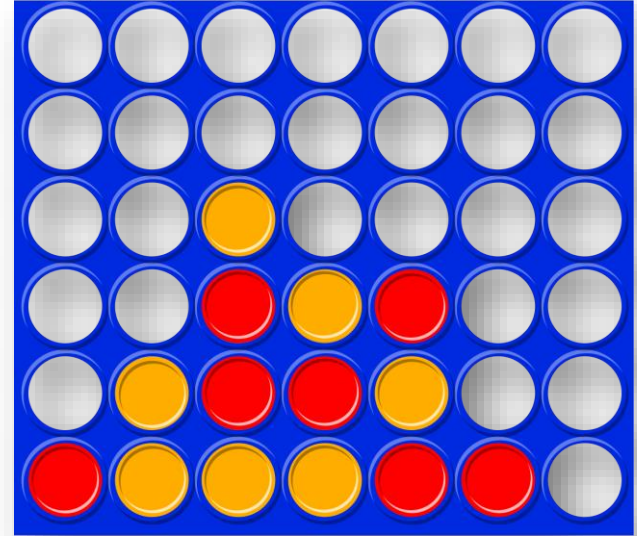


SUPSI

Forza 4

Main software development
project



Studenti

Alessandro Spagnuolo

Juan Pallares Da Cruz

Mattia Gerber

Corso di laurea

Ingegneria Informatica

Modulo

Ingegneria e sviluppo software 1

Anno

2023/2024

Committenti

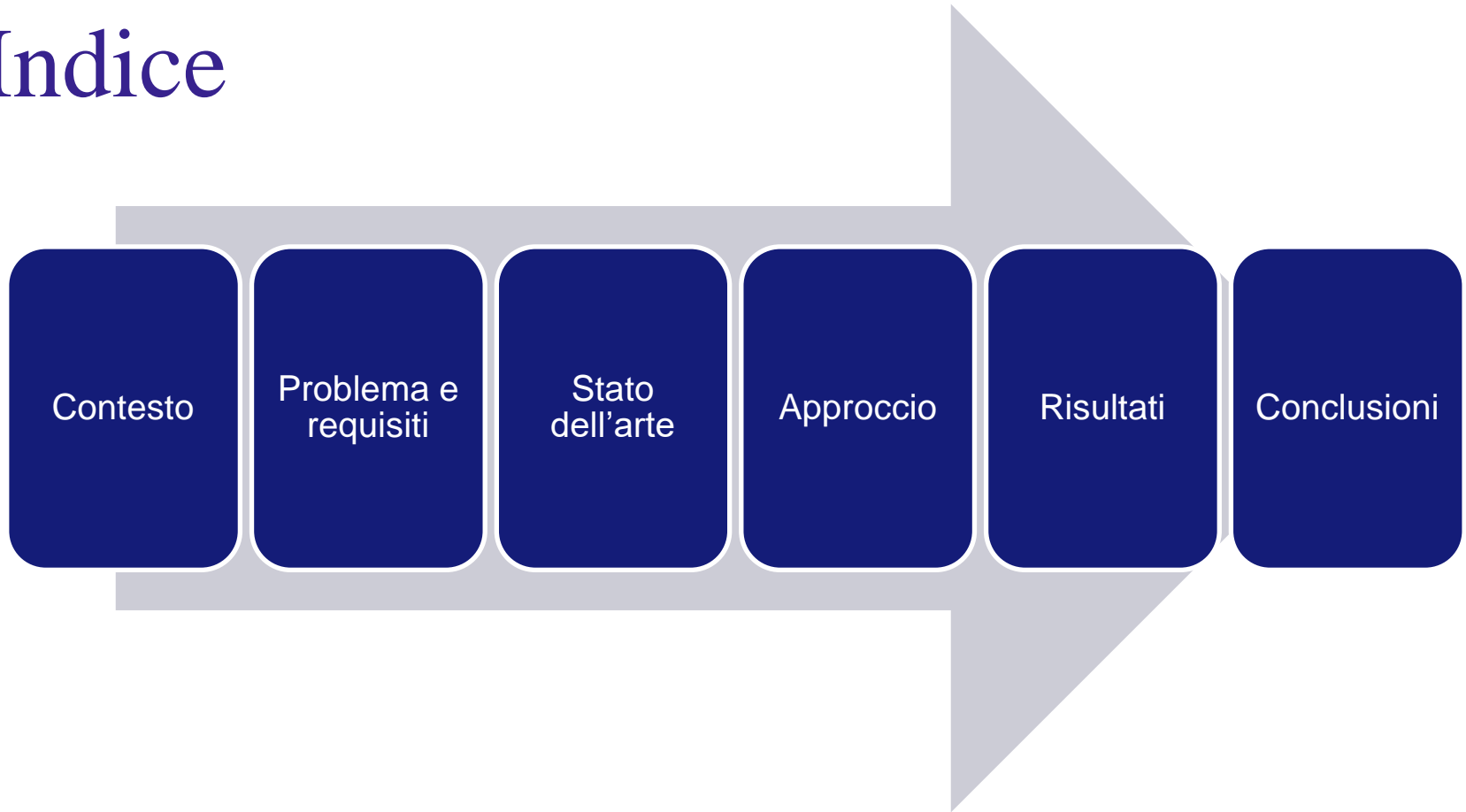
Prof. Giancarlo Corti

Prof. Tommaso Agnola

Data

04.06.2024

