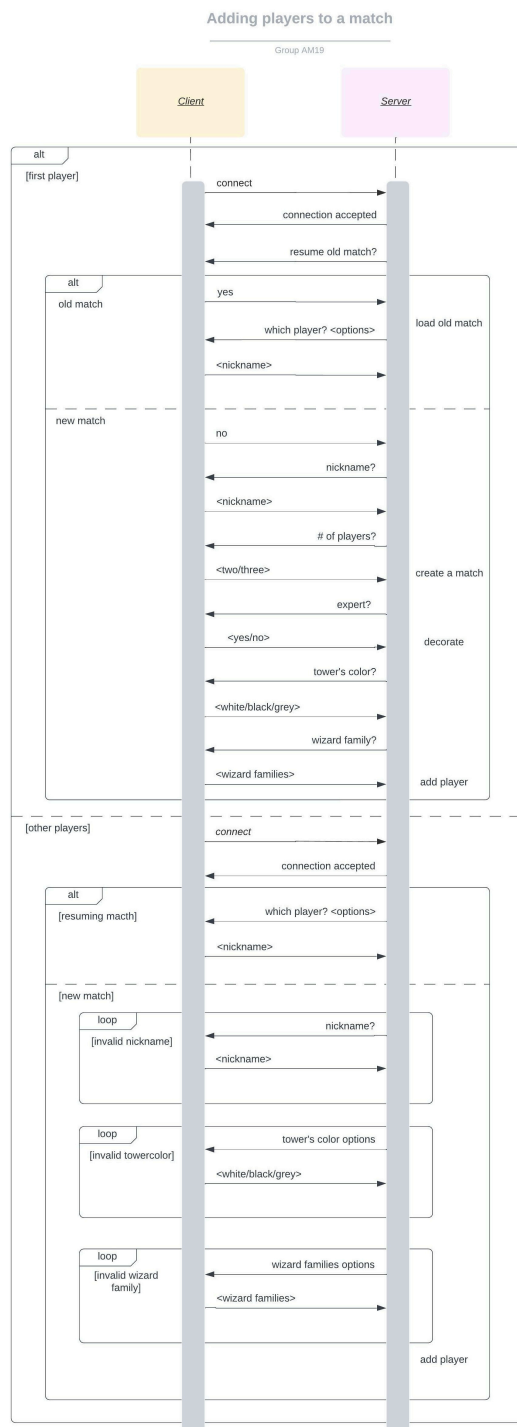


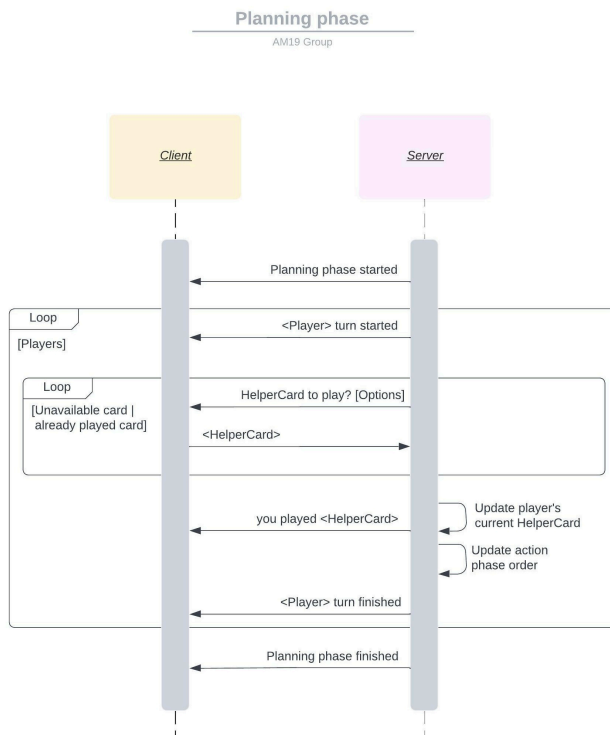
Relazione Communication Protocol - Gruppo AM19

