INGEGNERIA DEL SOFTWARE:

MONOPOLY

Documentazione del progetto “Elaborato 2014/2015”

Università degli Studi di Brescia

Dipartimento di Ingengeria dell'Informazione

Via Branze 38

25231- Brescia

**Relazione dell'elaborato, I parte**

Autori:

Nome: Falleti Davide

Matricola:

Nome: Mele Fabrizio

Matricola:

Nome: Cordioli Francesco

Matricola:

Sommario

# Introduzione..............................................3

# Strumenti applicativi utilizzati...................3

# Modello di processo...................................3

# Manuale d’uso...........................................4

# Casi d’uso..................................................9

# Testing.......................................................19

# JavaDoc...????????????????.......................61

Ingegneria del Software

Monopoly

Introduzione

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

Strumenti applicativi utilizzati

Linguaggio di programmazione: Java

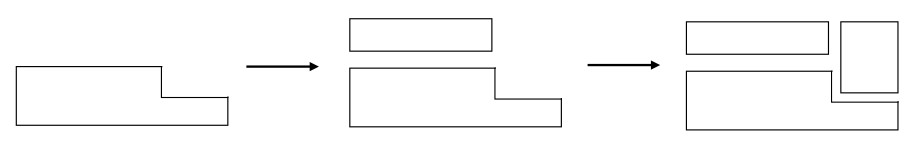
Ambiente di sviluppo: Eclipse

Strumenti di condivisione: GitHub,Sourcetree

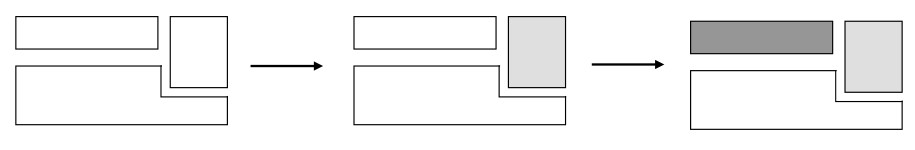
Modello di processo

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 suqualsiasi calcolatore che abbia installato JRE (Java Runtime Enviroment).

#### Esecuzione

**Todo**  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

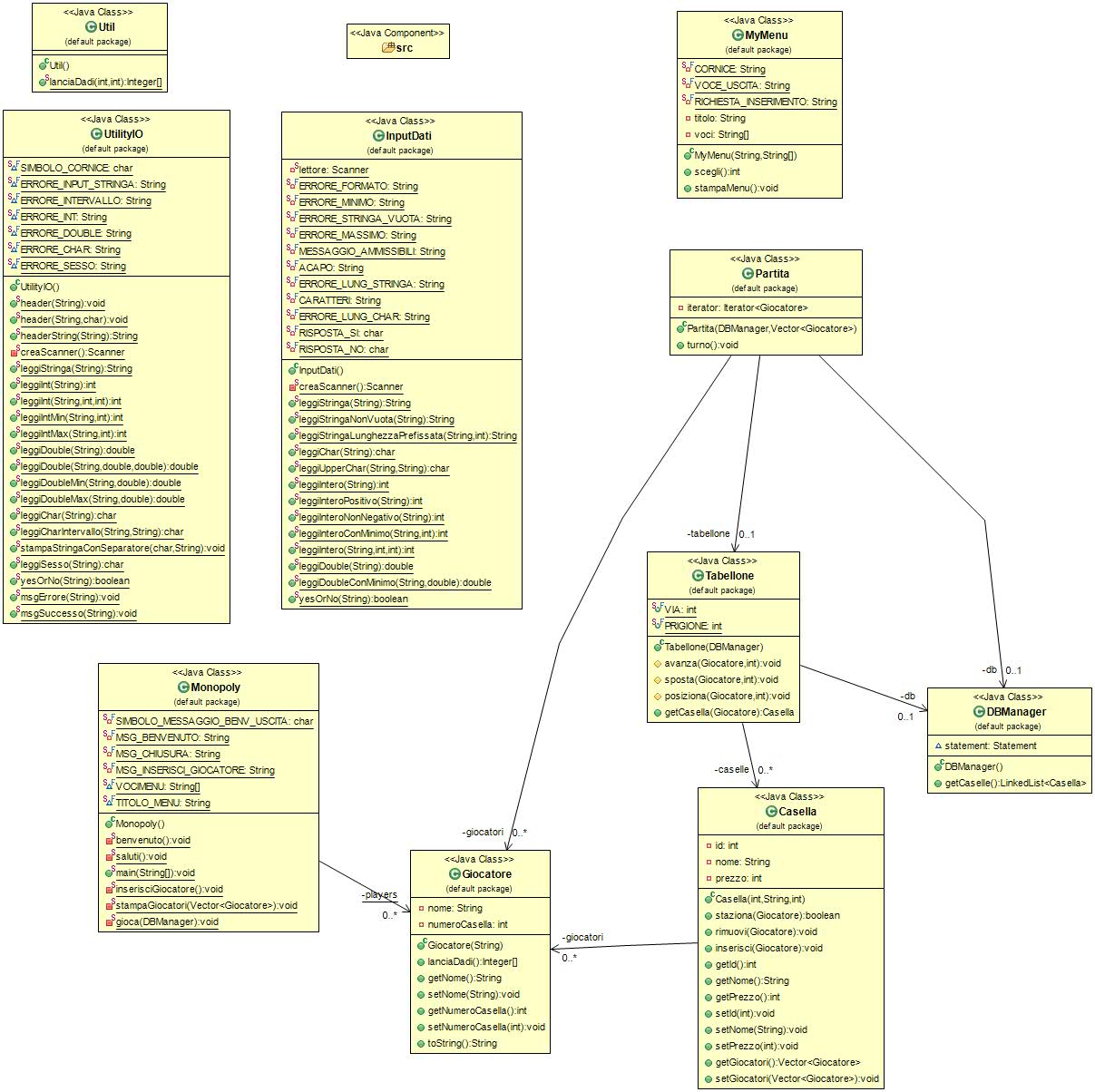
mettere screenshot menu principale di quando si avvia il monopoly

