

INGEGNERIA DEL SOFTWARE: MONOPOLY

Documentazione del progetto “Elaborato 2014/2015”

Università degli Studi di Brescia

Dipartimento di Ingegneria dell'Informazione

Via Branze 38

25231- Brescia



Relazione dell'elaborato, parte I

Autori

Nome: Falletti Davide

Matricola: 85973

Nome: Mele Fabrizio

Matricola: 85971

Nome: Cordioli Francesco

Matricola: 86538

Sommario

1 Introduzione.....	3
2 Strumenti applicativi utilizzati.....	3
3 Modello di processo.....	3
4 Manuale d'uso.....	4
5 Release 1.....	9
5.1 Diagramma UML delle classi.....	9
5.2 Casi d'uso.....	10
6 Release 2.....	12
6.1 Diagramma UML delle classi.....	12
6.2 Casi d'uso.....	13
7 Release 3.....	15
7.1 Diagramma UML delle classi.....	15
7.2 Casi d'uso.....	16
8 Release 4.....	19

8.1 Diagramma UML delle classi.....	2545
8.2 Casi d'uso.....	312423
9 Release 5.....	9
9.1 Diagramma UML delle classi.....	2545
9.2 Casi d'uso.....	312423
10 Testing.....	19

Ingegneria del Software

Monopoly

Introduzione

Il progetto ha l'obiettivo di realizzare tramite un processo di sviluppo incrementale un'applicazione software che consenta di giocare a Monopoly.

Strumenti applicativi utilizzati

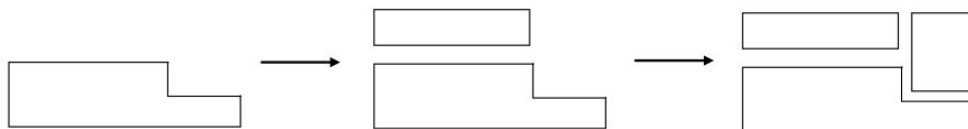
Linguaggio di programmazione:	Java
Ambiente di sviluppo:	Eclipse
Strumenti di condivisione:	GitHub,Sourcetree

Modello di processo

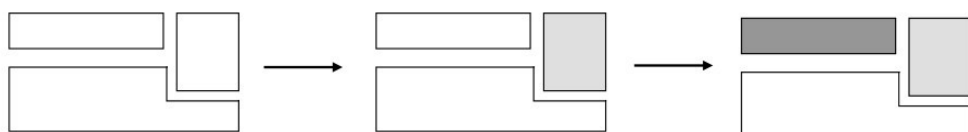
Il modello di processo adottato per lo sviluppo del software è stato il modello Incrementale/Iterativo. Questo modello consente di creare più versioni del software, ognuna delle quali integra un maggior numero di requisiti rispetto alla versione precedente

Le varie fasi possono essere rappresentate da questo modello:

Approccio incrementale:



Approccio Iterativo:



Manuale d'uso

Installazione

L'applicazione è stata scritta nel linguaggio di programmazione OO Java, quindi è eseguibile su qualsiasi calcolatore che abbia installato JRE (*Java Runtime Enviroment*).

Esecuzione

Per l'avvio dell'applicazione (.jar) eseguire i seguenti passaggi:

Aprire il Terminale di sistema (Shell di Unix o Prompt di Windows, a seconda del Sistema Operativo)

- Portarsi nella cartella in cui è presente il file **Progetto_Exe.jar**
- Eseguire il file .jar mediante il comando `"java -jar Progetto_Exe.jar"`

A questo punto, se non vengono visualizzati errori a causa di una errata versione di JRE, dovrebbe presentarsi questa situazione

Comandi

```
~~~~~  
~ BENVENUTO NEL GIOCO DEL MONOPOLY ~  
~~~~~
```

```
-----  
Menu' Monopoly  
-----  
1      Inserisci Nuovo Giocatore  
2      Stampa Elenco Giocatori  
3      Gioca  
4      Rimuovi Giocatore  
5      Reset  
  
0      Esci
```

Digita il numero dell'opzione desiderata >

Nome	Inserimento nuovo giocatore
Numero	1
Funzione	L'applicazione permette di inserire il nome di un nuovo giocatore.
Immagini	<pre>Digita il numero dell'opzione desiderata > 1 Inserisci un nuovo giocatore > Dab</pre>
Note	Se è già stato inserito il numero massimo di giocatori, il programma impedisce l'inserimento di un nuovo giocatore e lo comunica all'utente.

Nome	Stampa elenco giocatori
Numero	2
Funzione	L'applicazione permette di visualizzare l'elenco di tutti i giocatori che sono stati inseriti. caricato!il!modello!nel!sistema!
Immagini	! <p> Digita il numero dell'opzione desiderata > 2 1 - Dab[0e] 2 - fab[0e] 3 - cioppo[0e] </p>
Note	Se non ci sono giocatori inseriti, il programma lo comunica all'utente.

Nome	Gioca
Numero	3
Funzione	L'applicazione permette di giocare una partita a Monopoly, permettendo l'avanzamento dei turni tramite la pressione del tasto invio
Immagini	! <p> Digita il numero dell'opzione desiderata > 3 Turno 1 ***** </p> <p> [cioppo] lancia i dadi ed escono: 2 e 6. Si muove da Via![0] a Viale Monterosa[8]. [cioppo] ha acquistato Viale Monterosa Il giocatore ha 4900 euro </p> <p> ***** </p> <p> [fab] lancia i dadi ed escono: 6 e 6. Dadi doppi! Si muove da Via![0] a Societa Elettrica[12]. [fab] ha acquistato Societa Elettrica Il giocatore ha 4850 euro </p> <p> Ritira! ***** </p> <p> [fab] lancia i dadi ed escono: 1 e 3. Si muove da Societa Elettrica[12] a Via Verdi[16]. [fab] ha acquistato Via Verdi Il giocatore ha 4670 euro </p> <p> ***** </p> <p> [Dab] lancia i dadi ed escono: 6 e 5. Si muove da Via![0] a Via Accademia[11]. [Dab] ha acquistato Via Accademia Il giocatore ha 4860 euro </p> <p>!</p>

