Lorenzi Alessandro 1075244

Rota Gabriele 1079894

Rocco Alessandro 1081179

Per questo progetto abbiamo deciso di ricreare il gioco "Monopoly" in maniera digitale per avere un controllo automatico delle regole e velocizzare e semplificare i passaggi meccanici, come la conta del denaro durante i pagamenti, ma soprattutto per questioni pratiche come il salvataggio delle partite.

Il Monopoly è un gioco da tavolo nato agli inizi del 900' in cui i giocatori cercano di guadagnare il massimo denaro e causare la bancarotta agli altri partecipanti raggiungendo il monopolio di mercato.

I partecipanti si muovono attraverso una tabellone rappresentante una città, acquistano terreni, costruiscono case e hotel, fanno pagare gli altri giocatori quando transitano sulla loro proprietà e subiscono imprevisti o probabilità, che possono essere vantaggiose o svantaggiose.

DIFFICOLTA' INCONTRATE

- Utilizzo di java Swing;
- Comprensione delle funzionalità di GitHub;
- Suddivisione del lavoro;
- Rispetto dei requisiti e delle scadenze autoimposte;
- Creare una grafica accattivante;
- Fornire un'esperienza piacevole al player;

PARADIGMI DI PROGRAMMAZIONE

Linguaggio: l'applicativo è stato sviluppato interamente con il linguaggio Java, sfruttando la libreria Swing;

IDE: come ambiente di sviluppo abbiamo scelto Eclipse, insieme a CodeTogether e Jarchitect;

Comunicazione: per coordinarci e partecipare in modo concorrente allo sviluppo del software abbiamo optato per Discord e Google Meet. Tra meeting e brevi riunioni di allineamento sulle novità del codice siamo riusciti a rimanere aggiornati sulle modifiche effettuate dagli altri membri del gruppo;

Modellazione: StarUML ci è stato utile per la creazione dei modelli UML.

SOFTWARE CONFIGURATION MANAGEMENT

Inizialmente la documentazione in sviluppo e l'organizzazione del lavoro sono stati gestiti su Drive. Successivamente abbiamo utilizzato GitHub come SCM. Una volta creato il repository condiviso, abbiamo implementato le funzionalità base del progetto insieme ad un primo prototipo di interfaccia. Dopodichè sono nati i primi branch.

KanBan board

Ci ha aiutati ad organizzare il flusso di lavoro

Branch

Nascevano per implementare modifiche importanti e parti critiche del codice

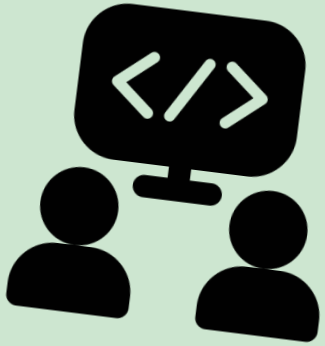
Issue

Hanno consentito al team di tenere traccia dei problemi, monitorandone il progesso nella loro risoluzione

