

PROTOCOLLO DI RETE AGGIORNATO (GRUPPO AM40)

Il flusso di rete si basa su oggetti che estendono la classe astratta **Data**, la quale contiene una descrizione del tipo di Data e un nickname, cioè un nome associato al giocatore a cui un Data è diretto o da cui un Data proviene.

Ogni Data è inviato tramite rete dalla classe *NetworkManager* con il metodo *sendSerializedMessage()*; se si tratta del manager del client (è il client che fa una richiesta a server) allora i Data prendono il nome di Active, mentre se è il server a rispondere ad una richiesta o a dover inviare aggiornamenti, per esempio ad un *NetworkParty* allora il Data è Passive.

Lato client lo specifico *<? Extends Data>* partono in base allo *State* in cui ci si trova. Anche per gli stati c'è suddivisione di stati Active (è l'user che deve immettere un input) e Passive (user riceve un update del gioco/sta giocando qualcun altro).

L'arrivo dei Passive Data su client determina cambiamenti di stato secondo il principio della FSM.

I Data in arrivo su client vengono gestiti e mostrati a schermo tramite la classe *Client* e il *ViewManager*, dove i Data si propagano sotto forma di *Message* interni al client.

Lato server esistono due classi che vengono aggiornate in base alle richieste di *join game* o *create game*:

- *NetworkParty*, che indica un gruppo di client che sono in gioco nella stessa partita o che intendono iniziare una
- *NetworkClient*, che rappresenta un singolo user che si è loggato al server; questa classe mantiene tutte le informazioni di un client per quanto riguarda la rete

I Data in arrivo su server vengono filtrati e propagati nell'MVC da una classe *VVServer* tramite un sistema di notifiers e listeners. Nell'MVC viaggiano *Actions*, oggetti che corrispondono ai Data nel contesto interno del server.

Per convertire un Data in una Action o in un Message abbiamo i metodi *onServer()* e *onClient()* che rispettivamente creano la Action su server e il Message su client.

