

Protocollo di comunicazione tra client e server

Progetto Ingegneria del Software - a.a. 2021/22 Politecnico di Milano

Il protocollo di comunicazione si basa sulla serializzazione di Java e sul pattern strategy.

I messaggi sono "atomici", ovvero ogni azione che il giocatore può compiere durante la partita è gestita tramite un unico messaggio.

Ciò rende un po' più complessa la loro creazione, ma ha il vantaggio di evitare inconsistenze nel modello in caso di errori per azioni complesse, qualora fossero gestite con lo scambio di più messaggi.

In allegato a questo documento:

- Sequence Diagram creazione partita 2 giocatori
- Sequence Diagram azione "Gioca un assistente"
- UML ad alto livello della parte di rete

Messaggi Request

I messaggi request sono messaggi che vengono generati dal client e inviati al server, possiamo distinguere 2 categorie

- Setup Request Messages
- Game Request Messages

Tutti implementano l'interfaccia *RequestMessage*, che a sua volta estende *Serializable*

Setup Request Messages

I Setup Request Messages, implementano l'interfaccia *SetupExecute*

Sono utilizzati nella fase iniziale del gioco e servono ad:

- Scegliere l'username
- Creare una lobby
- Selezionare una lobby esistente
- Scegliere il team, in una lobby per esperti

L'interfaccia *SetupExecute* ha il metodo *execute* con la seguente signature:

```
void execute(Connection connection);
```

Ogni Setup Request Message implementerà il metodo in base al suo scopo.

Nota: la classe *Connection* è la nostra Remote View

Game Request Messages

I Game Request Messages implementano l'interfaccia *ControllerExecute*, oppure

ControllerExecuteExpertMode, se stiamo giocando una partita per esperti.

Sono utilizzati durante lo svolgimento della partita e rispecchiano le azioni che un giocatore può effettuare, ovvero:

- Giocare una carta assistente
- Spostare studenti
- Spostare madre natura
- Scegliere una nuvola

- Giocare una carta personaggio, in una partita per esperti

L'interfaccia *ControllerExecute* ha il metodo *execute* con la seguente signature:

```
void execute(Controller controller);
```

Ogni Game Request Message implementerà il metodo andando a richiamare i metodi del controller opportuni.

Messaggi Response

I messaggi request sono messaggi che vengono generati dal server e inviati al client, possiamo distinguere 2 categorie

- Setup Response Messages
- ReducedModel Response Messages

Tutti implementano l'interfaccia *RequestMessage*, che a sua volta estende *Serializable*.

Tutti implementano l'interfaccia *ClientExecute*, la quale è composta dal metodo *execute* con la seguente signature:

```
void execute(Client client);
```

Ogni Game Message implementerà il metodo andando a richiamare i metodi del client opportuni.

Setup Response Messages

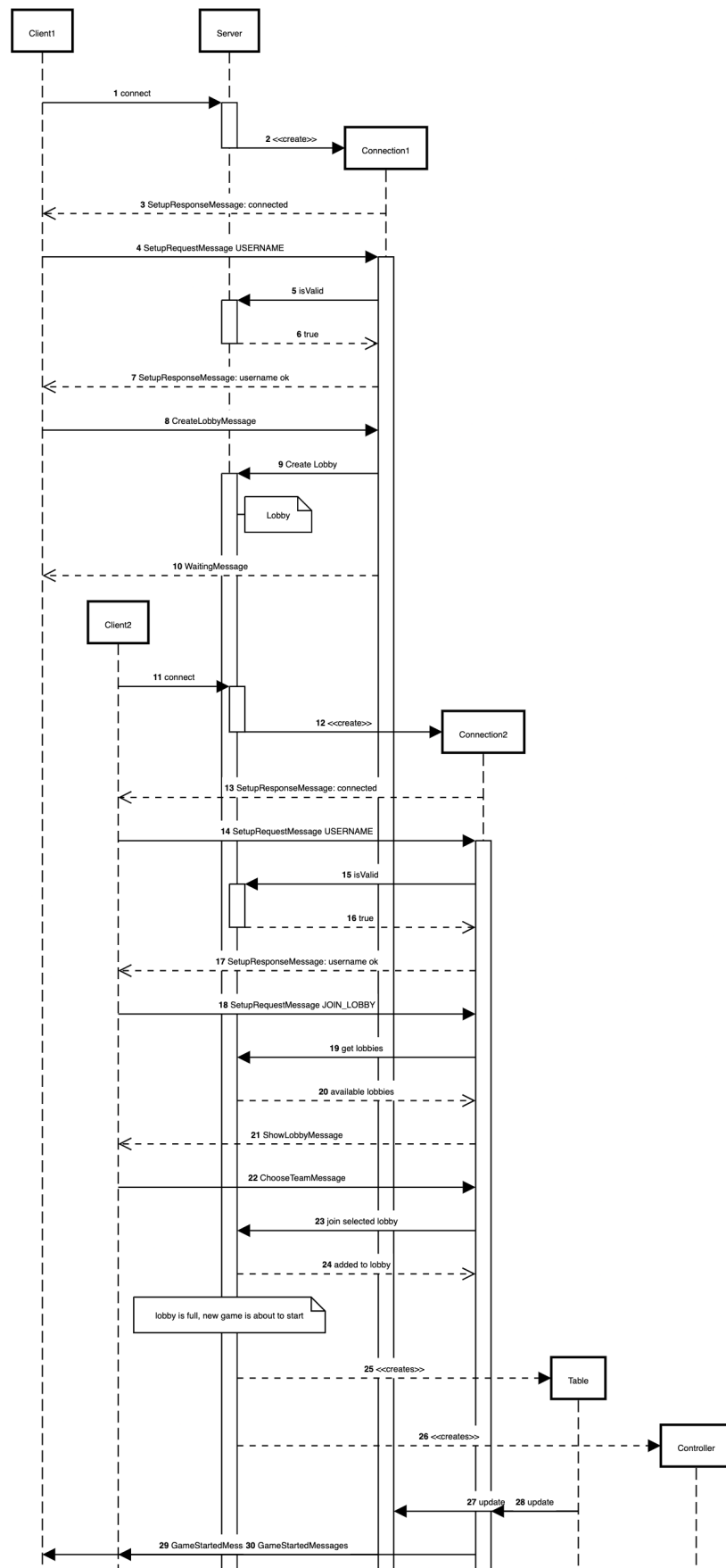
I Setup Response Messages sono messaggi di risposta ai Setup Request Message.