Indice



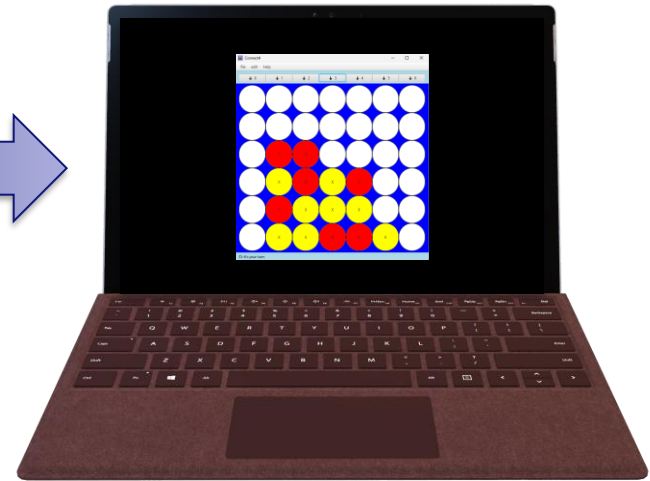
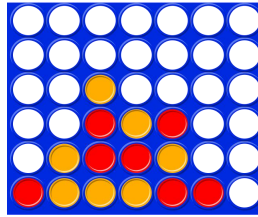
Contesto

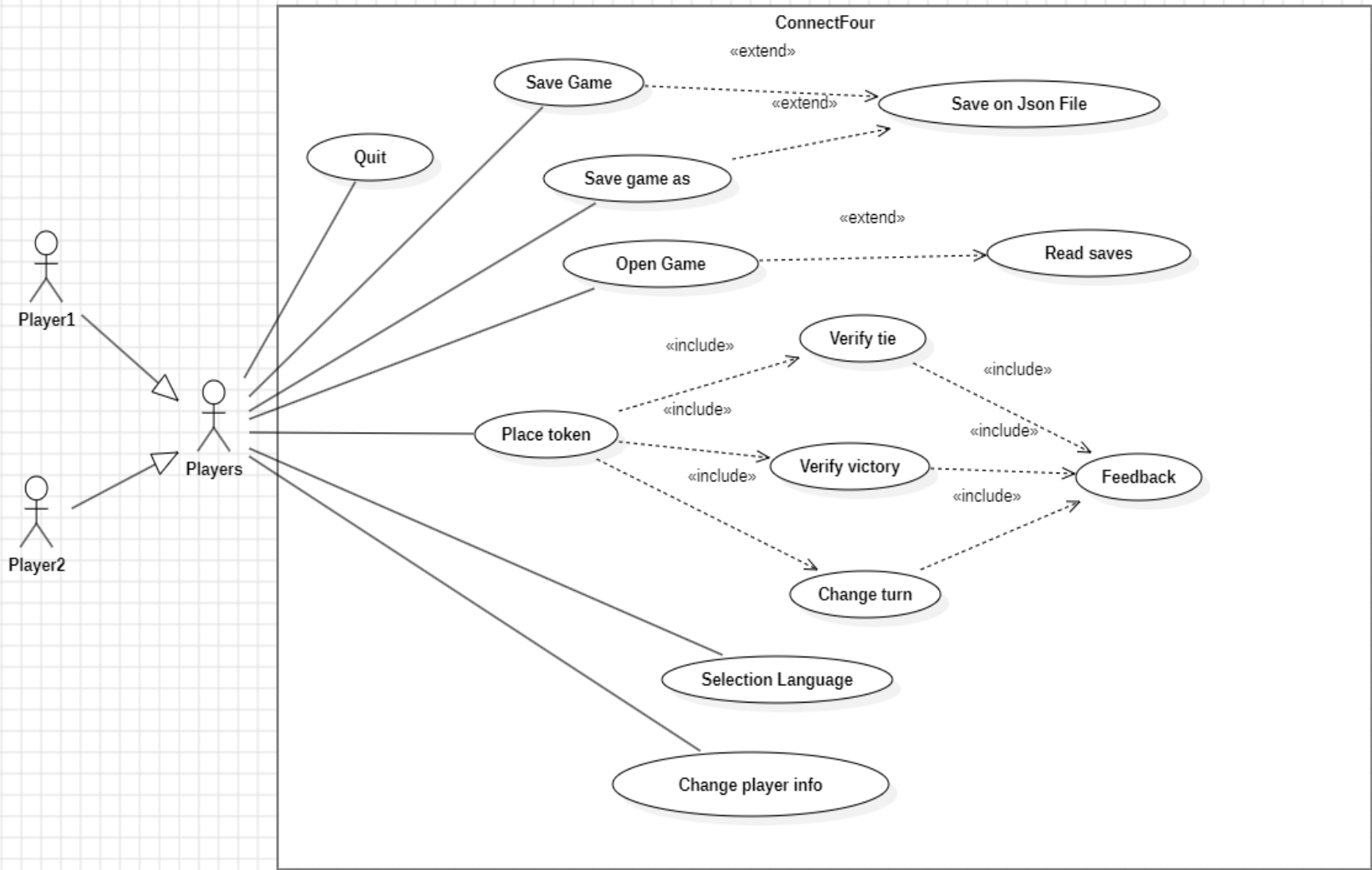
- **Progetto di gruppo** valutato per il corso di **Ingegneria e Sviluppo Software 1**
- Familiarizzare con i **processi di sviluppo e design** di un Software
- Realizzazione di un'**applicazione grafica** per il gioco Forza 4



