

SUPSI

Mine Sweeper project

Gruppo-05

Studente/i	Relatore	Correlatore
Nicholas Gherbi Lorenzo Figatti Andrea Perlini		
Corso di laurea	Modulo / Codice Progetto	Anno
Ingegneria Informatica	Ingegneria del software I	2024/2025
Committente	Data	
Giancarlo Corti Tommaso Agnola	02/06/2025	

Indice

1. Contesto e motivazione
2. Problemi
3. Stato dell'arte
4. Approccio
5. Risultati raggiunti
6. Conclusioni

Contesto e motivazione

Questo progetto nasce all'interno del corso di Ingegneria del Software, con l'obiettivo di applicare i principi di progettazione e sviluppo appresi durante il corso.

La creazione del gioco *Minesweeper* ci ha permesso di mettere in pratica la gestione del ciclo di vita del software, dalla raccolta dei requisiti fino all'implementazione.



Problemi/requisiti

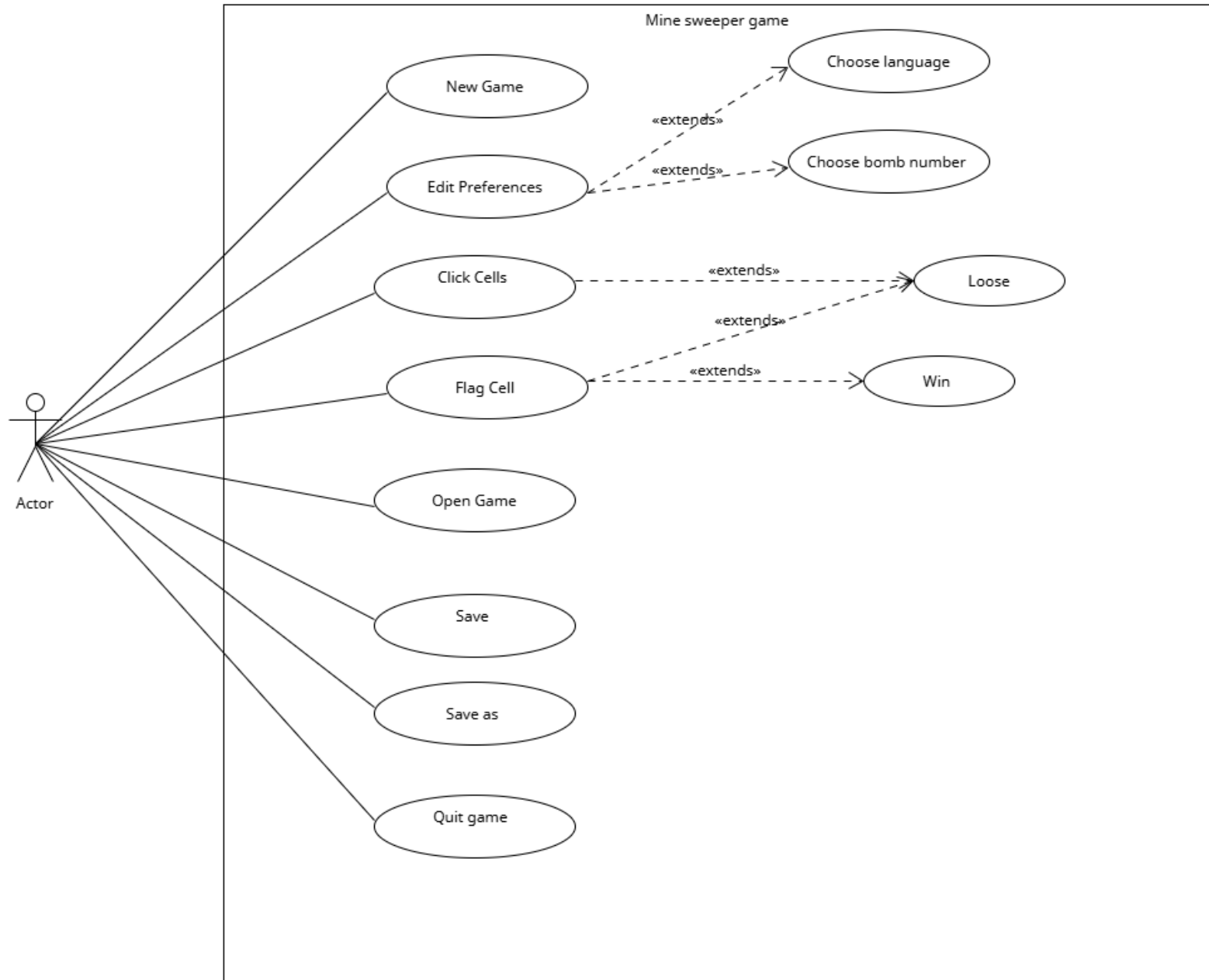
Il problema che ci è stato assegnato consiste nello sviluppo di un programma del gioco **MineSweeper**.