Match Setup



- Un nuovo client manda una richiesta di connessione al server
- Il server, se ci sono ancora posti disponibili all'interno della partita, la accetta. Se è il primo client che si connette gli si chiede se vuole riprendere una partita precedentemente interrotta
 - Nel caso in cui l'utente voglia riprendere un match interrotto in precedenza il server gli chiederà di indicare quale fosse il suo nickname in quella partita, dopo avergli presentato le opzioni disponibili, ricrea il match e gli assegna quel player
 - Altrimenti gli si chiede di indicare il nickname, dopo la sua risposta e gli si chiedono in successione, ma attendendo sempre una risposta da parte del client tra una domanda e l'altra, da quanti giocatori vuole creare la partita, quale ruleset, semplice o esperto, vuole utilizzare e infine, di che colore vuole le torri e che famiglia di carte aiutante vuole usare
- Nel caso in cui il client connesso non sia il primo, ma tuttavia entro il limite di giocatori che possono partecipare alla partita, si aprono nuovamente i due scenari di match ripreso e di match nuovo
 - Nel caso di match ripreso anche a questo utente verrà chiesto il nickname precedentemente utilizzato dopo avergli presentato le opzioni disponibili, il server aggiunge l'utente alla partita dandogli il player scelto
 - Nel caso di match nuovo gli verranno chiesti nell'ordine, nickname, colore delle torri e famiglia delle carte aiutante controllando che le scelte effettuate siano compatibili con quelle fatte dagli altri giocatori

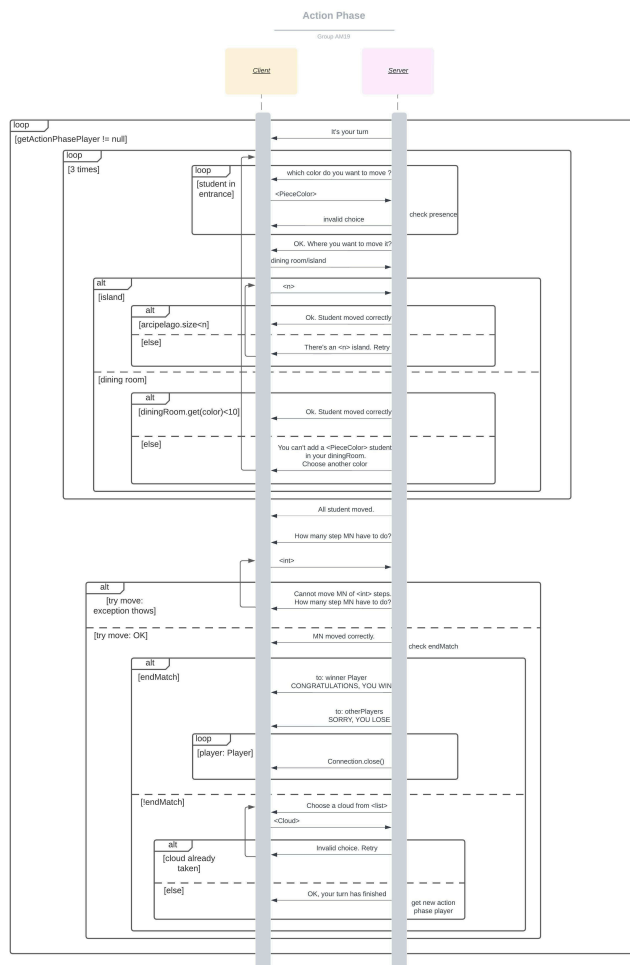
Planning Phase



Il diagramma simula lo scambio di messaggi tra client e server durante lo svolgimento della fase di pianificazione di una partita. L'interazione si articola secondo le seguenti modalità:

1. Il server notifica il client in merito all'inizio della planning phase
2. Lato server viene modificato il Model: le Clouds vengono riempite. A questo punto si itera su una lista di Player, ordinata in modo tale da avere come primo Player colui che nella planning phase precedente aveva giocato la carta assistente con numero più basso.
3. Per ciascun Player presente si comunica al client corrispondente l'inizio del proprio turno di pianificazione e si chiede quale carta assistente voglia essere giocata, fornendo come opzioni la lista numerata di carte ancora disponibili nel mazzo assistente. Questa richiesta viene inviata al client finché una carta valida non viene fornita.
4. Il client comunica al server la carta selezionata, fornendone il numero d'ordine nella lista di opzioni. L'invio di una opzione valida da parte del client scatena la restituzione di un messaggio di conferma e una serie di azioni interne al server: la carta scoperta del Player corrispondente viene aggiornata e il Player viene aggiunto ad una ulteriore lista di Player, che questa volta sarà costruita in maniera incrementale e ordinata in base all'ordine in cui la fase successiva dovrà essere giocata.
5. Al client viene comunicata la fine del proprio turno .
6. Finita l'interazione con tutti i client viene comunicata a tutti la fine della planning phase.

Action Phase



L'action phase si divide essenzialmente in 3 macro parti che vanno eseguite in ordine per ciascun player (ad eccezione dell'uso delle carte Personaggio, trattate su un sequence diagram diverso).

1. Muovere 3 studenti presenti in ingresso verso la sala o un'isola
In questo caso, inizialmente viene chiesto un colore di studente da spostare (controllando che la stringa in arrivo combaci con i 5 colori presenti nel gioco come "blue", "yellow"...e così via).

A quel punto va richiesto dove si vuole spostare lo studente:

- se viene scelta la sala, si controlla che non sia già piena per quel colore, altrimenti si richiede di riscegliere da capo il colore
- se si sceglie l'isola, viene richiesta la scelta del numero dell'isola corrispondente a cui si vuole aggiungere. Questo numero quindi dovrà essere compreso tra 1 e la size dell'arcipelago, altrimenti verrà richiesto di reinviare un numero corretto.

2. Movimento di Madre Natura

si chiede il numero corrispondente a quanti passi deve fare Madre Natura.

A quel punto si prova a muovere Madre Natura di quelle posizioni: in caso di errori, si ritorna indietro a chiedere un nuovo numero. Dopo il movimento di Madre Natura potrebbe essere scattata qualche condizione di fine partita immediata, in quel caso si notificano i giocatori e le connessioni vengono chiuse

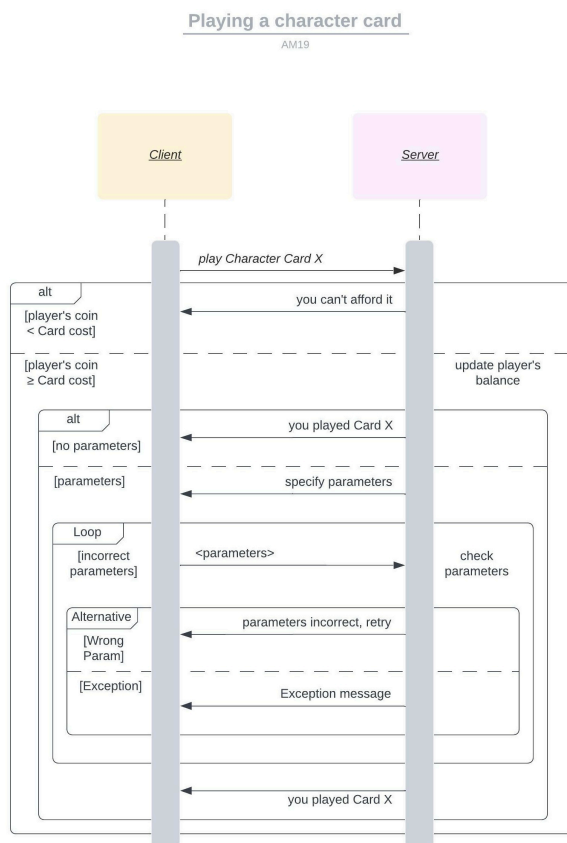
3. Scelta nuvole

Se non si sono verificate queste condizioni, allora si può continuare col terzo passo, in cui viene richiesto sempre un numero corrispondente alla nuvola da cui prendere gli studenti.

