistribuzione di tre carte per giocatore quando entrambi hanno terminato le carte presenti nella loro mano;
- **void printMazzi(int numMazzo):** metodo utilizzato per stampare i mazzi numero 1, 2, 3;
- **bool removeCentroAddinMazzo(int* &mano, int* &mazzo, int numMano, int numMazzo, int pos, int* &mazzoPart, int numMazzoPart):** metodo utilizzato per rimuovere una carta dal mazzo 1 (ossia le carte presenti al centro) ed aggiungerlo nel mazzo personale di un giocatore;

- *void **title**(int num)*: metodo abilitato alla stampa dell'ASCII ART che raffigura il nome del gioco;
- *bool **rubaMazzo**(int* &mazzo, int numMazzo, int scelta, int* &mazzoDest, int numDest)*: metodo che permette ad un giocatore di rubare il mazzo avversario quando c'è la possibilità di effettuare tale azione;
- *void **tryFindCarta**()*: metodo utilizzato quando si è deciso di giocare alla modalità GIOCATORE vs COMPUTER, tale metodo permette al computer di andare a confrontare tutte le carte della sua mano con le carte presenti nel tavolo, per cercare una possibile carta da prendere ed aggiungere al suo mazzo, se invece non c'è alcuna carta che si può prendere andrà a mettere una sua carta della mano nel centro

- **Metodi con AV privato**

- *int **typeGame**()*: metodo che permette di andare a selezionare quale modalità si vuole giocare;
- *bool **isPresent**(int elem, int* mazzo, int dim)*: metodo che restituisce true se la carta elem è presente in un mazzo altrimenti false;
- *int **searchCentro**(int elem)*: metodo di ricerca lineare che restituisce -1 se la carta non è stata trovata nel centro, altrimenti restituisce la posizione della carta elem nelle carte del centro.

DIAGRAMMA DI GANTT



[Link gantt](#)

REGOLAMENTO DEL GIOCO RUBAMAZZO

Rubamazzo si gioca in due (Giocatore 1 vs Giocatore 2 *oppure* Giocatore 1 vs computer). Si danno tre carte a ciascun giocatore, quindi mette quattro carte sul tavolo. Quando i due giocatori hanno giocato tutte le proprie carte, il mazziere ne distribuisce altre tre a ciascuno: questo si ripete fino ad esaurimento del mazzo.

Ciascun giocatore a turno gioca una carta fino ad esaurimento delle carte. Se la carta giocata è di valore uguale a quello di una carta, il giocatore prende la carta da lui giocata e quella di valore corrispondente, e le pone in un mazzetto di fronte a sè con la faccia rivolta verso l'alto.

E' possibile inoltre rubare il mazzo avversario se si gioca una carta il cui valore corrisponde a quello della carta disposta a faccia in su del mazzetto avversario.

Vince la partita il giocatore che ha un numero maggiore di carte nel proprio mazzo.

A cura di Catellani Filippo e Verma Abhijeet
4BI 25 - 02 - 2024