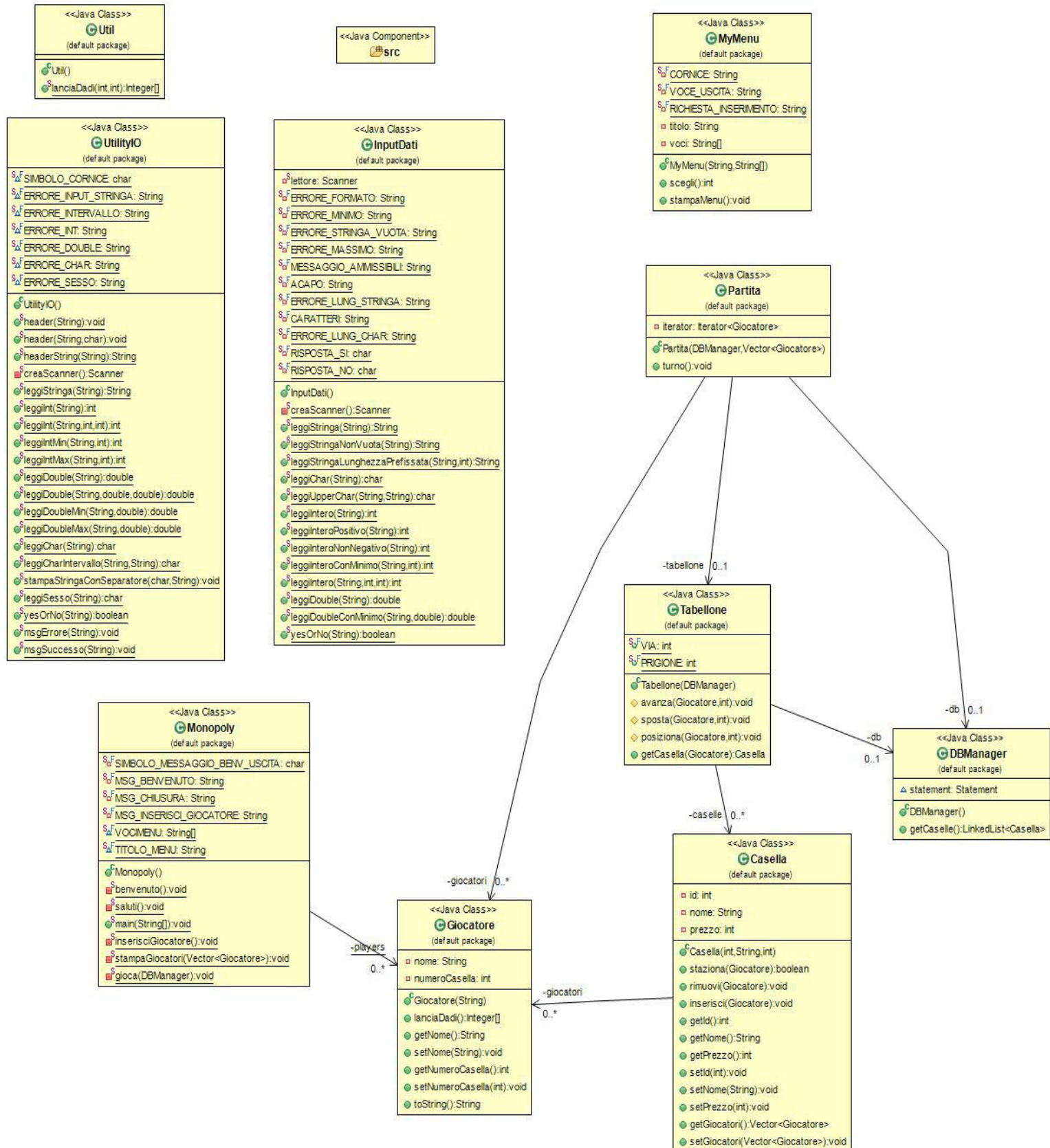
Immagini	<p>Turno 20 *****</p> <p>[cioppo] lancia i dadi ed escono: 5 e 5. Dadi doppi! Si muove da Via Marco Polo[21] a Via Roma[31].</p> <p>Il giocatore ha 4562 euro</p> <p>Ritira! *****</p> <p>[cioppo] lancia i dadi ed escono: 1 e 6. Si muove da Via Roma[31] a Tassa di Lusso[38].</p> <p>Il giocatore ha 4552 euro</p> <p>*****</p> <p>[fab] lancia i dadi ed escono: 3 e 4. Si muove da Stazione Est[35] a Via![0]. fab[4853e] pesca una carta Probabilita' *****</p> <p>Andate avanti sino al Via e ritirate 500e dalla banca. *****</p> <p>Il giocatore ha 5353 euro</p> <p>*****</p> <p>[Dab] lancia i dadi ed escono: 1 e 1. Dadi doppi! Si muove da Piazza università[14] a Via Verdi[16]. Dab paga a fab 18.0 Il giocatore ha 5737 euro</p> <p>Ritira! *****</p> <p>[Dab] lancia i dadi ed escono: 1 e 5. Si muove da Via Verdi[16] a Imprevisti[22]. Dab[5737e] pesca una carta Imprevisti' *****</p> <p>La banca vi sgancia gli interessi del vostro conto corrente: ritirate 125e *****</p> <p>Il giocatore ha 5862 euro</p> <p>Il vincitore e' Dab, con un capitale finale di 5862 euro</p>
----------	--

Nome	Rimuovi giocatore
Numero	4
Funzione	L'applicazione permette di rimuovere un giocatore, presentando la lista dei giocatori precedentemente inseriti
Immagini	<p>Digita il numero dell'opzione desiderata > 4</p> <pre> ----- Rimozione Giocatori ----- 1 cioppo[4552e] 2 fab[5371e] 3 Dab[5862e] 0 Esci Digita il numero dell'opzione desiderata ></pre>
Note	I valori da inserire possono variare rispetto a quelli presentati nell'immagine, a seconda delle opzioni disponibili.