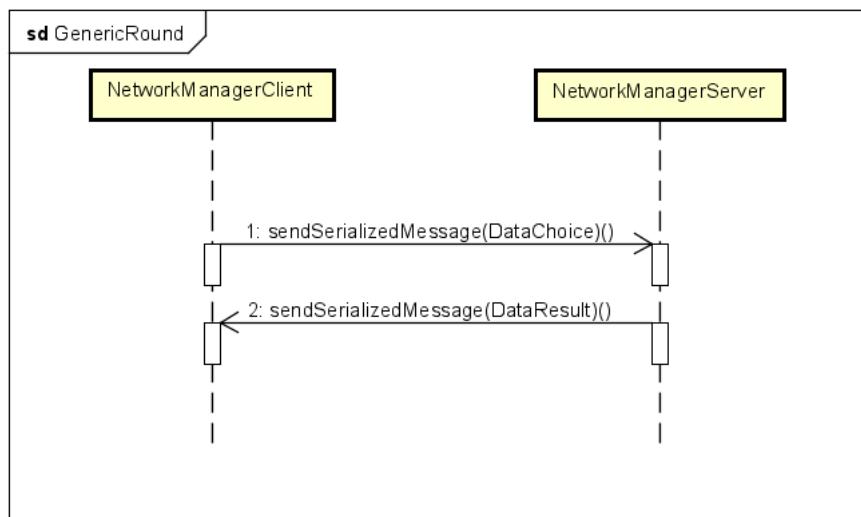
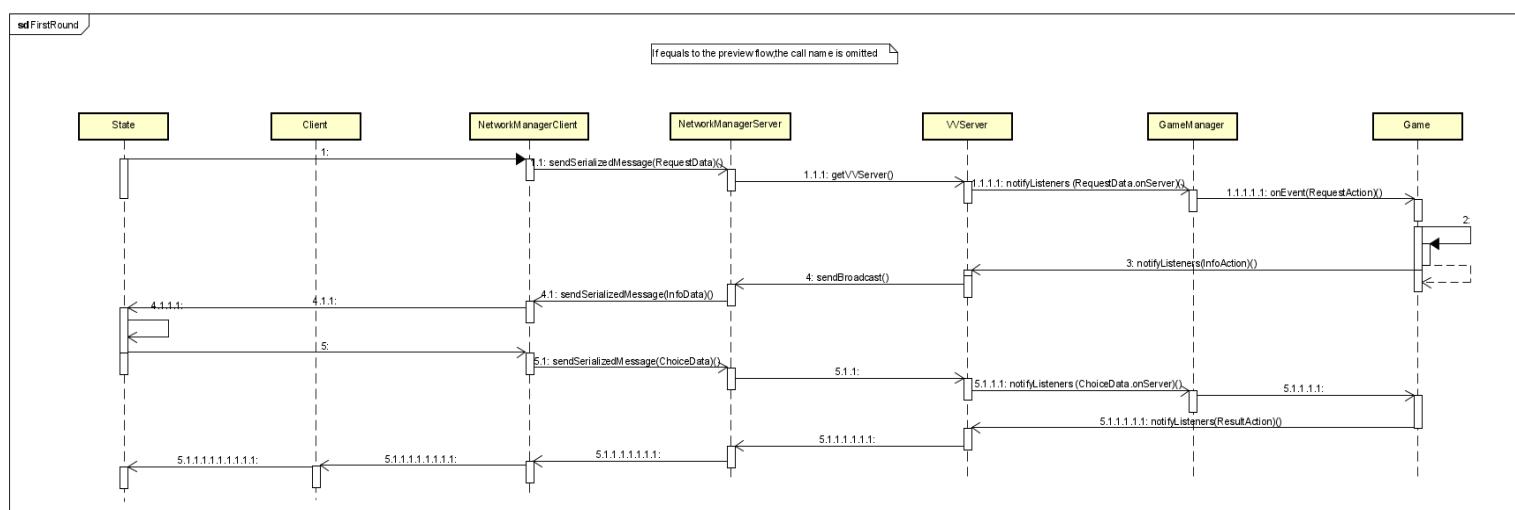
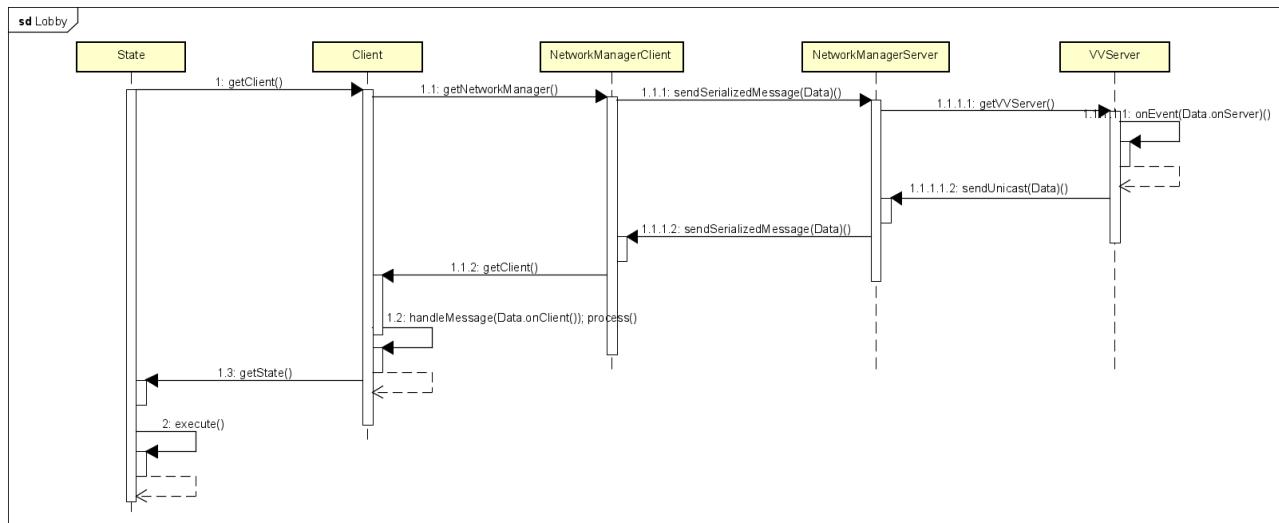
Possiamo dividere i Data nella gerarchia seguente:

- Active
 - *FirstRound*: situazione di primo round del player attivo.
 - *Flow*: gestione lobby (join o create di un game) e altri Data di utilità
 - *Round*: situazione di round generico del player giocante
- Passive
 - *FirstRound*: situazione di aggiornamento players riguardo a scelte del primo round
 - *Flow*: gestione lobby (join o create di un game) e altri Data di utilità
 - *Round*: situazione di aggiornamento client su azioni nel round generico

Ogni Data ha il suo specifico payload realizzato per mezzo di attributi (nelle classi *<? Extends Data>*).