Problema

- Creare un'esperienza di gioco **Forza 4** su desktop intuitiva e personalizzabile
- Segue **use case diagram**





Requisiti

Requisiti funzionali:

- Gestione dei turni
- Gestione delle preferenze dei giocatori
- Avvio, salvataggio e apertura di una partita

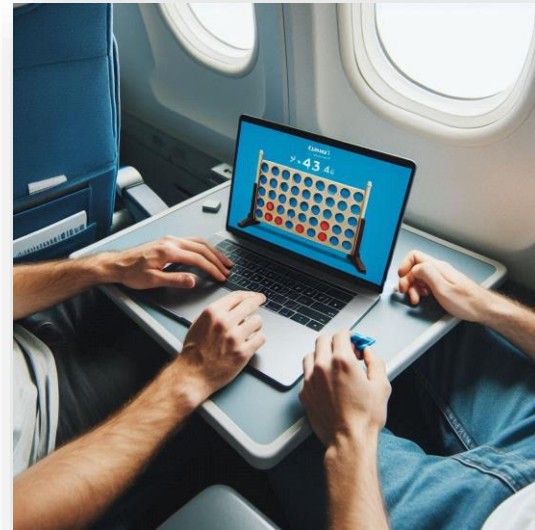
Requisiti non funzionali:

- Java e JavaFX 17
- Suddivisione backend – frontend
- Internazionalizzazione
- Maven
- GitLab



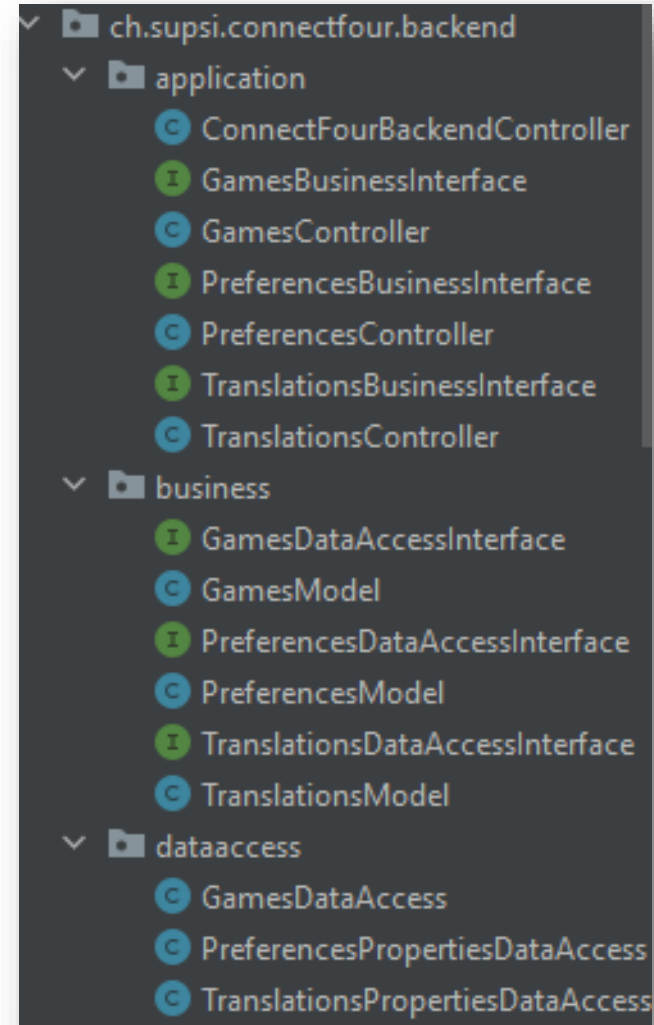
Stato dell'arte

- Gioco fisico → **occupa spazio**
- Simulatori online → serve connessione **internet**
- Applicazione JavaFX → in locale, **ovunque**
- ...