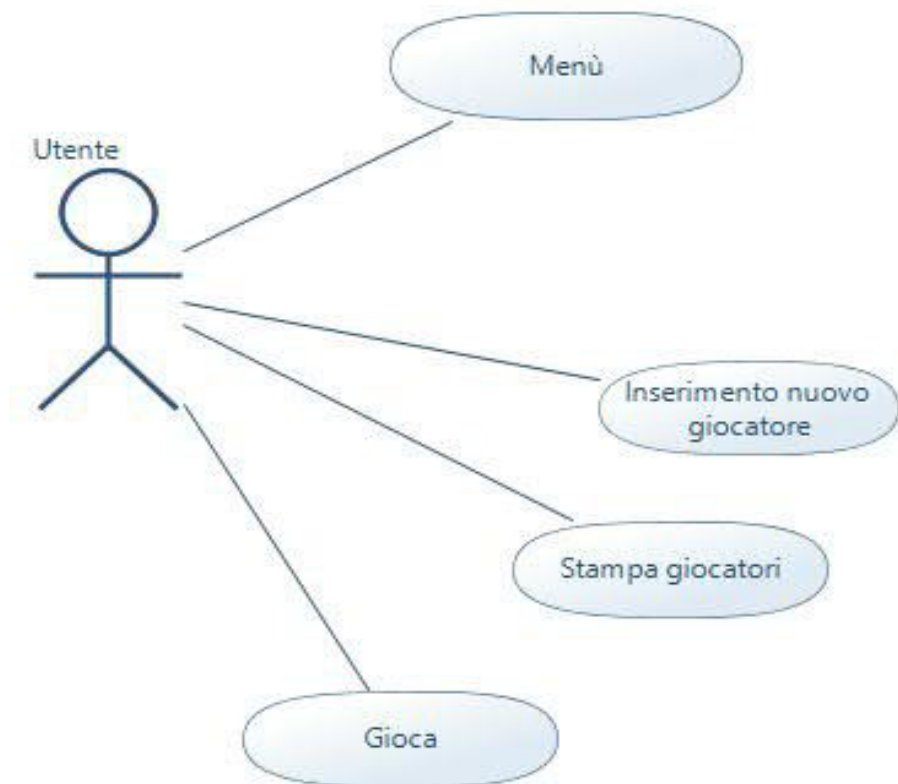
Nome	Esci
Numero	0
Funzione	L'applicazione si chiude.
Immagini	<pre> ' Digita il numero dell'opzione desiderata > 0 ~~~~~ ~ GRAZIE PER AVERE USATO IL PROGRAMMA - ARRIVEDERCI ~ ~~~~~ </pre>
Note	Per la riesecuzione del programma seguire nuovamente la Procedura di Esecuzione.

Release 1

Diagramma UML delle classi



Casi d'uso



Nome	Menu Principale
Attore	Utente
Scenario Principale	<ol style="list-style-type: none">1. L'applicazione presenta il menu di scelta2. L'utente sceglie l'azione da svolgere3. L'utente ha scelto di uscire dall'applicazione
Scenario Alternativo	<p>3.a L'utente ha scelto un'operazione da svolgere</p> <p>L'applicazione svolge l'operazione selezionata, torna al punto 1!</p>
Scenario Alternativo	<p>3.a L'utente digita un input errato</p> <p>Viene visualizzato un messaggio d'errore</p> <p>Torna al punto 1!</p>

Gestione del menu':

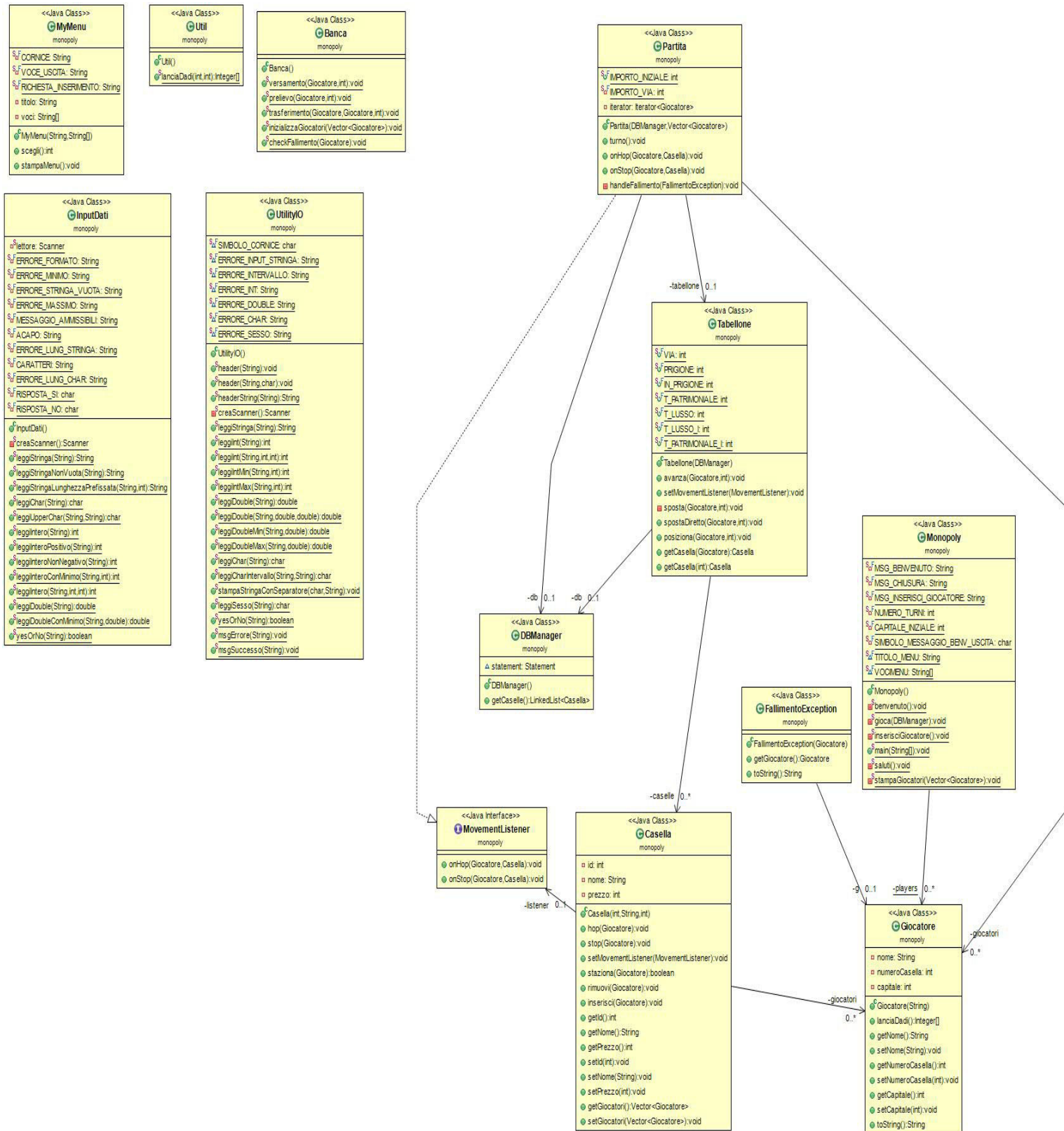
Nome	Inserimento nuovo giocatore
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente sceglie di inserire un nuovo giocatore 2. Il sistema fa inserire all'utente il nome del giocatore 3. Il sistema controlla che non ci siano più di 6 giocatori <p>Postcondizione: i nomi dei giocatori vengono memorizzati nel sistema.</p>
Scenario Alternativo	<p>1.a L'utente decide di inserire un nuovo giocatore</p> <p>Torna al punto 2</p>
Scenario Alternativo	<p>3.a L'utente inserisce più di 6 giocatori</p> <p>Il sistema rileva che ci sono più di 6 giocatori e stampa a video un messaggio d'errore</p>