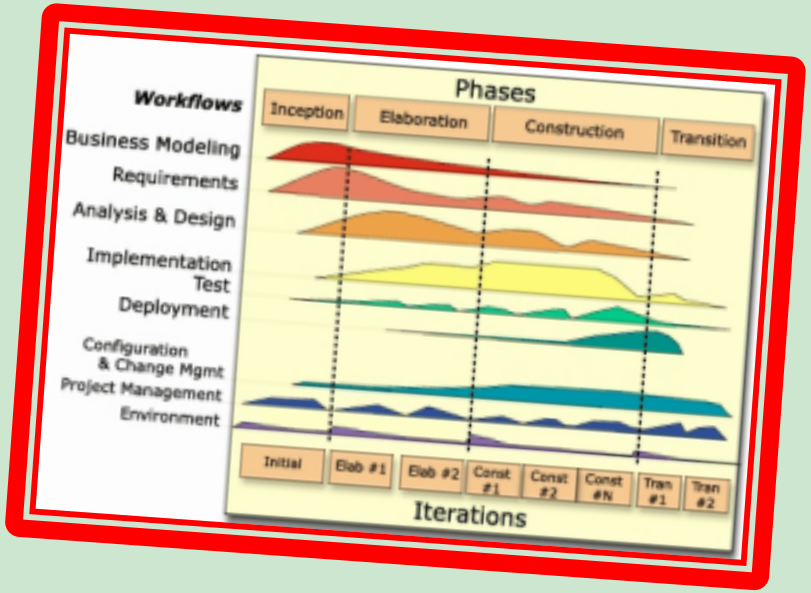
SOFTWARE LIFE CYCLE

Nello sviluppo abbiamo deciso di utilizzare un approccio agile, nello specifico il metodo RUP (*Rational Unified Process*), integrando pratiche agili.

Pair programming



CodeTogether



RUP

ANALISI DEI REQUISITI

MUST HAVE

- Rappresentazione accurata del gioco da tavolo Monopoli
- Salvare e ripristinare una partita
- Interfaccia base

PRICE $200

SHOULD HAVE

- Interfaccia fedele
- Modalità multi dispositivo

PRICE $160

COULD HAVE

- Giocatori automatici (bot)
- Salvataggio automatico

PRICE $140

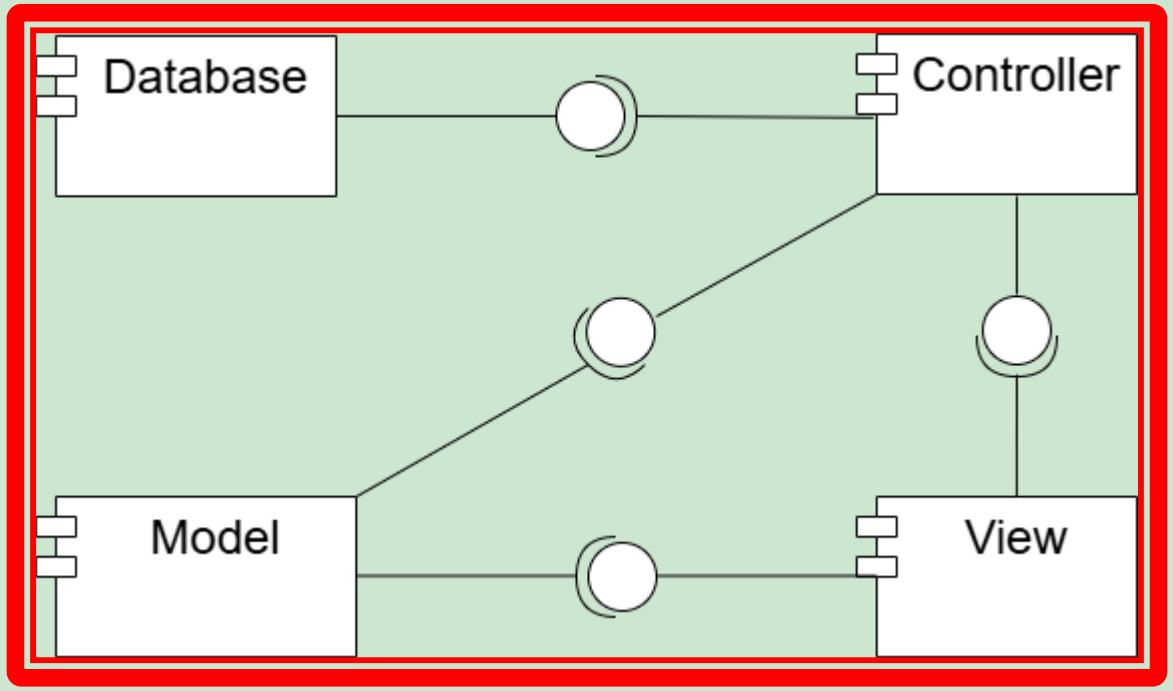
WON'T HAVE

- La versione mobile per telefono
- Le modalità alternative
- Statistiche di gioco