#### Comandi

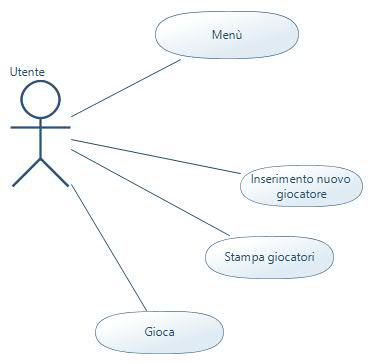
**Todo** quando avremo finito tutte e 5 le release

Release 1

#### Diagramma UML delle classi



#### Casi d'uso



|  |  |
| --- | --- |
| **Nome** | Menu Principale |
| **Attore** | Utente |
| **Scenario Principale** | 1. L’applicazione presenta il menu di scelta  2. L’utente sceglie l’azione da svolgere  3. L’utente ha scelto di uscire dall’applicazione  4. L’applicazione!viene!terminata! FINE! |
| **Scenario Alternativo** | 3.a L’utente ha scelto un’operazione da svolgere  L’applicazione svolge l’operazione selezionata Torna!al!punto!1! |
| **Scenario Alternativo** | 3.a L’utente digita un input errato  Viene visualizzato un messaggio d’errore Torna!al!punto!1.! |

Gestione del menu':