Nome	Stampa giocatori
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente sceglie di stampare a video la lista dei giocatori 2. Il sistema stampa la lista dei giocatori inseriti

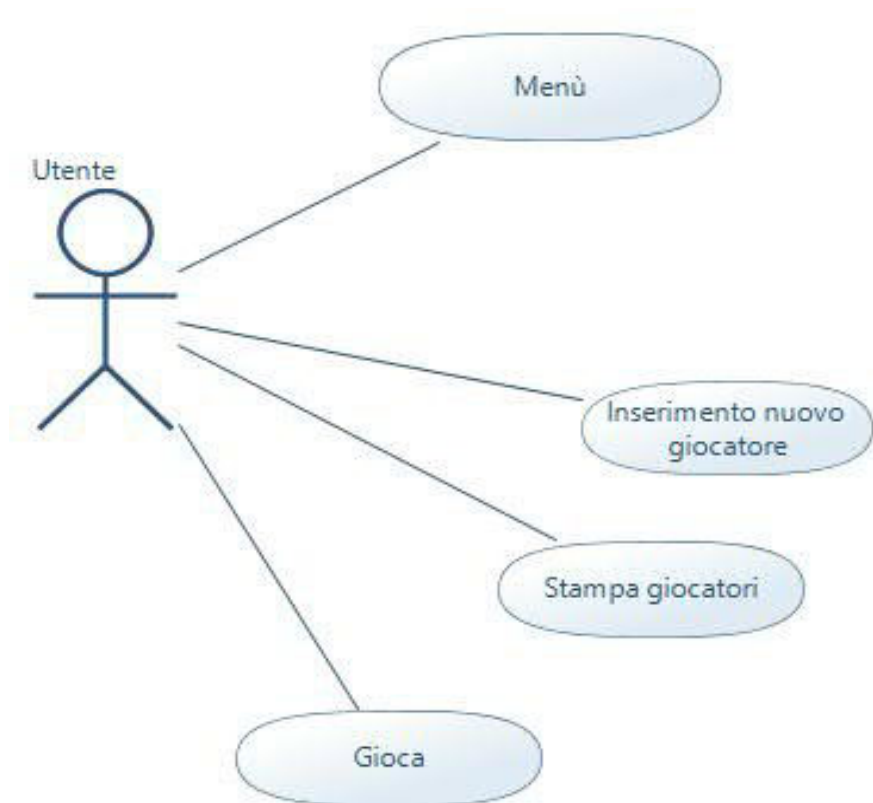
Nome	Gioca
Attore	Utente
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente sceglie di iniziare la partita 2. Il sistema controlla che i giocatori siano più di 2 3. Il sistema svolge la partita 4. Il sistema termina la partita e stampa a video tutti i turni con i relativi lanci di dadi dei giocatori e i loro movimenti
Scenario Alternativo	<p>2.a L'utente inserisce meno di 2 giocatori</p> <p>Il sistema rileva che ci sono meno di 2 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

Release 2

Diagramma UML delle classi



Casi d'uso



Nome	Menu Principale
Attore	Utente
Scenario Principale	<ol style="list-style-type: none">1. L'applicazione presenta il menu di scelta2. L'utente sceglie l'azione da svolgere3. L'utente ha scelto di uscire dall'applicazione
Scenario Alternativo	<p>3.a L'utente ha scelto un'operazione da svolgere</p> <p>L'applicazione svolge l'operazione selezionata</p>
Scenario Alternativo	<p>3.a L'utente digita un input errato</p> <p>Viene visualizzato un messaggio d'errore</p>

Gestione del menù:

Nome	Inserimento nuovo giocatore
Attore	Utente
Scenario Principale	<ol style="list-style-type: none">4. L'utente sceglie di inserire un nuovo giocatore5. Il sistema fa inserire all'utente il nome del giocatore6. Il sistema controlla che non ci siano più di 6 giocatori <p>Postcondizione: i nomi dei giocatori vengono memorizzati nel sistema.</p>

Scenario Alternativo	<p>1.a L'utente decide di inserire un nuovo giocatore</p> <p>Torna al punto 2</p>
Scenario Alternativo	<p>3.a L'utente inserisce più di 6 giocatori</p> <p>Il sistema rileva che ci sono più di 6 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