Approccio

- **Backend → Multi-Layer**
persistenza partite, gestione lingue, traduzioni
- Dipendenze
→ org.json



Approccio

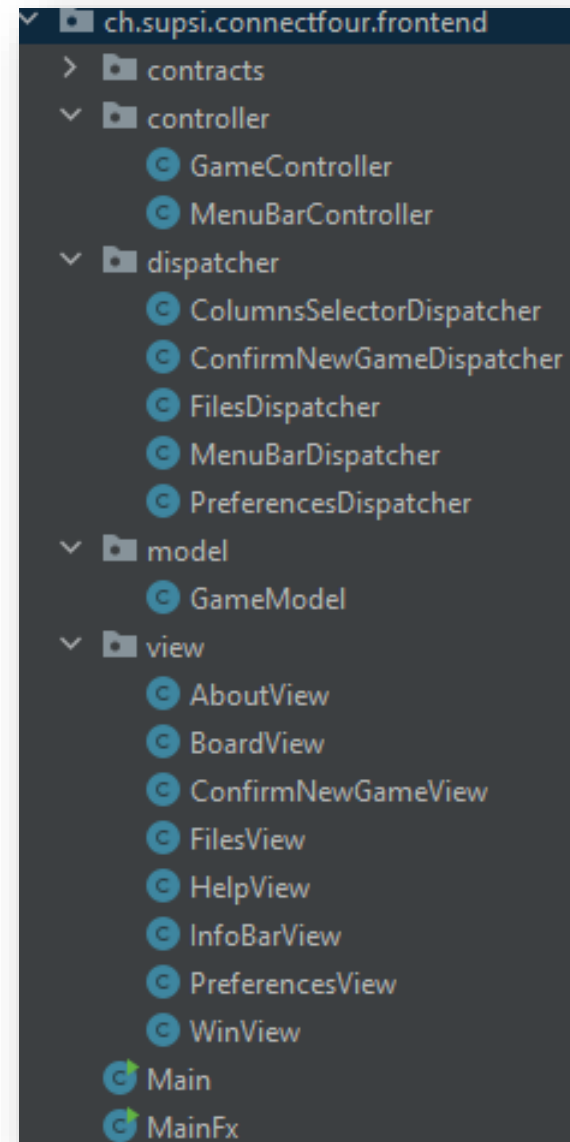
- **Frontend → MVC**

logica di gioco, interfaccia grafica,
interazioni tra le due e con il backend

- Dipendenze