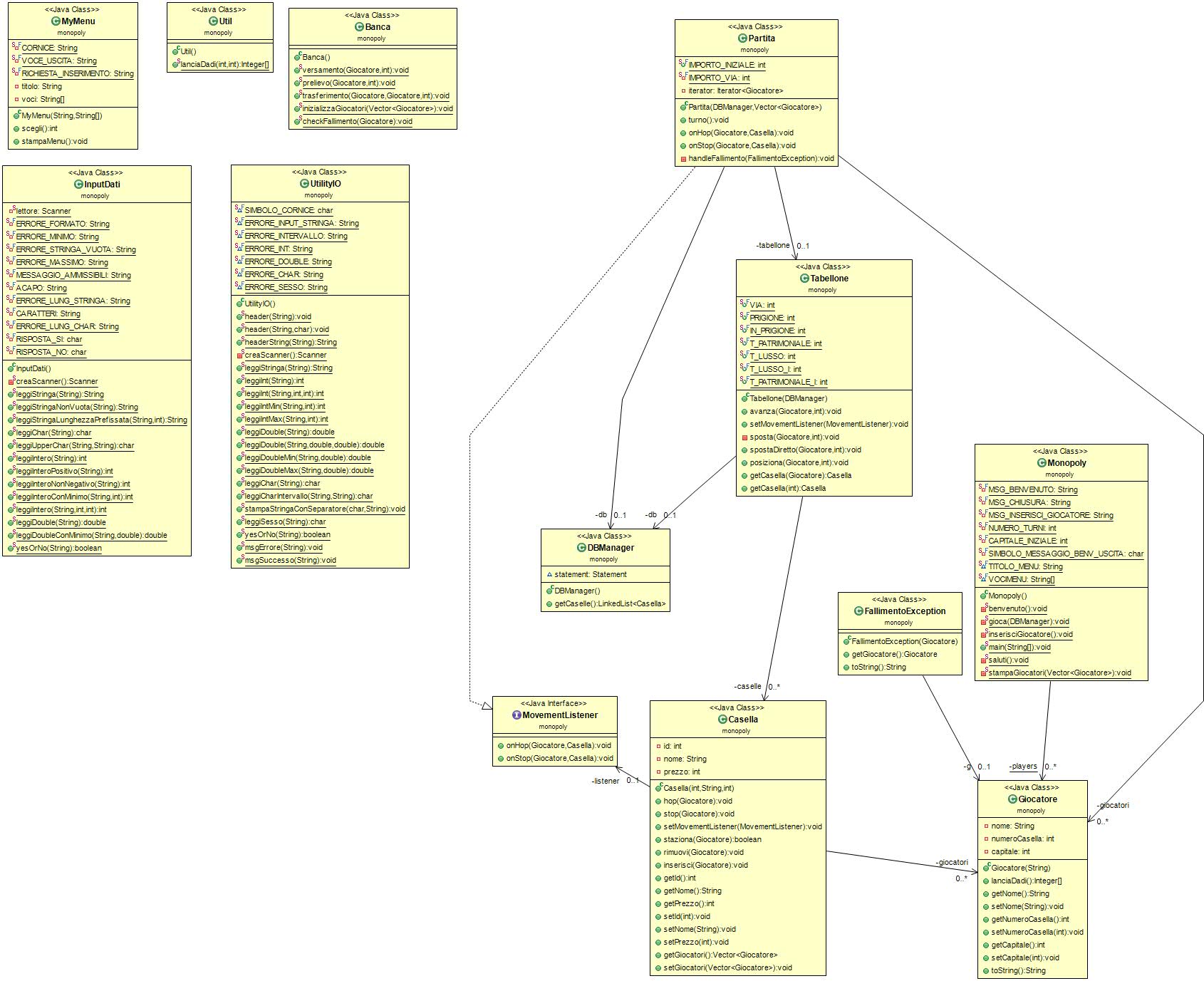
|  |  |
| --- | --- |
| **Nome** | Inserimento nuovo giocatore |
| **Attore** | Utente |
| **Scenario Principale** | 1. L’utente sceglie di inserire un nuovo giocatore 2. Il sistema fa inserire all’utente il nome del giocatore 3. Il sistema controlla che non ci siano più di 6 giocatori   Postcondizione: i nomi dei giocatori vengono memorizzati nel sistema. |
| **Scenario Alternativo** | 1.a L’utente decide di inserire un nuovo giocatore  Torna al punto 2 |
| **Scenario Alternativo** | 3.a L’utente inserisce più di 6 giocatori  Il sistema rileva che ci sono più di 6 giocatori e stampa a video un messaggio d’errore  Torna al menù principale |

|  |  |
| --- | --- |
| **Nome** | Stampa giocatori |
| **Attore** | Utente |
| **Scenario Principale** | 1. L’utente sceglie di stampare a video la lista dei giocatori 2. Il sistema stampa la lista dei giocatori inseriti |

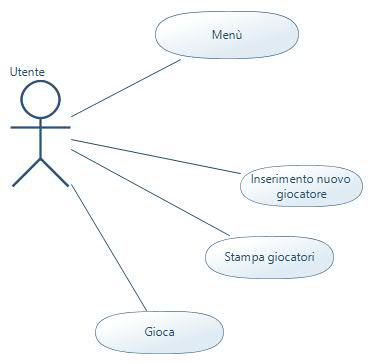
|  |  |
| --- | --- |
| **Nome** | Gioca |
| **Attore** | Utente |
| **Scenario Principale** | 1. L’utente sceglie di iniziare la partita 2. Il sistema controlla che i giocatori siano più di 2 3. Il sistema svolge la partita 4. Il sistema termina la partita e stampa a video tutti i turni con i relativi lanci di dadi dei giocatori e i loro movimenti |
| **Scenario Alternativo** | 2.a L’utente inserisce meno di 2 giocatori  Il sistema rileva che ci sono meno di 2 giocatori e stampa a video un messaggio d’errore  Torna al menù principale |

