

# **Progetto di Ingegneria del Software**

**Presentazione gruppo IS23-AM41**

**D'Alessio - De Ciechi - Deidier - Ermacora**

**A.A.: 2022/2023**

# ***Design Patterns***

- *Model-View-Controller* - gestione della logica di gioco e interfacce utente.
- *AbstractFactory* - board relativa a 2/3/4 giocatori.
- *State* - gestione dello stato del server e delle operazioni possibili.
- *Singleton* - oggetti instanziabili una sola volta (es.: *borsa oggetti*, *token fine partita*).

# ***Comunicazione Client-Server***

- *Transmission Control Protocol*

- Sia client che server implementano tre layer fondamentali:
  - **Socket layer** - gestione della connessione socket (server implementa *SocketManager* per gestire più connessioni socket verso i giocatori).
  - **Serialize/Deserialize layer** - conversione da stringa di byte a messaggio TCP e viceversa.
  - **TCP controller layer** - invocazione dei metodi del controller sulla base dell'*header* e *body* del messaggio TCP e viceversa.

- *Remote Method Invocation*

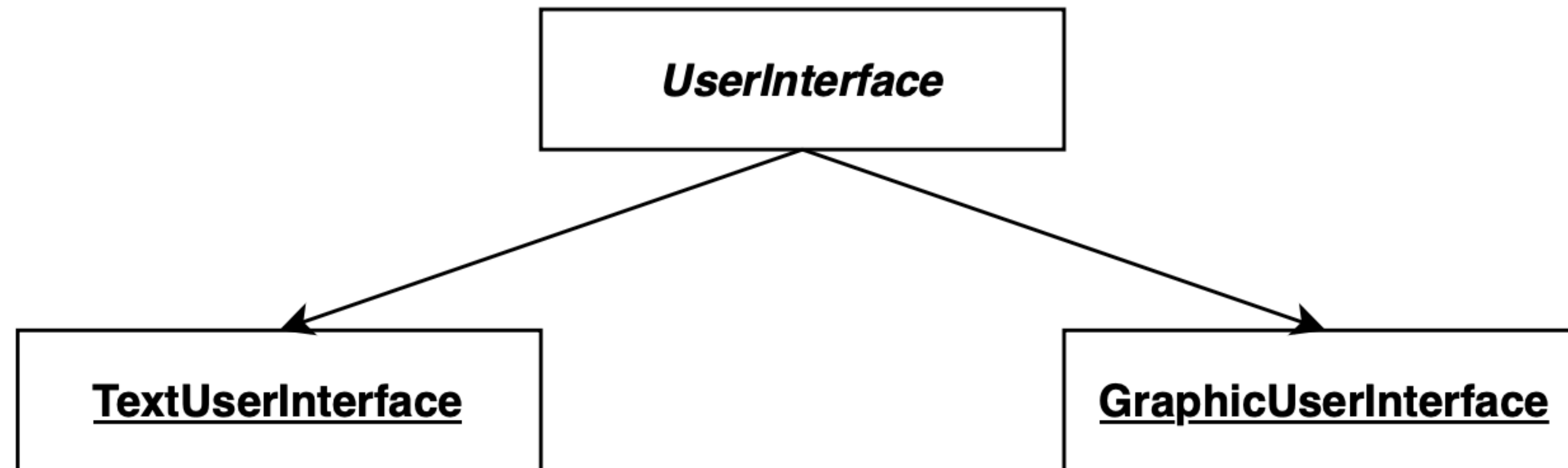
- Sia client che server espongono le proprie interfacce con i metodi invocabili da remoto.

# ***Gestione delle connessioni***

- *Connessioni server:*
  - Aperto ad entrambe le tipologie di connessioni su due porte differenti.
- *Connessioni client:*
  - L'utente sceglie che tipologie di connessione instaurare con il server.
  - Interfaccia Connection:
    - Espone i metodi *startConnection()* e *closeConnection()*.
    - A seconda della scelta dell'utente **dinamicamente** viene istanziata come una **ConnectionRMI** o **ConnectionTCP**.

# ***User Interface***

- Anche in questo caso è presente una *interfaccia* *UserInterface*:
  - Espone tutti i metodi per gestire eventi e cambiamenti nella vista.
  - **Dinamicamente** istanziata come **TextUserInterface** o **GraphicUserInterface**.



# ***Funzionalità aggiuntive***

- *Chat:*
  - Si appoggia ai protocolli di comunicazione.
  - Permette messaggi **broadcast** e messaggi **peer-to-peer**.
- *Persistenza del server:*
  - Implementata mediante un file di salvataggio (.json).
  - **Controller** lato server **serializzato** con annesso stato della partita.
- *Resilienza alle disconnessioni:*
  - Implementato tramite un sistema di ping-pong asincrono.
  - Inoltre, al client è permesso di **disconnettersi** e **riconnettersi volontariamente** sia durante la fase di lobby che durante la fase di gioco **senza interrompere** l'avanzamento della **partita**.

# **Grazie per la visione!**

**Gruppo IS23-AM41 - <https://github.com/SimoneDeidier/IS23-AM41>**

**D'Alessio Edoardo - De Ciechi Samuele - Deidier Simone - Ermacora Iacopo**