

Descrizione Architetture

Gruppo GC31

L'architettura sfruttata per il nostro applicativo è un'architettura distribuita di tipo client-server, in cui le connessioni vengono gestite mediante socket.

Andiamo di seguito a descrivere singolarmente l'architettura e le funzionalità delle classi lato server e lato client.

Server

La parte lato server si compone di 4 classi principali:

- **Server** = classe con main eseguibile che permette di creare e configurare il server. All'interno del server viene gestita la lobby prepartita e la generazione dei clientID, per poi istanziare il GameHandler nel controller.
- **ServerSocketHandler** = classe istanziata dal server che permette di gestire le connessioni, quindi di accettare e aprire socket con i vari client.
- **SocketClientConnection** = classe che, una volta istanziata una connessione, permette di scambiare messaggi con i vari client.
- **VirtualClientView** = classe ausiliaria lato server che, per ogni utente, racchiude le informazioni principali (id, nickname, connessione) in modo tale da poterle interrogare quando necessario.

Client

La parte lato client si compone di 4 classi principali:

- **CLI** = classe con main eseguibile che genera un client e cerca di aprire una connessione socket con il server. Gestisce poi la stampa di messaggi provenienti dal server.
- **ClientConnection** = classe che, una volta instaurata una connessione, gestisce l'invio di messaggi da client a server.
- **ServerListener** = classe che, fin quando la connessione è aperta, ascolta risposte dal server e le inoltra all'ActionHandler.
- **ActionHandler** = classe che gestisce i vari tipi di risposte arrivate dal server.

Communication Protocol

Vi lasciamo anche una sintesi del nostro protocollo di comunicazione.

Per lo scambio di messaggi abbiamo deciso di sfruttare Serializable, generando le classi Answer (Server) e Message (Client) che la estendono e le relative classi SerializedAnswer e SerializedMessage che la implementano in modo tale da inviare tali messaggi attraverso la rete.

Le uniche interfacce che gestiscono tali messaggi sono:

- **SocketClientConnection** (lato server)
- **ClientConnection** e **ServerListener** (lato client)

In queste classi viene gestita la serializzazione e de-serializzazione dei messaggi.