Release 2

#### Diagramma UML delle classi



#### Casi d'uso



|  |  |
| --- | --- |
| **Nome** | Menu Principale |
| **Attore** | Utente |
| **Scenario Principale** | 1. L’applicazione presenta il menu di scelta  2. L’utente sceglie l’azione da svolgere  3. L’utente ha scelto di uscire dall’applicazione  4. L’applicazione!viene!terminata! FINE! |
| **Scenario Alternativo** | 3.a L’utente ha scelto un’operazione da svolgere  L’applicazione svolge l’operazione selezionata Torna!al!punto!1! |
| **Scenario Alternativo** | 3.a L’utente digita un input errato  Viene visualizzato un messaggio d’errore Torna!al!punto!1.! |

Gestione del menù:

|  |  |
| --- | --- |
| **Nome** | Inserimento nuovo giocatore |
| **Attore** | Utente |
| **Scenario Principale** | 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   Postcondizione: i nomi dei giocatori vengono memorizzati nel sistema. |
| **Scenario Alternativo** | 1.a L’utente decide di inserire un nuovo giocatore  Torna al punto 2 |
| **Scenario Alternativo** | 3.a L’utente inserisce più di 6 giocatori  Il sistema rileva che ci sono più di 6 giocatori e stampa a video un messaggio d’errore  Torna al menù principale |

|  |  |
| --- | --- |
| **Nome** | Stampa giocatori |
| **Attore** | Utente |
| **Scenario Principale** | 1. L’utente sceglie di stampare a video la lista dei giocatori 2. Il sistema stampa la lista dei giocatori inseriti e il relativo capitale |

|  |  |
| --- | --- |
| **Nome** | Gioca |
| **Attore** | Utente |
| **Scenario Principale** | 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, i loro movimenti e i loro capitali attuali 5. Il sistema dichiara il vincitore |
| **Scenario Alternativo** | 2.a L’utente inserisce meno di 2 giocatori  Il sistema rileva che ci sono meno di 2 giocatori e stampa a video un messaggio d’errore  Torna al menù principale |

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** | *Driver* Main *Oracle:* Main  *Stub:* |
| **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** | *Driver* Main *Oracle:* Main  *Stub:* |
| **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** | *Driver* Main  *Oracle:* Main  *Stub:* |
| **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** | *Driver* Main  *Oracle:* Main  *Stub:* |
| **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** | *Driver:*  Main  *Oracle:* Main  *Stub:* |
| **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** | Francesco Cordioli |
| **Data** | 09/03/2015 |

|  |  |
| --- | --- |
| **Test Case** | 1 |
| **Obiettivo** | Verificare il corretto inserimento dei dati |
| **Ambiente** | *Driver:* Terminal\_in\_out  *Oracle:* Main  *Stub:* |
| **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** | *Driver:* Terminal\_in\_out  *Oracle:* Main  *Stub:* |
| **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** | *Driver:* Terminal\_in\_out  *Oracle:* Main  *Stub:* |
| **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** | Francesco Cordioli |
| **Data** | 09/03/2015 |

|  |  |
| --- | --- |
| **Test Case** | 1 |
| **Obiettivo** | Verificare la corretta visualizzazione della lista dei giocatori |
| **Ambiente** | *Driver:* Terminal\_in\_out  *Oracle:* Main  *Stub:* |
| **Dati in ingresso** | Nessuno. |
| **Uscite attese** | Lista dei nomi. |
| **Uscite effettive** | Le stesse. |
| **Risultato'** | PASS!  Il programma risponde correttamente ai comandi dell’utente. |
| **Osservazioni** |  |

|  |  |
| --- | --- |
| **Test Suite** | 4 |
| **Software Testato** | Gioco-Partita |
| **Versione** | 1 |
| **Obiettivo** | Verificare il corretto funzionamento del gioco |
| **Risultato** | Il gioco funziona come da consegna |
| **Autore** | Francesco Cordioli |
| **Data** | 09/03/2015 |

|  |  |
| --- | --- |
| **Test Case** | 1 |
| **Obiettivo** | Verificare il corretto funzionamento del gioco (o.o) |
| **Ambiente** | *Driver:* Terminal\_in\_out  *Oracle:* Main  *Stub:* |
| **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 |