PRICE $60

ARCHITETTURA

Abbiamo deciso di implementare l'app applicando il modello di architettura **Model View Controller**



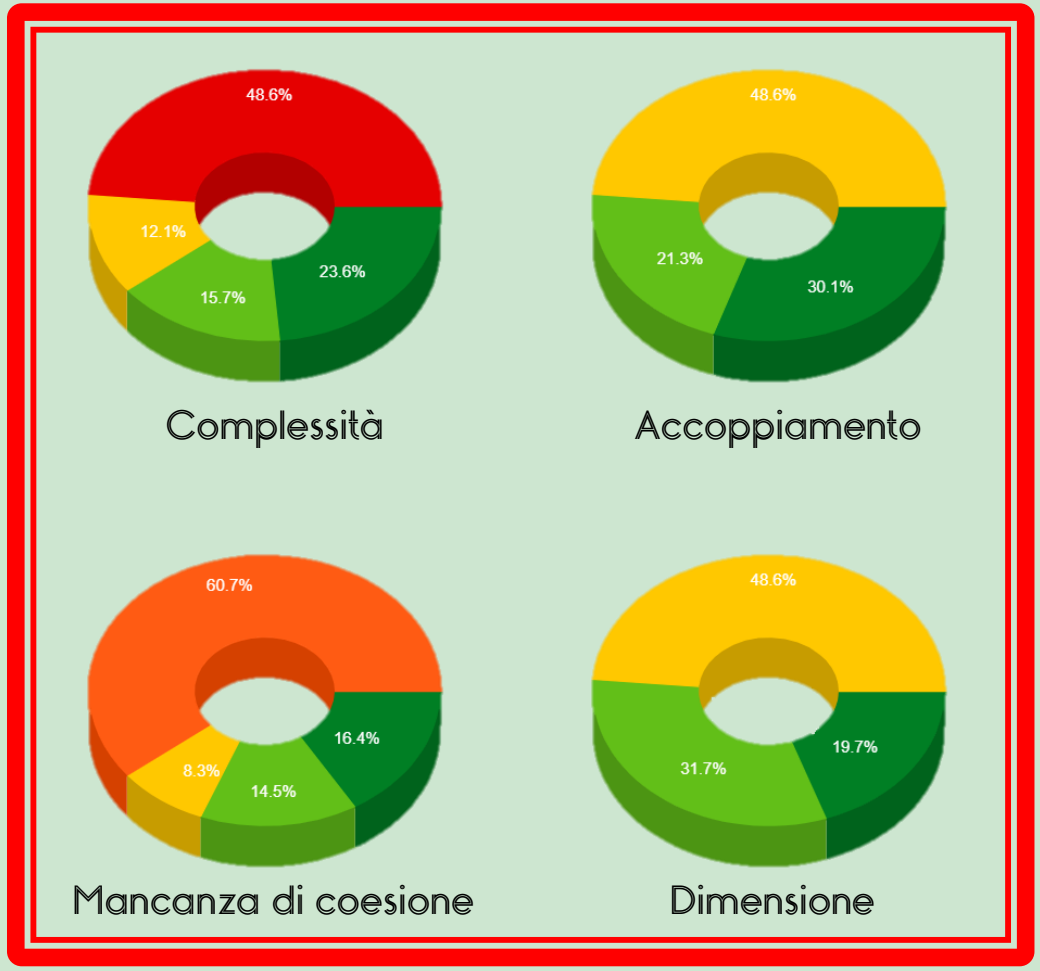
DESIGN PATTERN

Singleton pattern: il pattern per cui abbiamo optato è il Singleton. E' stato applicato nei Controller e nelle View del menu iniziale per evitare che ne venissero istanziati più d'uno, e in modo da avere un programma più stabile e fluido.