Nome	Stampa giocatori
Attore	Utente
Scenario Principale	<p>3. L'utente sceglie di stampare a video la lista dei giocatori</p> <p>4. Il sistema stampa la lista dei giocatori inseriti e il relativo capitale</p>

Nome	Gioca
Attore	Utente
Scenario Principale	<p>1. L'utente sceglie di iniziare la partita</p> <p>2. Il sistema controlla che i giocatori siano più di 2</p> <p>3. Il sistema svolge la partita</p> <p>4. Il sistema termina la partita e stampa a video tutti i turni con i relativi lanci di dadi dei giocatori, i loro movimenti e i loro capitali attuali</p> <p>5. Il sistema dichiara il vincitore</p>
Scenario Alternativo	<p>2.a L'utente inserisce meno di 2 giocatori</p> <p>Il sistema rileva che ci sono meno di 2 giocatori e stampa a video un messaggio d'errore</p> <p>Torna al menù principale</p>

Release 3

Diagramma UML delle classi

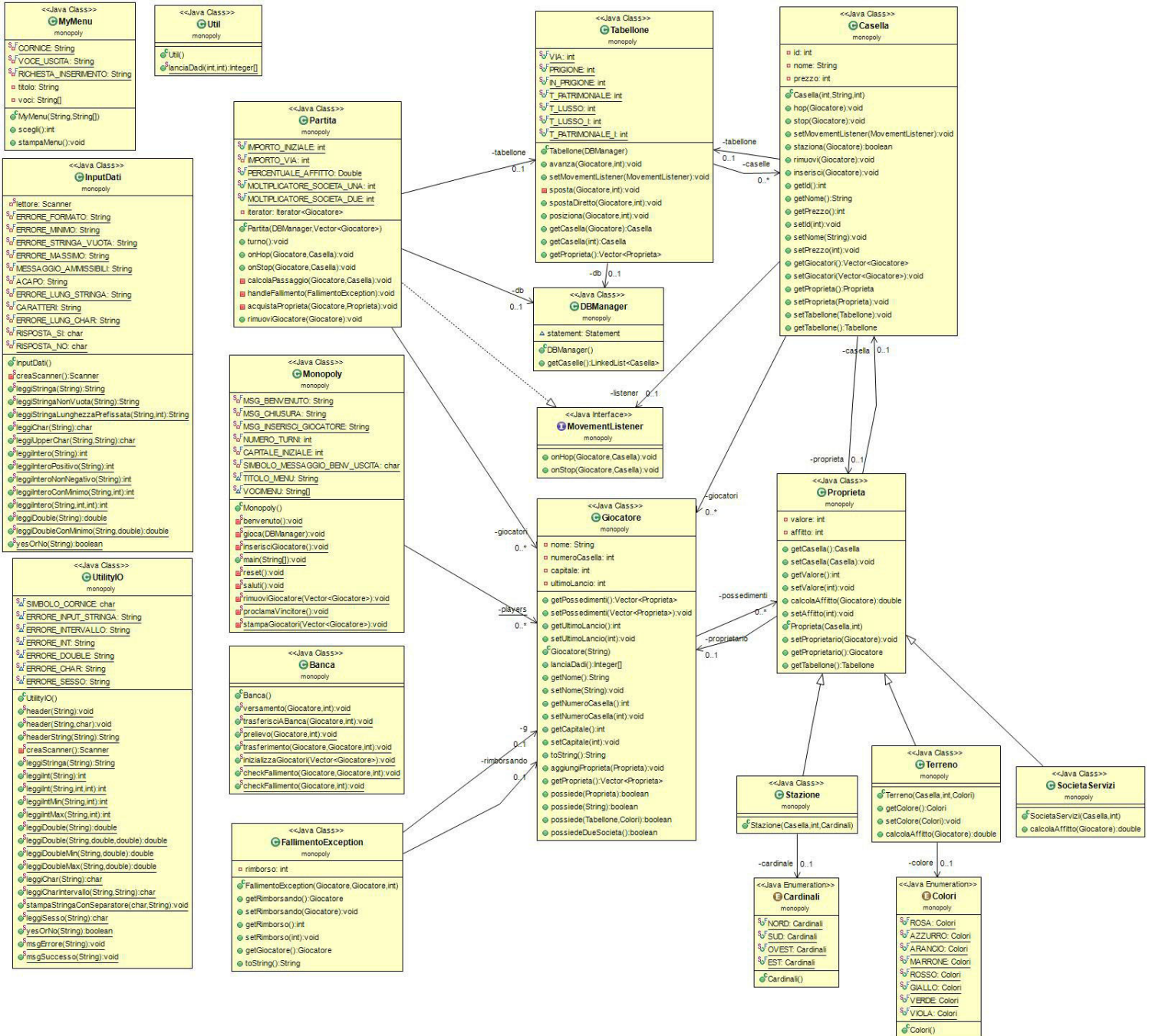
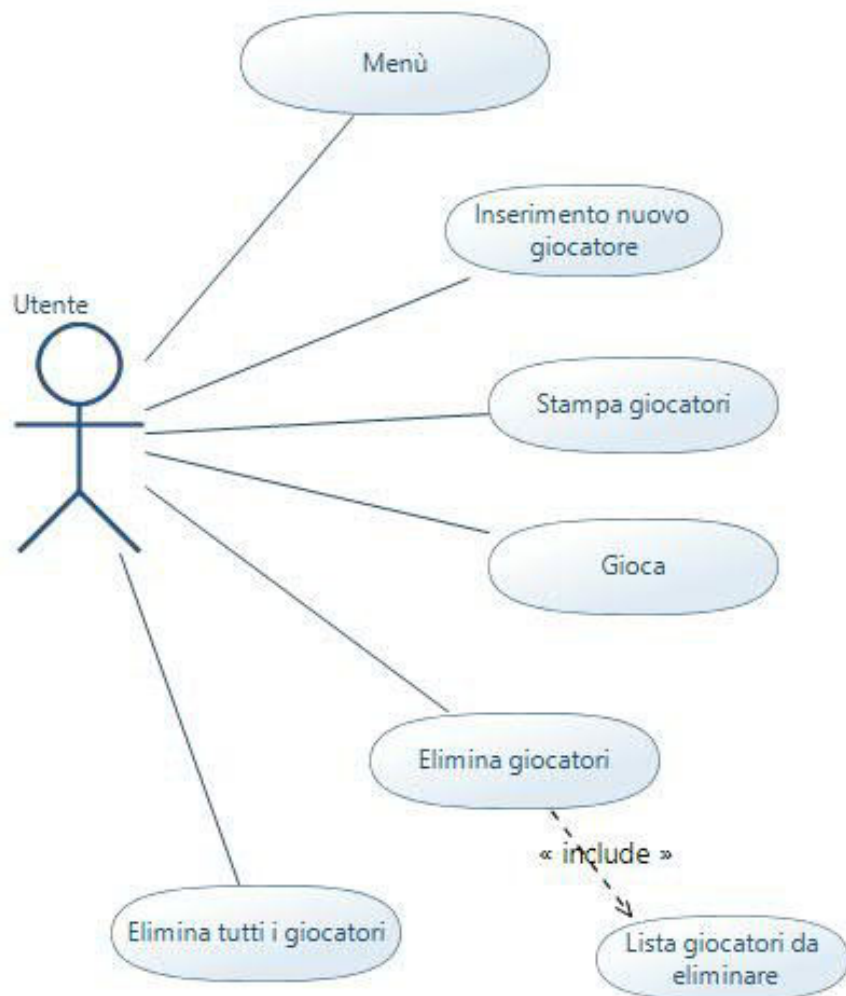