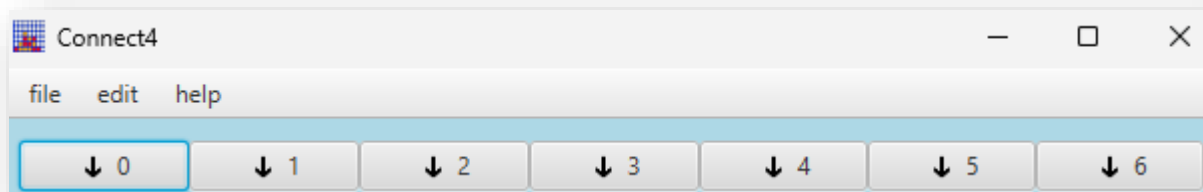
→ org.openjfx

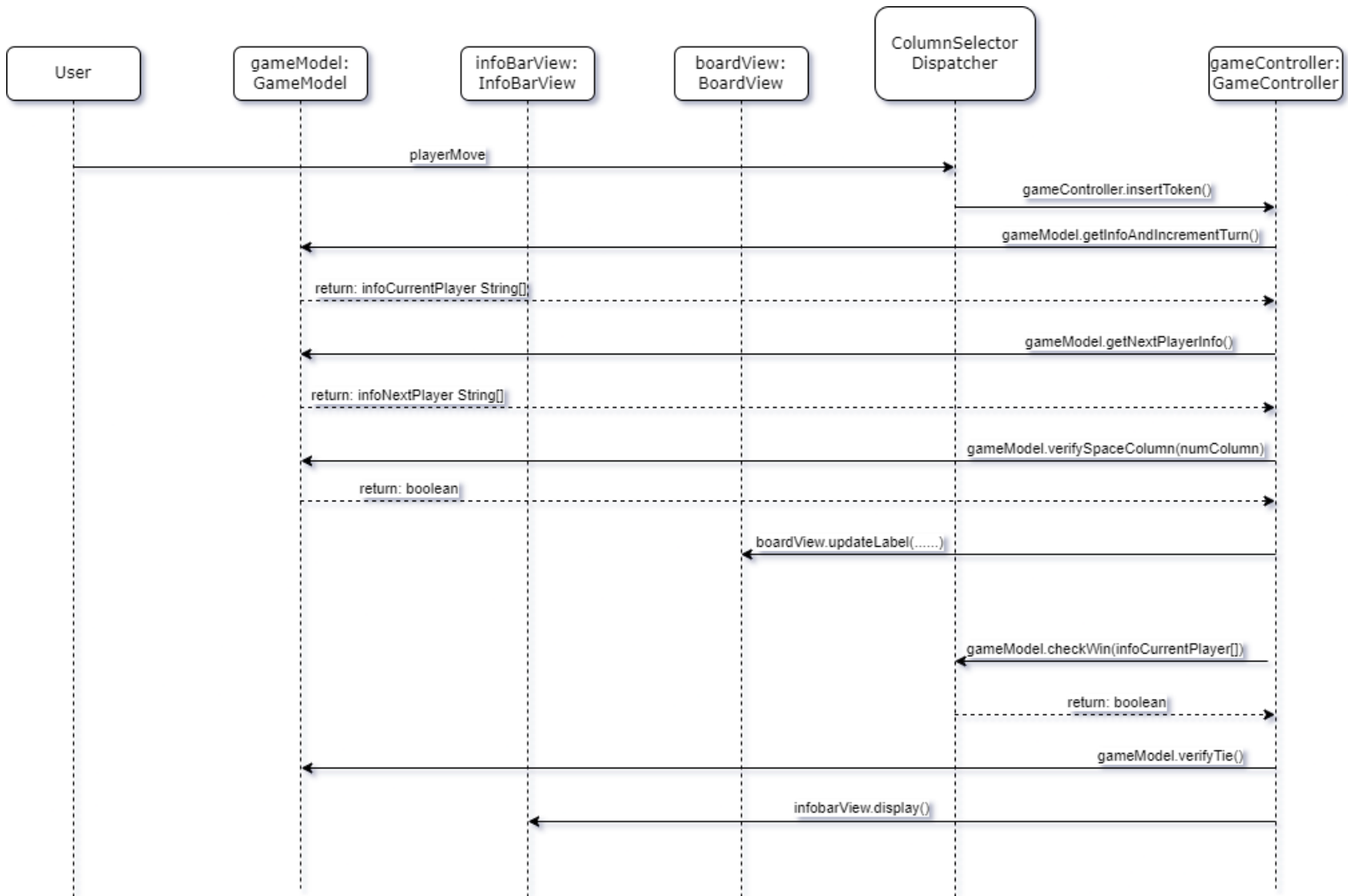
→ backend



UML sequence diagram 1 - pedina

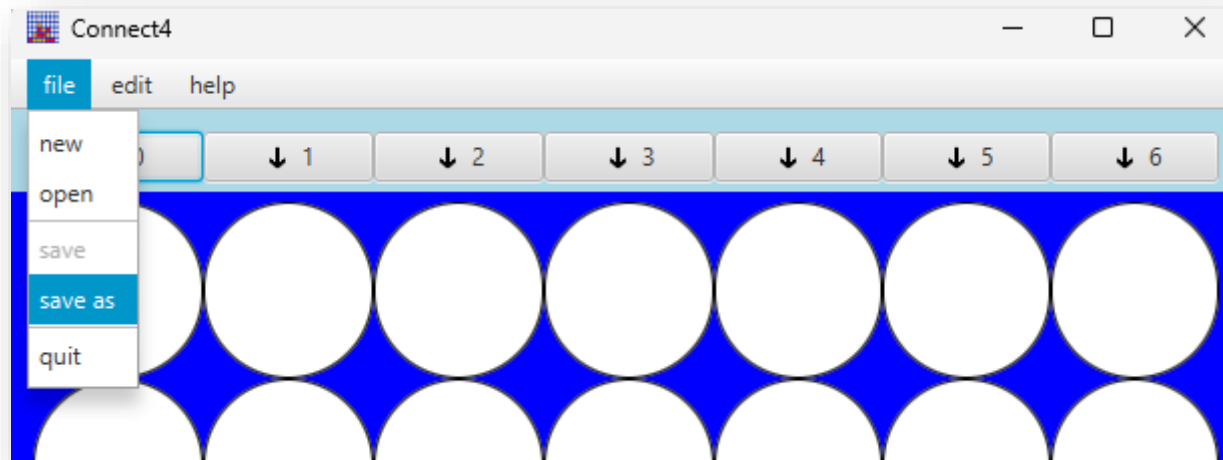
- **Come avviene l'inserimento di una pedina?**
- Come interagiscono tra di loro i vari componenti del frontend?