Abstract Occurrence: la classe Mazzo è stata resa *abstract* per la condivisione di alcuni campi e metodi tra MazzoImprevisti e MazzoProbablità.

Delegation pattern: per la gestione degli eventi legati ai bottoni di interfaccia abbiamo applicato questo pattern che sfrutta il riuso del codice dei Listener.

METRICHE DI QUALITA'



Element	Quality Attributes	LOC	Coupling	Complexity	Size	Lack of Cohesio
Monopoly						
Model		1066	low	low-medi...	medium-...	low
Monopoly		518	medium-...	very-high	medium-...	high
arrivoCasella( ): void		52	medium-...	low-medi...	medium-...	low
azioneCarta(Carta): void		100	medium-...	very-high	medium-...	low
caricamento(MonopolyGI		35	medium-...	medium-...	low-medi...	low
Monopoly(int, String, Mo		13	low-medi...	low	low	low
compraProprieta( ): void		12	low-medi...	low	low	low
costruisci(Proprieta): voi		39	low-medi...	low-medi...	low-medi...	low
demolisci(Proprieta): voi		29	low-medi...	low	low	low
ipoteca(Proprieta): void		22	low-medi...	low-medi...	low	low
aggiornaVisualizzazioneIr		19	low	low	low	low
astaBancarotta( ): void		4	low	low	low	low
attivitaPostBancarotta( ): v		3	low	low	low	low
controlloPassaggioVia( ): v		4	low	low	low	low
disipoteca(Proprieta): vo		8	low	low	low	low
getConta( ): boolean		7	low	low	low	low
getCorrispondenzaPlayer(		5	low	low	low	low
getCorrispondenzaProprie		5	low	low	low	low
getDadi( ): Dadi		2	low	low	low	low
getGiCorrente( ): Player		2	low	low	low	low
getGiocatoriString( ): Arra		5	low	low	low	low
getListaGiocatoriScambi(		7	low	low	low	low

Analisi del package Model

IMPLEMENTAZIONE

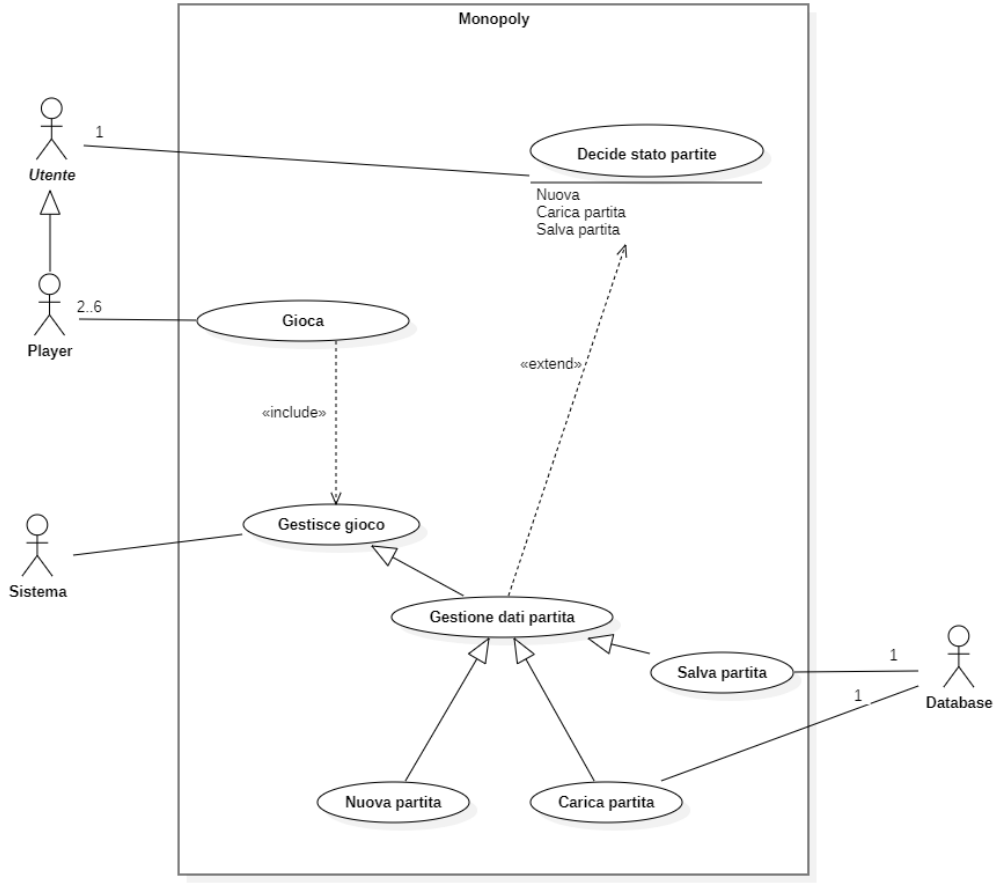
Rispetto ai requisiti indicati nel modello MoSCoW, abbiamo implementato:

- Rappresentazione accurata del gioco da tavolo Monopoly;
- Salvare e ripristinare una partita;
- Interfaccia fedele;
- Stabilità del software in funzione del numero di giocatori;

MODELLAZIONE

Grazie a StarUML abbiamo modellato i seguenti diagrammi UML, utili per la modellazione logica del progetto:

- Diagramma delle classi;
- Diagramma di stato;
- Diagramma di sequenza;
- Diagramma dei casi d'uso;
- Diagramma dei componenti;
- Diagramma delle attività.

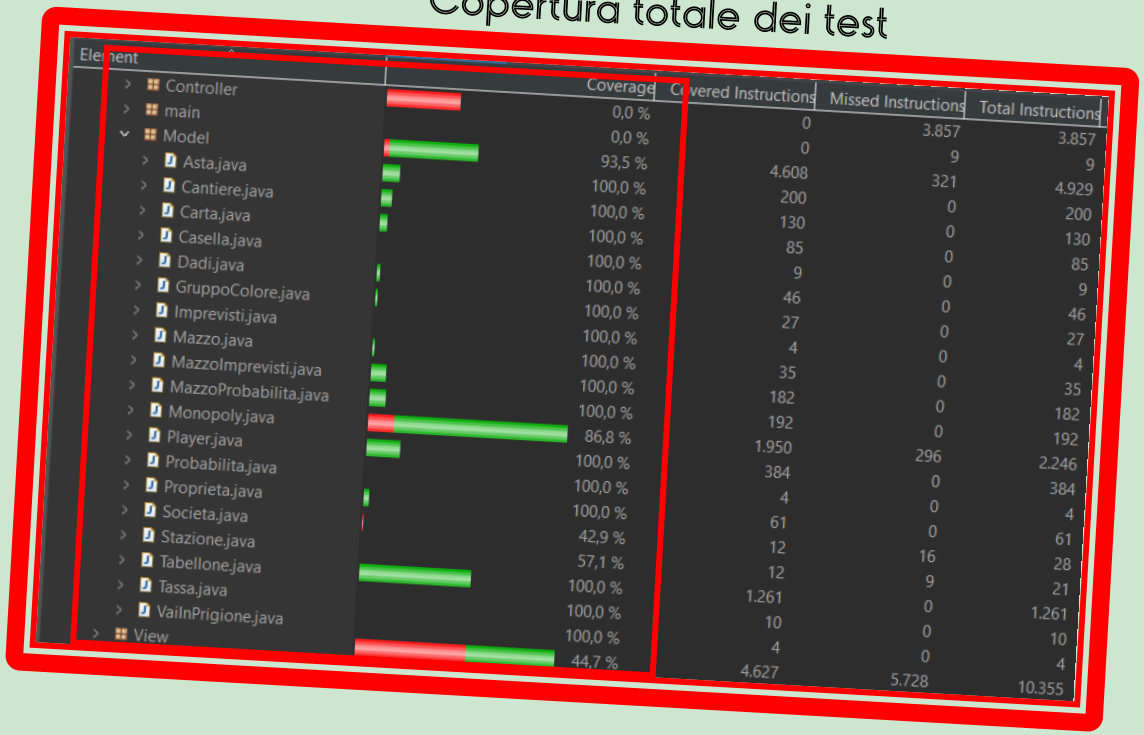


JUnit

TESTING

Attraverso i test in JUnit abbiamo potuto individuare alcune criticità nel nostro codice e di conseguenza migliorarlo, rendendolo più sicuro e rafforzandolo.

Copertura totale dei test



Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Controller	0,0 %	0	3.857	3.857
main	0,0 %	0	9	9
Model	93,5 %	4.608	321	4.929
Asta.java	100,0 %	200	0	200
Cantiere.java	100,0 %	130	0	130
Carta.java	100,0 %	85	0	85
Casella.java	100,0 %	9	0	9
Dadi.java	100,0 %	46	0	46
GruppoColore.java	100,0 %	27	0	27
Imprevisti.java	100,0 %	4	0	4
Mazzo.java	100,0 %	35	0	35
