Peer-Review 2: Communication Protocol

Ding JiaLiang, Cai Simona, Mao YangHao, Chen Ken

Gruppo 21

Valutazione del Protocollo di comunicazione del gruppo 30.

# Lati positivi

A primo impatto, il diagramma appare sintetico e ben oraganizzato, così da semplificarne la lettura.

Infatti, dopo aver analizzato i messaggi del Client verso il Server possiamo affermare che sono adeguati e contengono tutte le informazioni necessarie al Server per eseguire i vari metodi, tranne che per il MoveCommand. Giustificheremo meglio questa presa di posizione nei “Lati negativi”.

Invece, ci sentiamo d’accordo con la scelta fatta per le tipologie d’errore che interessano JoinGameCommand, anche se NicknameAlreadyExisting sembra essere mal piazzato.

Da come suggerisce la presenza del GameCode, ipotizziamo ci sia l’intenzione di implementare la funzione Multi-partita, dunque la verifica dell’unicità del nome dovrebbe avvenire prima del Join ad una partita.

Anche per i messaggi del Server rivolti al Client ci sentiamo di dire che sono pressoché completi.

Non abbiamo notato i messaggi del Server che annunciano l’inizio della partita o la conquista di un Common Goal. Ipotizziamo che l’omissione sia dovuta al fatto che non è necessaria una reazione del giocatore ad essi.

In generale, la maggior parte delle interazioni inizializzate dal client sono stati coperti dal ‘sequence diagram(without diagrams)’ e da ‘chat’.

# Lati negativi

1. Messaggi d’errore:

Ipotizzando che le nostre assunzioni su alcuni messaggi non siano errate, il messaggio InvalidTilesNumber viene inviato quando il numero di carte selezionate non è adeguato per l’inserimento. Allora cosa succede se il numero di carte è entro i limiti, ma la colonna selezionata per l’inserimento non ha spazio sufficiente?

Secondo noi, questo è uno scenario da tenere in considerazione e che nel diagramma manca.

Inoltre non ci è chiaro se in ReceivedPrivateMessage è tenuto in considerazione sia il fallimento dell’invio che l’assenza del destinatario nel gioco. A proposito di questo, nel metodo sembra non essere specificato il nome del destinatario. Non ci è chiaro se il nome si trovi all’interno del messaggio o se sia stato omesso.

Per quanto riguarda il login, pensiamo che manchi una risposta del Server in caso sia stato scelto un nome già preso da un altro giocatore, che dovrebbe avvenire prima del JoinGame.

1. MoveCommand:

Guardando i pacchetti dei messaggi, ipotizziamo che le azioni di selezione delle celle (con ordine) e scelta della colonna per l’inserimento vengano svolte dal lato Client e successivamente tutte queste informazioni vengono inviati al Server per essere approvate.

Quindi, a seguito dell’ interpretazione appena fatta, pensiamo che sia svantaggioso per il client comprimere tutte queste informazioni in un unico pacchetto. Infatti, in caso di risposta negativa del server, il client è costretto a dover ripercorrere tutte le fasi di scelta, anche quando l’errore è dovuto solo ad una parte del messaggio.

# Confronto tra le architetture

Inanzitutto, il diagramma presentato dal gruppo 30, grazie alla sua sinteticità, è più facile da leggere rispetto al nostro, che d’altro canto è leggermente più dettagliato nel mostrare le varie fasi di gioco in diagrammi separati.

Nel complesso, però, l’approccio su alcuni aspetti è molto simile nei due gruppi, soprattutto per quanto riguarda la comunicazione durante il JoinGame e la chat.

Invece la parte più discordante è probabilmente la parte che descrive i “movimenti” del giocatore.

Infatti il nostro gruppo ha preferito scomporre le parti del MoveCommand in parti più piccole, aumentando i contatti al Server, per avere un risponso più immediato e preciso in caso di errore.