Anche in questo caso, in caso non sia possibile prendere quella determinata isola, viene richiesto di reinserire il numero finché non va a buon fine.

Al termine, si passa ad un altro player finché tutti non hanno giocato per quel round.

Play a Character Card



- Da parte del client arriva una richiesta una data Character Card, ad esempio l'Araldo
- Lato server il controller verifica che il giocatore abbia abbastanza monete per giocare la carta, se non dovesse potersela permettere, questa interazione termina e la partita riprende dalla fase in cui si era interrotta. Nel caso in cui il giocatore abbia monete a sufficienza, queste gli verranno scalate e verrà verificato che parametri richiede la carta, in questo esempio un'isola. Viene inviato un messaggio al client per richiedere i parametri
- Il client risponde con i parametri, in questo caso, come avviene per la Action Phase, verrà inviato un messaggio contenente il numero dell'isola indicato dall'utente
- Se il parametro è corretto l'azione viene eseguito e viene comunicato al client che ha giocato con successo la carta Araldo, altrimenti verranno inviati dei messaggi di errore e sarà richiesto un parametro diverso

In merito a quest'ultimo punto sono possibili due messaggi di errore differente poiché uno viene generato dalla carta nel caso in cui, ad esempio, si cerchi di prendere uno studente non presente sulla carta Giullare, mentre l'altro tipo di messaggio viene generato da un controllo dei parametri eseguito nel controller, ad esempio, richiedo un calcolo d'influenza sull'isola 6, ma ho solo cinque isole in partita in quel momento

Messages

▼ LoginPlayersOptions

messaggio mandato al giocatore che si sta connettendo ad un match precedentemente interrotto per permettergli di scegliere qual è il nickname del suo player virtuale

▼ LoginNickname

contiene il nickname di un nuovo giocatore o quello di un giocatore entrante in un match precedentemente interrotto

▼ LoginTowerColor

contiene il colore della torre scelto dal nuovo giocatore

▼ LoginWizardFamily

contiene la famiglia di carte scelta dal nuovo giocatore

▼ LoginReply

messaggio inviato dal serve per dire che il login è stato effettuato correttamente

▼ PlayCharacterCard

deve contenere l'ID, ovvero il nome, della carta giocata

▼ RequestColor

<color>

▼ RequestIsland

island <n>

▼ RequestPieceColorList

list<PieceColor>

▼ CharacterCardParameters

deve contenere i parametri passati per giocare la carta

<PieceColor>

<Island>

<PieceColorList>

▼ GenericMessage

▼ "Connection accepted"

▼ "Resume old match?"

▼ "write your nickname"

▼ "How many players?"

▼ "do you want to play with an expert game?"

▼ "You can't afford it"

▼ <PhaseType> + "started"

▼ <PhaseType> + "finished"

▼ "Clouds have been refilled"

▼ "Final round started"

▼ <Player> + <phase> + "started"

▼ <Player> + <phase> + "finished"

▼ "On which color you want to activate the effect?"

▼ "On which island you want to activate the effect?"

▼ "Write which colors you want to swap around"

▼ "which color you want to move?"

▼ "where do you want to move?"

▼ "which cloud do you want to take? + <list>"

▼ "How many step MN have to do?"

▼ "Last round"

▼ ErrorMessage

▼ String error

▼ "Can't move MN for <int> step. Retry"

▼ "Invalid choice"

▼ PlayHelperCard

<n> of next round order

▼ PlayableHelperCard

list<n> of next Round Order card still available to use

▼ EntranceToDining

```
//ackMessage  
//parameterProblems
```

▼ EntranceToIsland

```
<n> island
```

▼ MotherNatureStep

```
<n> step
```

▼ ChooseCloud

```
cloud <n> to take
```

▼ EndMatchMessage

```
server → players: Player <nickname> wins this match
```