Diagramma UML casi d'uso



Casi d'uso

Release 4

Diagramma UML delle classi

Diagramma UML casi d'uso

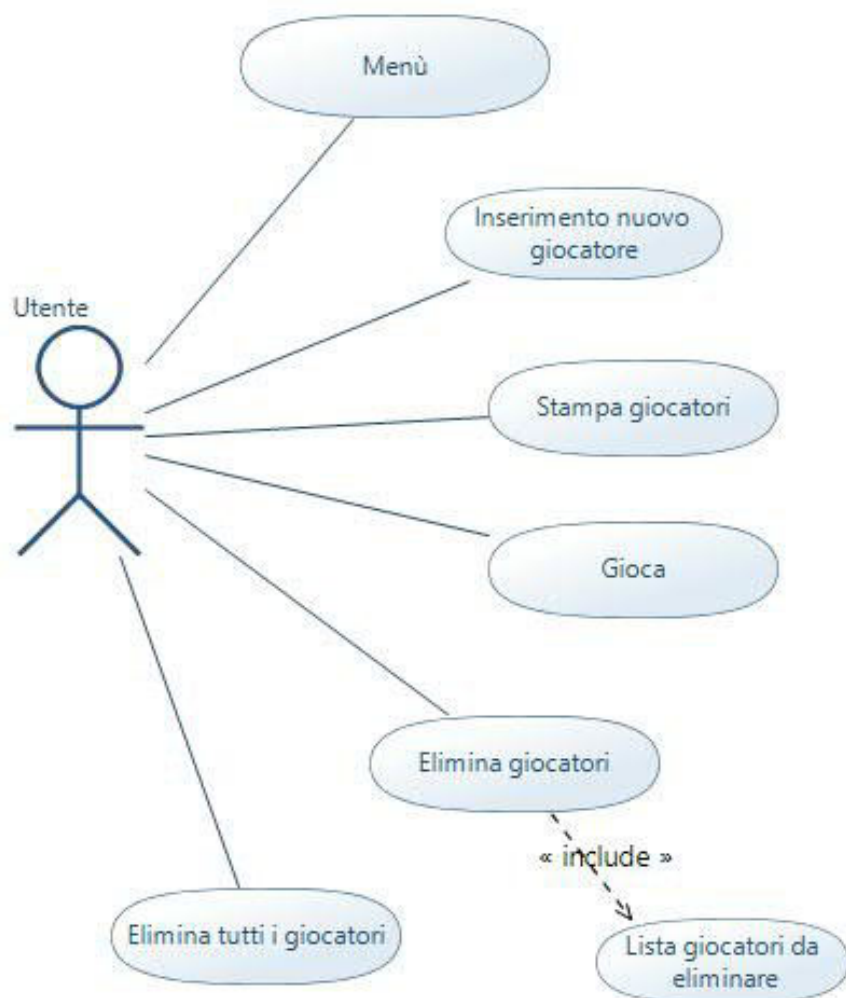
Casi d'uso

Release 5

Diagramma UML delle classi



Diagramma UML casi d'uso



Casi d'uso

Testing

Test Suite	1
Software Testato	Menu Principale.
Versione	1
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Risultato	Il Menu Principale risponde correttamente ai comandi dell'utente.
Autore	Cordioli Francesco.
Data	09/03/2015

Test Case	1
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Ambiente	<i>Driver</i> Main <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	0
Uscite attese	Uscita dal programma con relativo messaggio di uscita.
Uscite effettive	Le stesse.
Risultato'	PASS! Il menu principale risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	2
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Ambiente	<i>Driver</i> Main <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	1
Uscite attese	Inserimento e salvataggio di un nuovo giocatore.
Uscite effettive	Le stesse.
Risultato'	PASS! Il menu principale risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	3
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Ambiente	<i>Driver</i> Main <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	2
Uscite attese	Visualizzazione dei giocatori inseriti dall'utente.
Uscite effettive	Le stesse.
Risultato'	PASS! Il menu principale risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	4
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Ambiente	<i>Driver</i> Main <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	3

Uscite attese	Visualizzazione della partita giocata con il numero della faccia dei dadi tirati e spostamento del giocatore.
Uscite effettive	Le stesse.
Risultato'	PASS! Il menu principale risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	5
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Ambiente	<i>Driver:</i> Main <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	4
Uscite attese	Messaggio di errore e ritorno al menu principale.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	E' stato dato in ingresso un'opzione del menù che non esiste.