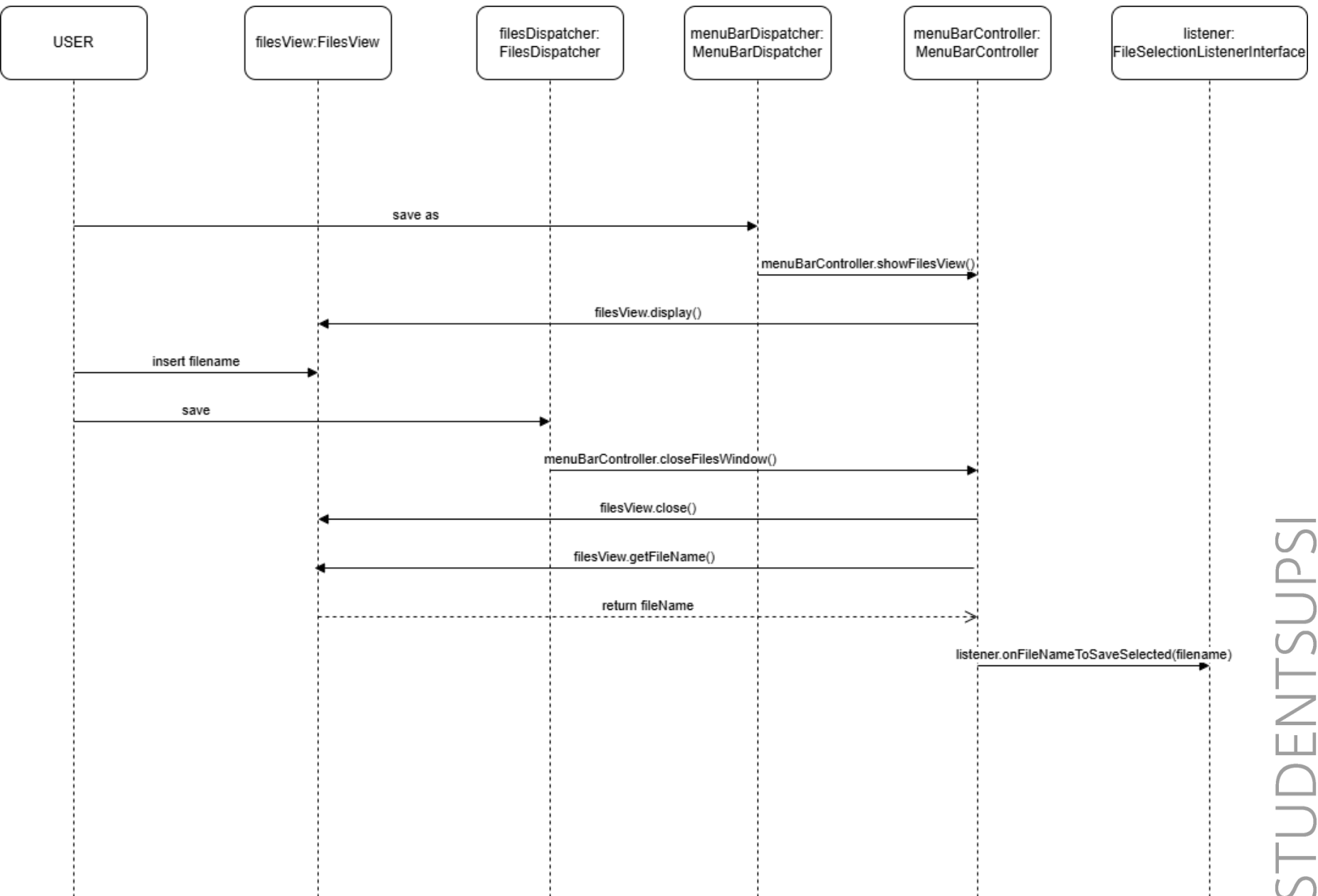




UML sequence diagram 2 - salvataggio

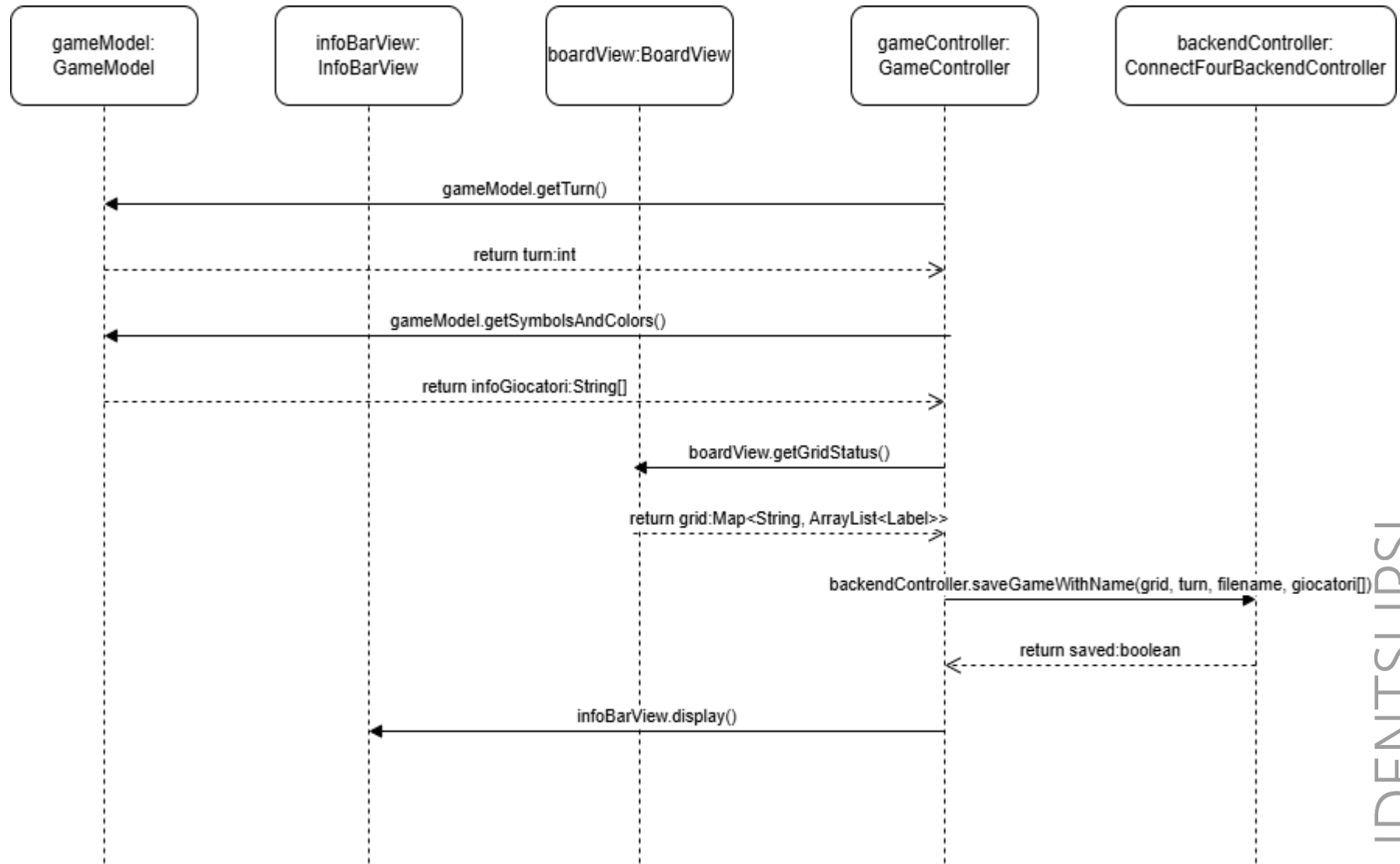
- **GameController** è un `FileSelectionListener` **observer** di **MenuBarController**
- Osserva quando viene scelto il **nome del file** con il quale **salvare la partita**
- Stesso procedimento per l'**apertura** di una partita





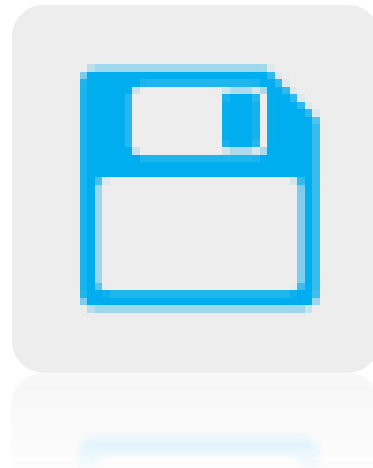
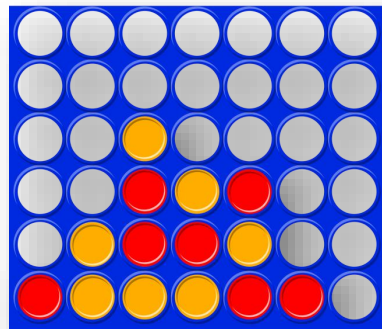
UML sequence diagram 3 - salvataggio

- **GameController** recupera le informazioni da salvare
→ poi interagisce con il **ConnectFourBackendController**



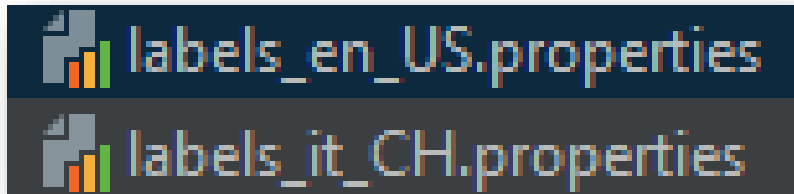
Salvataggio partita - backend

- I file delle **partite** vengono messi nella **home** directory dell'utente
→ **.connectFour\games**
- **Contenuto**
→ file **json** contenente la griglia di gioco e le info dei giocatori