Nel First round il flusso è: *(client)requestData -> (server)InfoData -> (client)ChoiceData -> (server)ResultData*

Nelle altre situazioni abbiamo le data Attive e Passive accoppiate a due a due più Data che segnalano Fail per alcune azioni



Per gli Active abbiamo i seguenti Data (signature):

```
@JsonSubTypes.Type(value = CreaterequestData.class, name = "CREATE_GAME"),
@JsonSubTypes.Type(value = GameIDChoiceData.class, name = "GAME_ID_CHOICE"),
@JsonSubTypes.Type(value = JoinrequestData.class, name = "JOIN_GAME"),
@JsonSubTypes.Type(value = ReadyToPlayData.class, name = "READY_TO_PLAY"),

@JsonSubTypes.Type(value = StartingCardRequestData.class, name =
"STARTING_CARD_REQUEST"),
@JsonSubTypes.Type(value = StartingCardChoiceData.class, name =
"STARTING_CARD_CHOICE"),
@JsonSubTypes.Type(value = TokenChoiceData.class, name = "TOKEN_SELECTION"),
@JsonSubTypes.Type(value = TokenRequestData.class, name = "TOKEN_REQUEST"),
@JsonSubTypes.Type(value = DealCardsData.class, name = "CARDS_DEAL"),
@JsonSubTypes.Type(value = AimCardRequestData.class, name = "AIM_CARD_REQUEST"),
@JsonSubTypes.Type(value = AimCardChoiceData.class, name =
"AIM_CARD_SELECTION"),
@JsonSubTypes.Type(value = PlayersOrderRequestData.class, name =
"PLAYERS_ORDER_REQUEST"),

@JsonSubTypes.Type(value = PlacingData.class, name = "PLACING"),
@JsonSubTypes.Type(value = DrawData.class, name = "DRAW"),
@JsonSubTypes.Type(value = ChangeTurnRequestData.class, name = "CHANGE_TURN"),
```

Per i Passive le seguenti:

```
@JsonSubTypes.Type(value = CreateresponseData.class, name = "CREATE_RESULT"),
@JsonSubTypes.Type(value = FailedCreationData.class, name = "FAILED_CREATION"),
@JsonSubTypes.Type(value = JoinresponseData.class, name = "JOIN_RESPONSE"),
@JsonSubTypes.Type(value = NoActivePartiesData.class, name =
"NO_ACTIVE_PARTIES"),
@JsonSubTypes.Type(value = GameIDResultData.class, name = "GAME_ID_RESULT"),
@JsonSubTypes.Type(value = FailedGameIDData.class, name = "FAILED_GAME_ID"),
@JsonSubTypes.Type(value = GameInitResultData.class, name = "GAME_INIT_RESULT"),
@JsonSubTypes.Type(value = NotEnoughPlayersData.class, name =
"NOT_ENOUGH_PLAYERS"),

@JsonSubTypes.Type(value = StartingCardInfoData.class, name = "STARTING_INFO"),
@JsonSubTypes.Type(value = StartingCardResultData.class, name =
"POSITIVE_STARTING_CARD"),
@JsonSubTypes.Type(value = TokenInfoData.class, name = "TOKEN_INFO"),
@JsonSubTypes.Type(value = PositiveTokenColorData.class, name =
"POSITIVE_TOKEN_COLOR"),
@JsonSubTypes.Type(value = DealCardsResultData.class, name =
"CARDS DEAL RESULT"),
@JsonSubTypes.Type(value = AimCardInfoData.class, name = "AIM_CARD_INFO"),
@JsonSubTypes.Type(value = AimCardResultData.class, name = "AIM_CARD_SELECTED"),
@JsonSubTypes.Type(value = PlayersOrderInfoData.class, name =
"PLAYERS_ORDER_INFO"),
@JsonSubTypes.Type(value = PositivePlacingData.class, name =
"POSITIVE_PLACING"),
@JsonSubTypes.Type(value = RepeatPlacingData.class, name = "REPEAT_PLACING"),
@JsonSubTypes.Type(value = PositiveDrawData.class, name = "POSITIVE_DRAW"),
@JsonSubTypes.Type(value = RepeatDrawData.class, name = "REPEAT_DRAW"),
@JsonSubTypes.Type(value = ChangeTurnInfoData.class, name = "CHANGE_TURN_INFO"),
@JsonSubTypes.Type(value = LastRoundsInfoData.class, name = "LAST_ROUNDS"),
@JsonSubTypes.Type(value = EndGameData.class, name = "ENDGAME"),
@JsonSubTypes.Type(value = PingData.class, name = "PING"),

@JsonSubTypes.Type(value = HostNameData.class, name = "HOST_NAME"),
@JsonSubTypes.Type(value = InterruptedGameData.class, name =
```

```
"INTERRUPTED_GAME"),
@JsonSubTypes.Type(value = FailedHostNameData.class, name = "FAILED_HOST_NAME"),
@JsonSubTypes.Type(value = ClientDisconnectedData.class, name =
"CLIENT_DISCONNECTED")
```