ReducedModel Response Messages

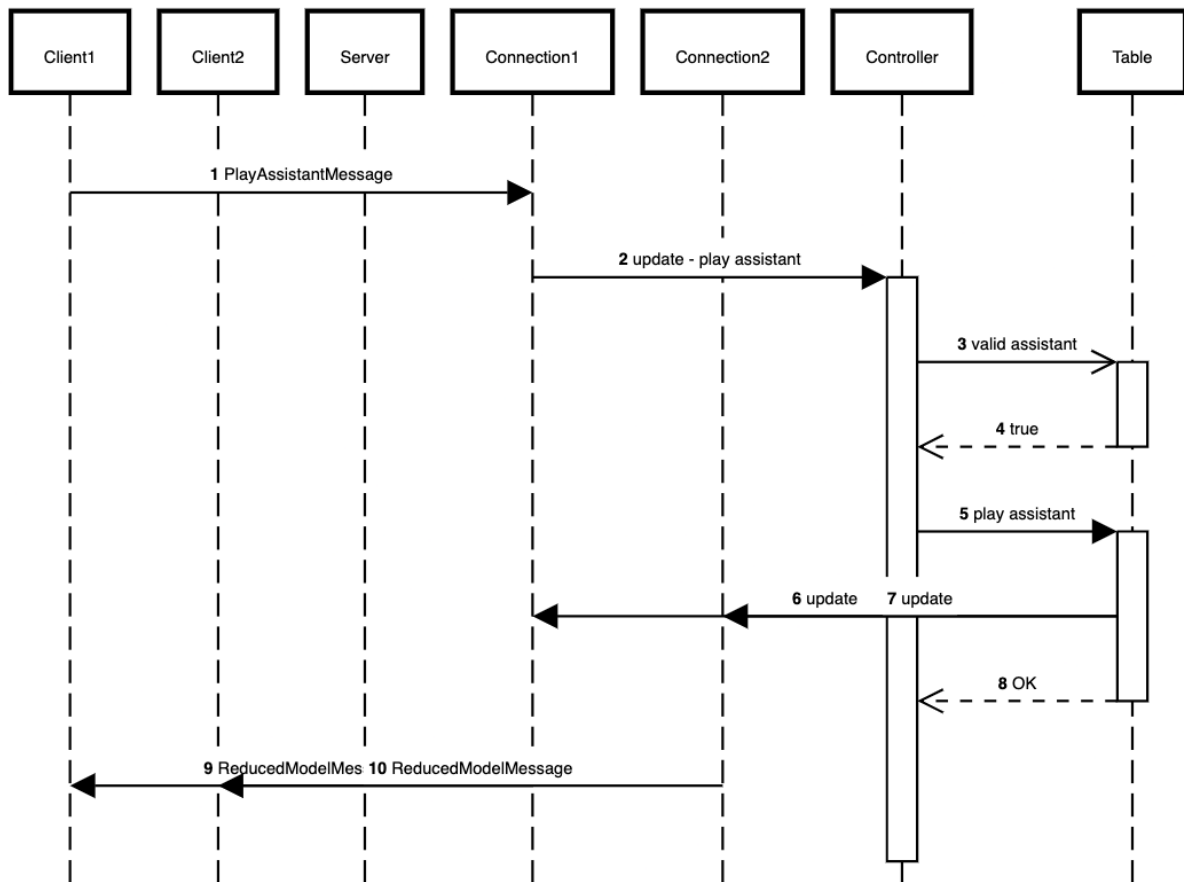
I ReducedModel Response Messages sono principalmente messaggi di risposta ai Game Request Message, in particolare abbiamo:

- *ReducedModelMessage*: tramite questo messaggio viene inviato al client un nuovo modello ridotto.
- *GameStartedMessage*: tramite questo messaggio viene notificato al client l'inizio di una partita.
- *EndGameMessage*: tramite questo messaggio viene notificato al client la fine di una partita.
- *ServerErrorMessage*: tramite questo messaggio viene notificato al client un errore generico.
- *CharacterPaidMessage*: tramite questo messaggio viene notificato al client che il personaggio scelto è già stato pagato.
- *DisconnectErrorMessage*: tramite questo messaggio viene notificata una disconnessione.

Allegato 1 - Sequence Diagram creazione partita di 2 giocatori



Allegato 2 - Sequence Diagram fase “Gioca un Assistente”



Allegato 3 - UML ad alto livello della parte di rete