MazzoImprevisti.java	100,0 %	182	0	182
MazzoProbabilita.java	100,0 %	192	0	192
Monopoly.java	86,8 %	1.950	296	2.246
Player.java	100,0 %	384	0	384
Probabilita.java	100,0 %	4	0	4
Proprieta.java	100,0 %	61	0	61
Societa.java	42,9 %	12	16	28
Stazione.java	57,1 %	12	9	21
Tabellone.java	100,0 %	1.261	0	1.261
Tassa.java	100,0 %	10	0	10
VailnPrigione.java	100,0 %	4	0	4
View	44,7 %	4.627	5.728	10.355

MonopolyTest

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Model	77,8 %	3.834	1.095	4.929
Asta.java	25,0 %	50	150	200
Cantiere.java	69,2 %	90	40	130
Carta.java	81,2 %	69	16	85
Casella.java	100,0 %	9	0	9
Dadi.java	100,0 %	46	0	46
GruppoColore.java	74,1 %	20	7	27
Imprevisti.java	100,0 %	4	0	4
Mazzo.java	82,9 %	29	6	35
MazzoImprevisti.java	100,0 %	182	0	182
MazzoProbabilita.java	100,0 %	192	0	192
Monopoly.java	73,4 %	1.648	598	2.246
Player.java	55,5 %	213	171	384
Probabilita.java	100,0 %	4	0	4
Proprieta.java	90,2 %	55	6	61
Societa.java	42,9 %	12	16	28
Stazione.java	57,1 %	12	9	21
Tabellone.java	94,0 %	1.185	76	1.261
Tassa.java	100,0 %	10	0	10
VailnPrigione.java	100,0 %	4	0	4