Internazionalizzazione

- Utilizzo di file **.properties** nel **backend** contenente le **traduzioni**

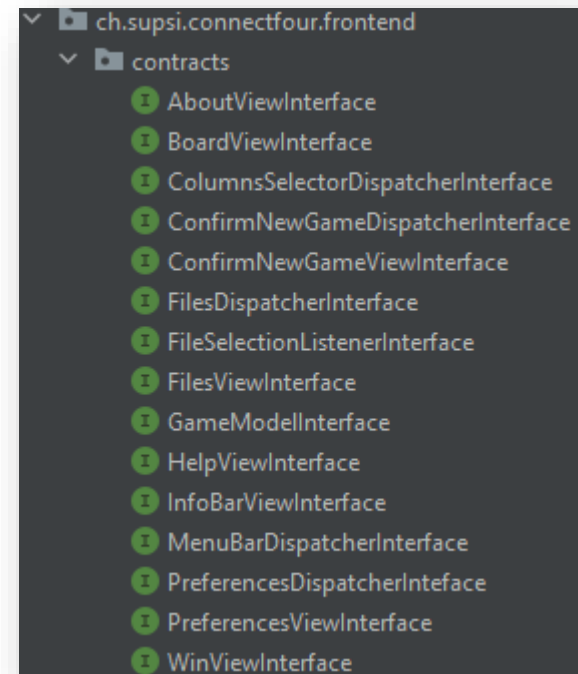


```
label.chooseFileName=scegli un nome per il file
label.textFile = inserisci il nome del file...
label.gameSaved=la partita \u00E8 stata salvata
label.gameNotSaved=la partita non \u00E8 stata salvata
```

```
label.welcome =Welcome to connect4
label.saveas=save as
label.save=save
label.open=open
label.new=new
label.about=about
label.help=help
label.quit=quit
label.edit=edit
label.preferences=change symbols & colors
label.changeLanguage=change language
```

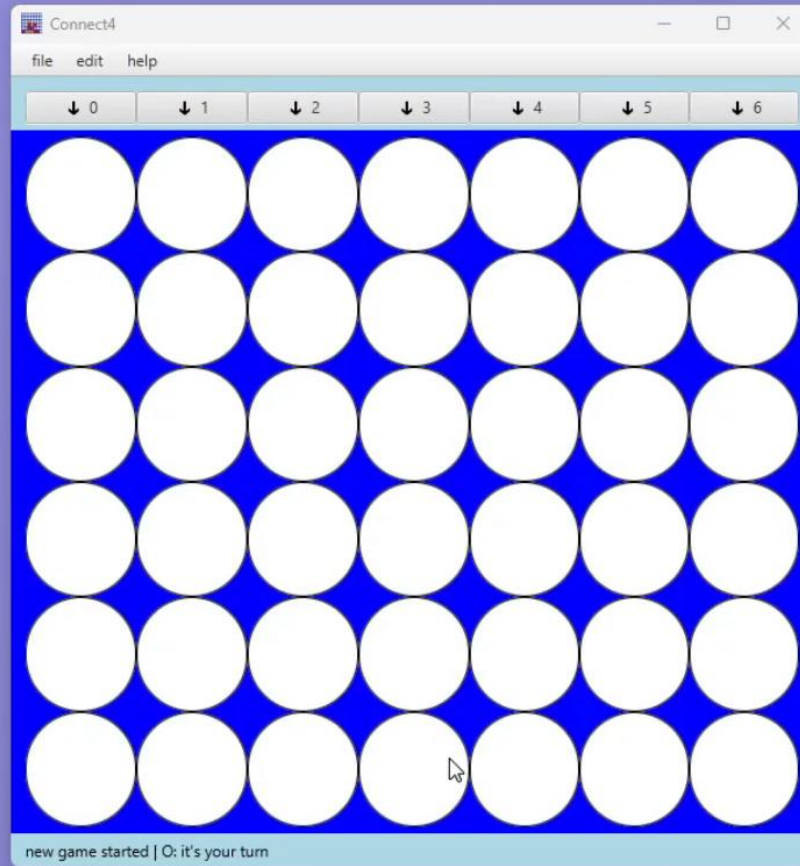
Contracts = meno coupling

- Esempio:
GameController stipula un contratto con l'interfaccia **GameModelInterface** che mette a disposizione i **metodi necessari** che sono poi **implementati dalla classe GameModel**



Risultati

- Demo



Conclusioni

- Applicazione **pratica** dei concetti teorici
- Competenze nello sviluppo di **applicazioni desktop**
- Principi **SOLID**, architettura **MVC** e **Multi-Layer**, ...
- Lavoro di **squadra**
- **Ciclo di vita** del software
- Sviluppi futuri