Test Suite	2
Software Testato	Inserimento dei giocatori.
Versione	1
Obiettivo	Verificare il corretto inserimento dei giocatori e il numero massimo di essi.
Risultato	L'inserimento risponde correttamente ai comandi dell'utente.
Autore	Cordioli Francesco.
Data	09/03/2015

Test Case	1
------------------	----------

Obiettivo	Verificare il corretto inserimento dei dati.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	“Dab”.
Uscite attese	Salvataggio del nome.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	Se vengono inseriti più di 6 giocatori viene stampato a video un messaggio che informa l'utente che è stato raggiunto il numero massimo di giocatori.

Test Case	2
Obiettivo	Verificare il corretto inserimento dei dati
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Messaggio d'errore.
Uscite effettive	Salvataggio del nome.
Risultato'	FAIL! Il programma non risponde correttamente ai comandi dell'utente.
Osservazioni	Manca il controllo sul mancato inserimento del nome del giocatore.

Test Case	3
Obiettivo	Verificare il corretto inserimento dei dati
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.

Uscite attese	Messaggio d'errore.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	Sistemato il controllo sull'inserimento del nome.

Test Suite	3
Software Testato	Visualizzazione dei giocatori.
Versione	1
Obiettivo	Verificare la corretta visualizzazione della lista dei giocatori inseriti dall'utente.
Risultato	La visualizzazione risponde correttamente ai comandi dell'utente.
Autore	Cordioli Francesco.
Data	09/03/2015

Test Case	1
Obiettivo	Verificare la corretta visualizzazione della lista dei giocatori.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Lista dei nomi.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

	4
Software Testato	Gioco-Partita.

Versione	1
Obiettivo	Verificare il corretto funzionamento del gioco.
Risultato	Il gioco funziona come da consegna.
Autore	Cordioli Francesco.
Data	09/03/2015

Test Case	1
Obiettivo	Verificare il corretto funzionamento del gioco.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Lista di 20 turni con varie descrizioni di dadi e posizioni aggiornate.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	Il programma da messaggio di errore se sono stati inseriti meno di 2 giocatori.

Test Case	2
Obiettivo	Verificare che tutti i giocatori inizino la partita dalla casella VIA!
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	I giocatori si trovano tutti sulla casella VIA!
Uscite effettive	Le stesse.

Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	3
Obiettivo	Verificare che dopo 3 volte che il giocatore ottiene lo stesso punteggio per entrambi i dadi finisce in prigione.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore dopo 3 tiri uguali finisce in prigione.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	4
Obiettivo	Verificare che una volta che il giocatore arriva alla casella 40 riparte dalla casella VIA!
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Alla conclusione di ogni giro il conteggio della posizione del giocatore riparte da 0.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.

Test Case	5
Obiettivo	Verificare che una partita termina dopo 20 turni.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Termine della partita.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Suite	5
Software Testato	Gioco-partita con l'interazione della banca.
Versione	1
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Risultato	Il risultato finale corrisponde alla consegna data.
Autore	Cordioli Francesco.
Data	10/03/2015

Test Case	1
Obiettivo	Verificare che ad ogni giocatore sia assegnata la cifra di 5000€.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.

Uscite attese	Ogni giocatore ha un capitale di 5000€.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	2
Obiettivo	Verificare che ad ogni passaggio dal via il giocatore riceva 500€.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Ogni utente riceve 500€ dalla banca.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	3
------------------	----------

Obiettivo	Verificare che dalla casella 30 ("IN PRIGIONE") venga spostato alla casella 10 ("PRIGIONE/TRANSITO") senza ricevere i 500€ dalla banca per il passaggio dal via.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Passaggio dalla casella 30 alla casella 10 e i soldi devono rimanere gli stessi.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	4
Obiettivo	Verificare il pagamento di 250€ alla banca per aver sostato sulla casella TASSA PATRIMONIALE.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore ha 250€ in meno.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	5
Obiettivo	Verificare il pagamento di 10€ alla banca per aver sostato sulla casella TASSA DEL LUSO.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore ha 10€ in meno.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	6
Obiettivo	Verificare il fallimento di un giocatore e l'uscita di esso dalla partita.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore esce dalla partita.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	7
Obiettivo	Verificare il vincitore della partita giocata.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Messaggio a video con il nome del vincitore.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	Il giocatore ha vinto perché o tutti gli altri giocatori hanno fallito oppure è quello più ricco.

Test Suite	6
Software Testato	Menu Principale.
Versione	1
Obiettivo	Verificare il corretto funzionamento delle due voci aggiunte al menù principale.
Risultato	Il Menu Principale risponde correttamente ai comandi dell'utente.
Autore	Falletti Davide.
Data	20/03/2015

Test Case	1
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Ambiente	<i>Driver</i> Main <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	4

Uscite attese	Visualizzazione di una lista di giocatori per l'eliminazione.
Uscite effettive	Le stesse.
Risultato'	PASS! Il menu principale risponde correttamente ai comandi dell'utente.
Osservazioni	Visualizzo una lista di giocatori da eliminare; per poter scegliere quale eliminare premo il numero della lista corrispondente al giocatore.

Test Case	2
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Ambiente	<i>Driver</i> Main <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	5
Uscite attese	Eliminazione di tutti i giocatori.
Uscite effettive	Le stesse.
Risultato'	PASS! Il menu principale risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Suite	7
Software Testato	Gioco-partita con l'interazione della banca e con il valore su ogni casella.
Versione	1
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Risultato	Il risultato finale corrisponde alla consegna data.
Autore	Falletti Davide.

Data	20/03/2015
------	------------

Test Case	1
Obiettivo	Verificare che ogni casella abbia un valore in euro.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Ogni casella ha il suo costo.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	2
Obiettivo	Verificare che quando un giocatore arriva su una casella che non è ancora stata acquistata, la acquista.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore acquista automaticamente la proprietà.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	3
Obiettivo	Verificare che il giocare acquisti la proprietà solo se ha denaro sufficiente.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	Nessuno.
Uscite attese	Messaggio di fondi insufficienti e non avviene l'acquisto della proprietà.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	4
Obiettivo	Verificare che quando un giocatore arriva su un terreno o una stazione di proprietà di un altro giocatore, deve pagare a quest'ultimo il 10% del valore della casella.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	Nessuno.
Uscite attese	Pagamento del 10% della casella all'avversario.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	5
-----------	---

Obiettivo	Verificare che quando un giocatore arriva su una società di proprietà di un altro giocatore, l'affitto deve essere 4 volte il numero ottenuto con i dadi.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	Nessuno.
Uscite attese	L'affitto pagato (in €) all'avversario è dato dal valore dei dadi moltiplicato per 4.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	Questo avviene se l'avversario possiede una sola società di servizi.