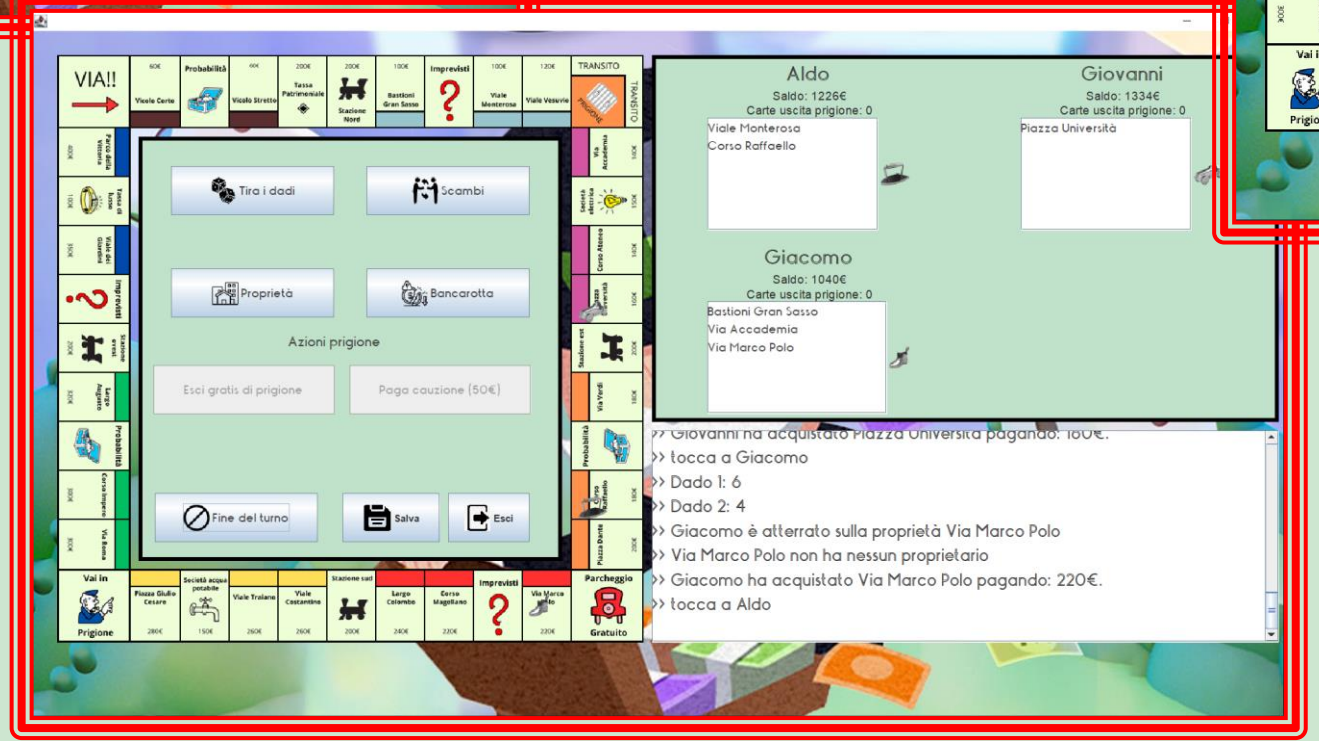
MazzoECarteTest

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Model	10,0 %	494	4.435	4.929
Asta.java	0,0 %	0	200	200
Cantiere.java	0,0 %	0	130	130
Carta.java	100,0 %	85	0	85
Casella.java	0,0 %	0	9	9
Dadi.java	0,0 %	0	46	46
GruppoColore.java	0,0 %	0	27	27
Imprevisti.java	0,0 %	0	4	4
Mazzo.java	100,0 %	35	0	35
MazzoImprevisti.java	100,0 %	182	0	182
MazzoProbabilita.java	100,0 %	192	0	192

DEMO


MONOPOLY
Nuova partita
Carica partita
Esci


Salvataggio
Inserisci il nome del salvataggio
Salvataggio_1
Salva
Annulla


Tira i dadi
Scambi
Proprietà
Bancarotta
Azioni prigione
Esci gratis di prigione
Paga cauzione (50€)
Fine del turno
Salva
Esci
Aldo
Saldo: 1226€
Carte uscita prigione: 0
Viale Monterosa
Corso Raffaello
Giovanni
Saldo: 1334€
Carte uscita prigione: 0
Piazza Università
Giacomo
Saldo: 1040€
Carte uscita prigione: 0
Bastioni Gran Sasso
Via Accademia
Via Marco Polo
>> tocca a Giacomo
>> Dado 1: 6
>> Dado 2: 4
>> Giacomo è atterrato sulla proprietà Via Marco Polo
>> Via Marco Polo non ha nessun proprietario
>> Giacomo ha acquistato Via Marco Polo pagando: 220€.
>> tocca a Aldo

GRAZIE PER L’ATTENZIONE