L'applicazione doveva soddisfare i seguenti requisiti funzionali:

- 1) **Chiara separazione tra le componenti.**
- 2) **Visualizzazione di feedback all'utente.**
- 3) **Configurazione del numero di bombe**, modificabile dall'utente.
- 4) **Supporto multilingua**, con possibilità di cambiare la lingua tramite interfaccia.
- 5) Funzionalità **di salvataggio e caricamento della partita**.
- 6) Realizzazione **applicazione stand-alone**, eseguibile da terminale.






Use Case Diagram



Stato dell'arte

Negli anni sono state create numerose versioni e reinterpretazioni del gioco:

-  **Versioni desktop:** disponibili sui vari sistemi operativi, hanno grafiche semplici ma efficaci.
-  **Versioni web:** realizzate con HTML5 e JavaScript, giocabili direttamente dal browser e sono quelli più utilizzati.
-  **App mobili:** presenti su Android e iOS, spesso con funzionalità aggiuntive come classifiche online, livelli personalizzati e interfacce più moderne.

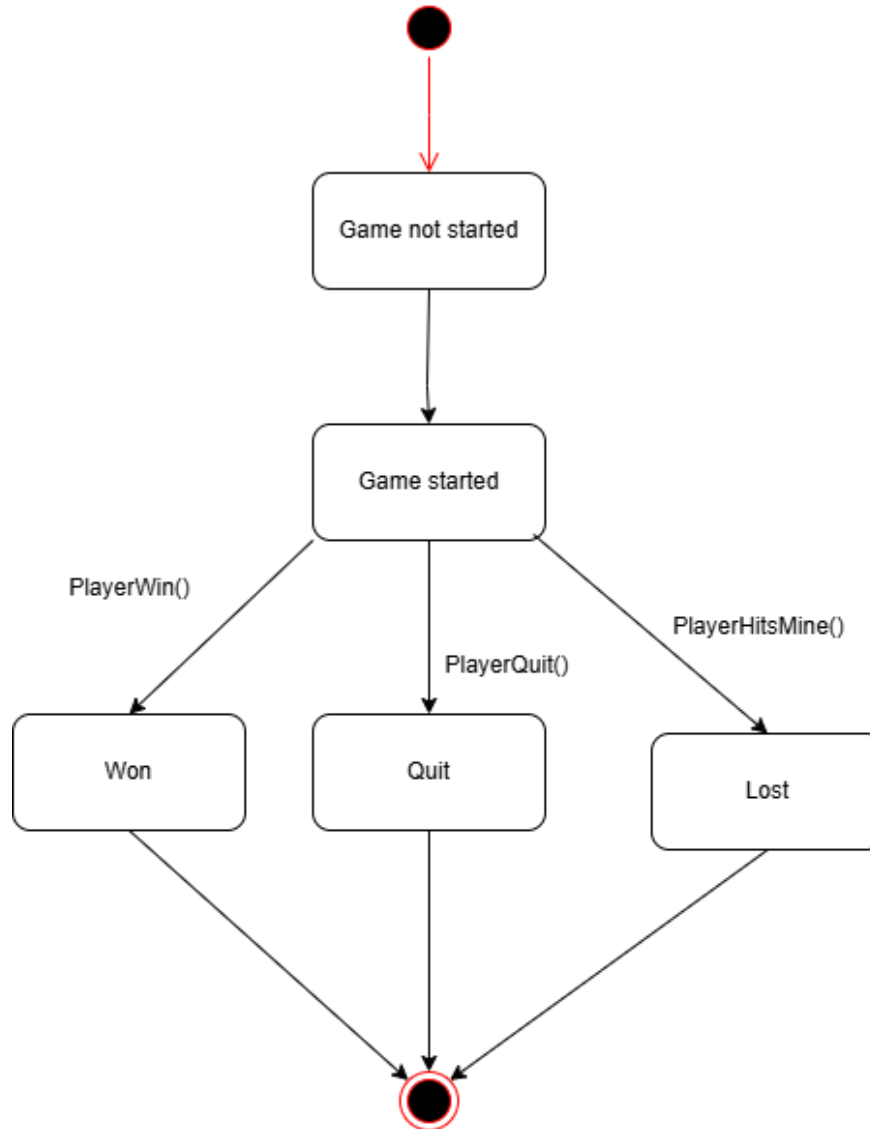
La nostra versione implementa il salvataggio e la possibilità di ricaricare la partita, cosa che non è disponibile in molte delle soluzioni trovate online.