Test Case	6
Obiettivo	Verificare che quando un giocatore arriva su una società di proprietà di un altro giocatore, l'affitto deve essere 10 volte il numero ottenuto con i dadi.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle</i> : Main <i>Stub</i> :
Dati in ingresso	Nessuno.
Uscite attese	L'affitto pagato (in €) all'avversario è dato dal valore dei dadi moltiplicato per 10.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	Questo avviene se l'avversario possiede entrambe le società di servizi.

Test Case	7
------------------	----------

Obiettivo	Verificare che se il giocatore possiede tutti i terreni di un certo colore, l'affitto raddoppia.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	L'affitto pagato all'avversario è doppio.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	8
Obiettivo	Verificare il fallimento di un giocatore.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore esce dal gioco.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	Questo avviene se un giocatore non possiede un capitale sufficiente per pagare per intero l'affitto dovuto a un altro giocatore.
Test Case	9
Obiettivo	Verificare che se un giocatore si sposta su una casella di un altro giocatore ma non ha i soldi per pagare l'affitto, fallisce dando il suo capitale al giocatore che possiede la proprietà e quest'ultimo riceve la differenza del valore dovuto dalla banca.

Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	L'avversario riceve i soldi dal giocatore che ha fallito e riceve il resto dei soldi dalla banca.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	10
Obiettivo	Verificare che le proprietà del giocare fallito ritornino acquistabili.
Ambiente	<i>Driver</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Le proprietà tornano acquistabili.
Uscite effettive	Le stesse.
Risultato	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Suite	8
Software Testato	Gioco-partita con l'aggiunta delle PROBABILITA' e degli IMPREVISTI.
Versione	1

Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Risultato	Il risultato finale corrisponde alla consegna data.
Autore	Falletti Davide.
Data	27/03/2015

Test Case	1
Obiettivo	Verificare che all'inizio di ogni partita ciascuno dei due mazzi (IMPREVISTI e PROBABILITA') venga mescolato casualmente.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Mazzo mescolato casualmente.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	2
Obiettivo	Verificare che il giocatore quando arriva su una casella di probabilità o imprevisi, peschi la carta in cima al mazzo.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore pesca la prima carta del mazzo.
Uscite effettive	Le stesse.
Risultato'	PASS!
Osservazioni	

Test Case	3
Obiettivo	Verificare che il giocatore esegua l'operazione scritta sulla carta di probabilità o imprevisti.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore si comporterà in modo diverso in base all'operazione scritta sulla carta.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	4
Obiettivo	Verificare che il giocatore depositi la carta in fondo al mazzo.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore mette la carta in fondo al mazzo.
Uscite effettive	Le stesse.
Risultato'	PASS!
Osservazioni	

Test Case	5
Obiettivo	Verificare il fallimento di un giocatore.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore fallisce ed esce dalla partita.

Uscite effettive	Le stesse.
Risultato'	PASS!
Osservazioni	Un giocatore fallisce quando non possiede un capitale sufficiente da eseguire l'azione scritta sulla carta di probabilità o di imprevisti.

Test Case	6
Obiettivo	Verificare che se un giocatore fallisce, l'eventuale resto che dovrebbe dare all'avversario venga fornito dalla banca.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	L'avversario riceve una certa quantità di denaro dalla banca.
Uscite effettive	Le stesse.
Risultato'	PASS!
Osservazioni	L'eventuale resto è dovuto dal fatto che il giocatore non ha soldi per eseguire l'azione scritta sulla carta di probabilità o imprevisti e fallendo cede tutto il suo capitale all'avversario a cui è dovuto.

Test Suite	9
Software Testato	Gioco-partita con l'aggiunta di alcune regole per la gestione della PRIGIONE.
Versione	1
Obiettivo	Verificare il corretto funzionamento del Menu Principale.
Risultato	Il risultato finale corrisponde alla consegna data.
Autore	Cordioli Francesco.
Data	03/04/2015

Test Case	1
------------------	---

Obiettivo	Verificare che il giocatore finisca in prigione quando pesca una carta di PROBABILITA' O IMPREVISITI che gli indica di andare in prigione.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore entra in prigione.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	2
Obiettivo	Verificare che il giocatore finisca in prigione quando, tirando i dadi, esce per tre volte dadi doppi.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore finisce in prigione.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	3
Obiettivo	Verificare che, al turno successivo, prima di tirare i dadi, il giocatore paghi 50€ alla banca per uscire di prigione.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>

Dati in ingresso	Nessuno.
Uscite attese	Il giocatore esce di prigione e la banca ha 50€ in più.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	4
Obiettivo	Verificare che se un giocatore è in prigione e non possiede un capitale sufficiente per uscirne, fallisce.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore fallisce.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	5
Obiettivo	Verificare che tutte le proprietà del giocatore fallito ritornino acquistabili.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Le proprietà sono riacquistabili.
Uscite effettive	Le stesse.

Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	6
Obiettivo	Verificare che tutto il capitale del giocatore fallito venga ceduto alla banca.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	La banca ha in più il denaro del giocatore fallito.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	

Test Case	7
Obiettivo	Verificare che il giocatore fallito esca dalla partita.
Ambiente	<i>Driver:</i> Terminal_in_out <i>Oracle:</i> Main <i>Stub:</i>
Dati in ingresso	Nessuno.
Uscite attese	Il giocatore non c'è più nella partita.
Uscite effettive	Le stesse.
Risultato'	PASS! Il programma risponde correttamente ai comandi dell'utente.
Osservazioni	