Es. Minesweeper Google



State machine diagram

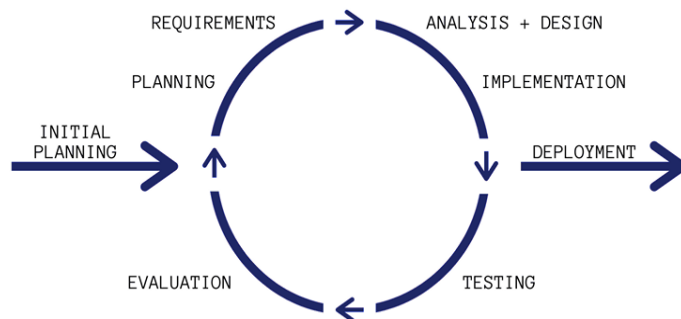


Approccio

1. Stack Tecnologico
Linguaggio: Java 17
Grafica: JavaFx
Dipendenze: Jackson



2. Approccio di sviluppo:
Come approccio di sviluppo abbiamo scelto: Iterativo Incrementale.



Approccio

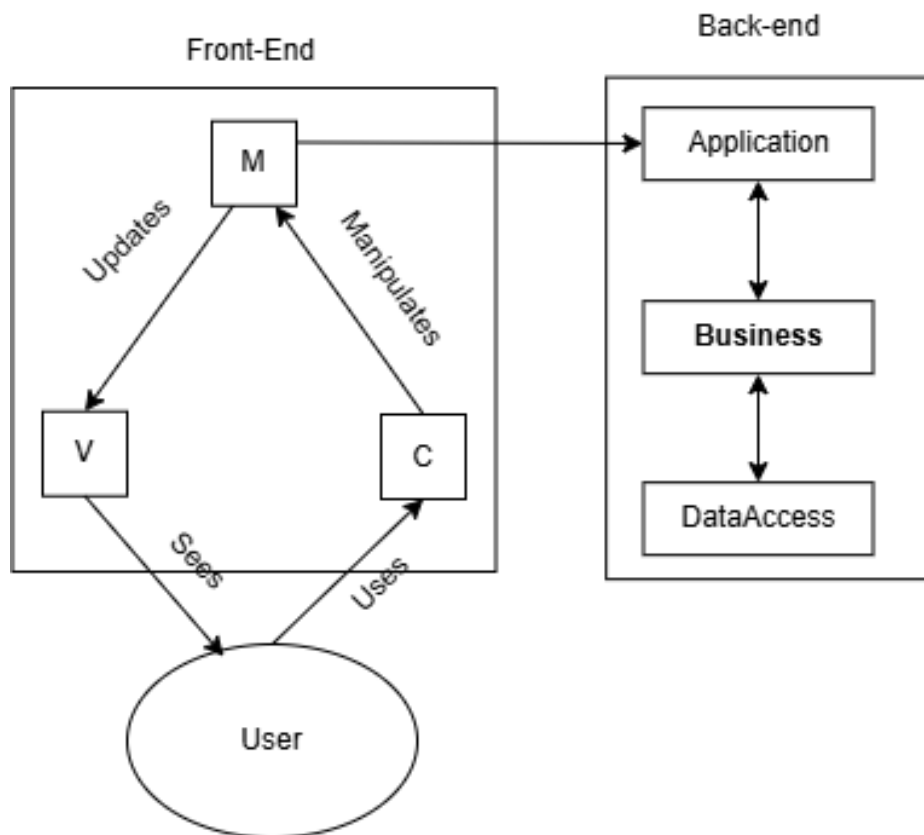
3. Architettura- **divisione front-end e back-end**

- il **Frontend** utilizza l'architettura MVC.
- il **Backend** utilizza l'architettura a layer.

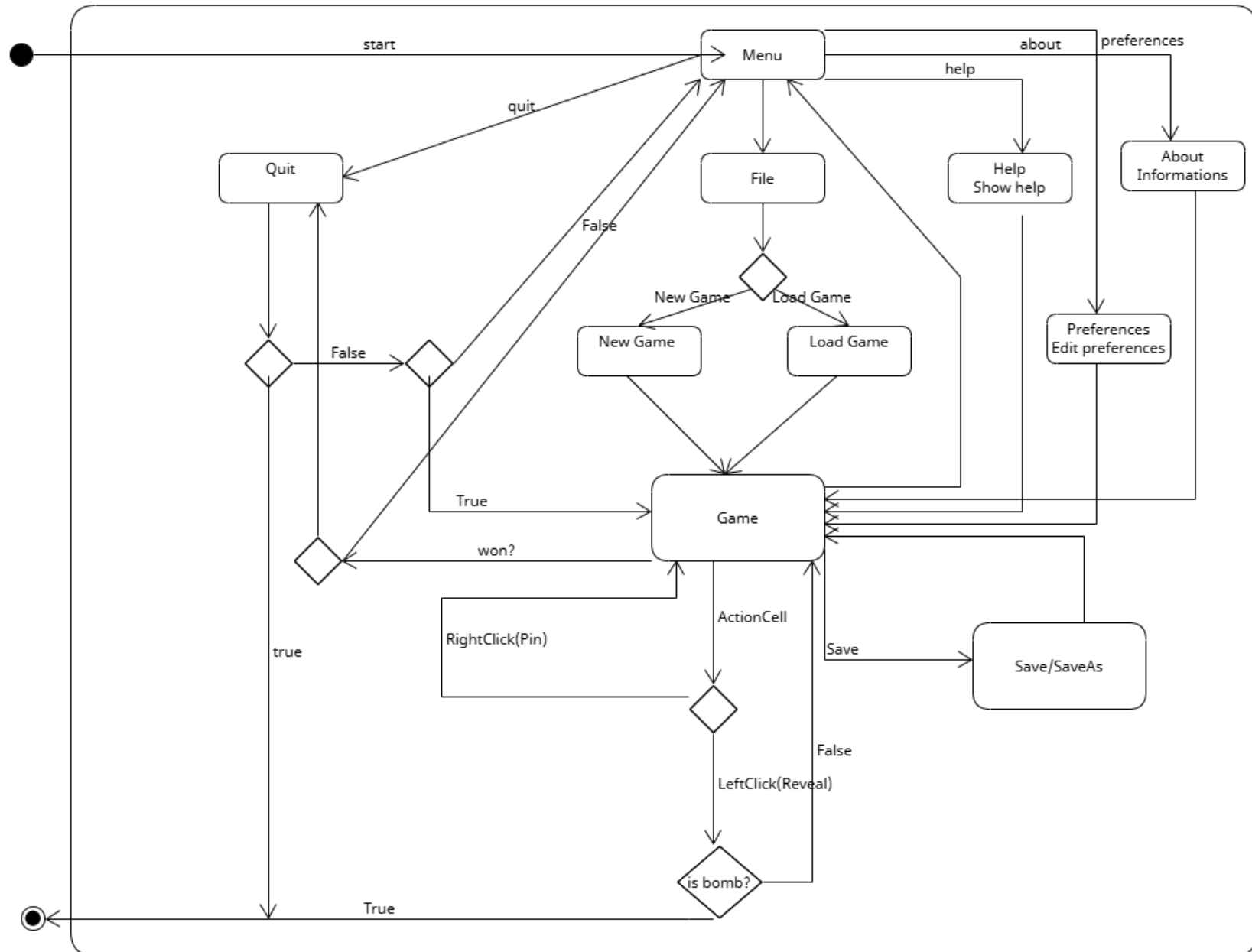
4. Distribuzione ed Esecuzione

Artifact: JAR (con dipendenze incluse) su Maven.

Esecuzione: JAR eseguibile con `java -jar`, compatibile con tutte le piattaforme.



Activity Diagram



Risultati Raggiunti

- Possibilità di salvare la partita e ricaricarla
- Cambio della lingua da GUI e file.
- Selezione del numero di bombe da GUI e file.
- Menù completo: Nuova Partita, Apri, Esci, Help.
- Utilizzare l'applicazione in modo stand alone.

DEMO

Conclusioni

Cosa abbiamo imparato

- Abbiamo messo in pratica concetti fondamentali di **ingegneria del software**: gestione dei requisiti, design, sviluppo e testing.
- Abbiamo collaborato efficacemente in team, anche nella **divisione dei compiti e nell'integrazione del codice**.

Difficoltà incontrate

- La suddivisione in due moduli partendo da un modulo unico.
- Rispettare l'architettura stabilita