

INTRODUZIONE

La comunicazione tra Client e Server può avvenire, a scelta del giocatore, tramite Socket o tramite RMI (il server verrà avviato in entrambe le modalità).

Nel caso di connessione attraverso Socket tra le due parti coinvolte, i messaggi, prima di essere inviati, vengono serializzati tramite JSON e sono poi deserializzati sull'altro lato. Su quest'ultimo verrà poi analizzata la stringa messaggio e verranno chiamati i metodi corrispondenti all'azione richiesta.

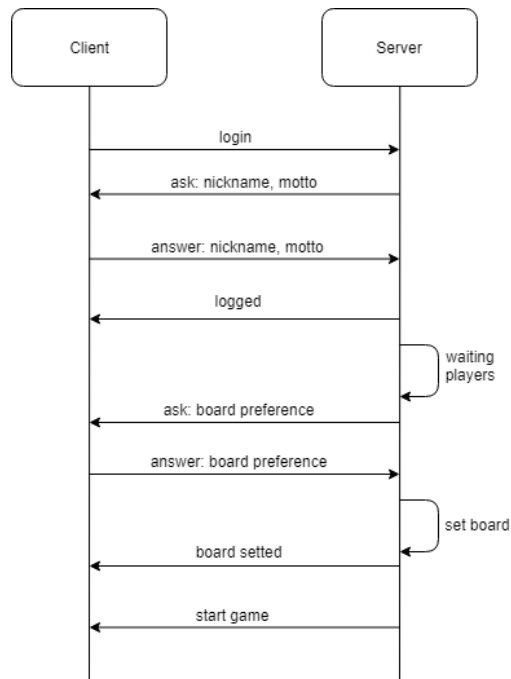
Invece, nel caso in cui il giocatore scelga il protocollo RMI, non verranno scambiati messaggi tra client e server, ma la comunicazione avverrà tramite la chiamata di metodi remoti. Ad esempio il client avrà la possibilità di chiamare direttamente i metodi login, deregister e sendMatchAnswer presenti sul server.

Tranne che nel caso di nickname già in uso durante il login, abbiamo deciso di "impedire di sbagliare" al giocatore controllando in anticipo le possibili azioni che può eseguire e dandogli la possibilità di selezionare solo una di queste.

DESCRIZIONE FASI DI COMUNICAZIONE

- 1) Initialization
- 2) First Turn
- 3) Powerup scenario
- 4) Action scenario
- 5) Requesting player info
- 6) Deregister client

Initialization



Il client chiama la funzione di login facendo richiesta di entrare nella waiting room.

Il server richiede nickname e motto da collegare al client richiedente.

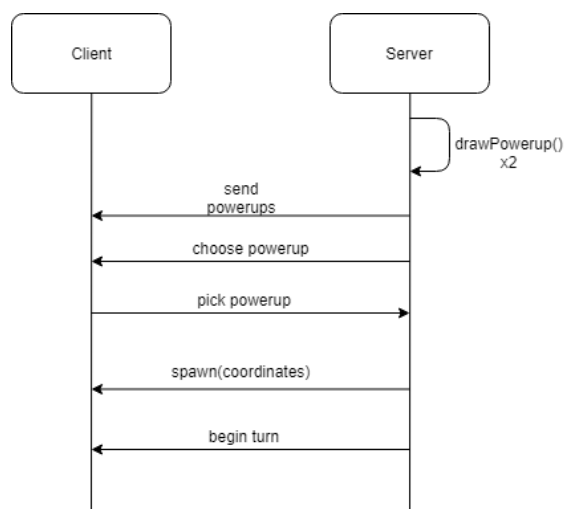
Il client risponde e il server controlla che non ci sia un altro player con nickname uguale.

Dopo aver loggato il client, il server richiede una preferenza sulla game board e attende relativa risposta.

Calcolate le preferenze dei giocatori il server setta la board e la comunica al client.

Pronti, VIA! new Game();

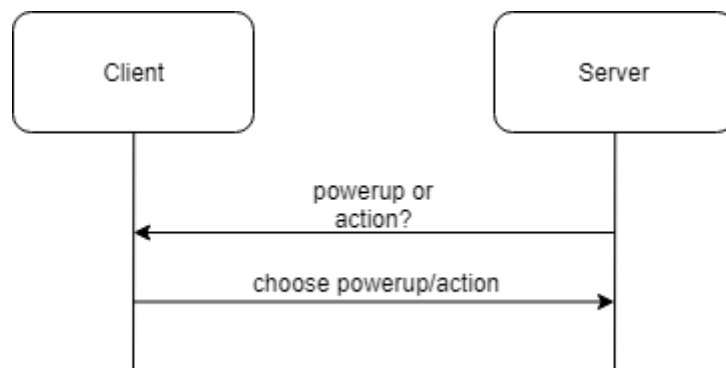
First Turn



Nella fase di pre-turno vengono pescati 2 powerups (se è il primo turno, 1 se il giocatore è in fase di respawn) e viene chiesto al client di sceglierne uno

Comunicata la scelta del powerup al server, viene fatto un check sul colore del powerup scartato e mandato un messaggio al client con le coordinate del quadrato di spawn del player.

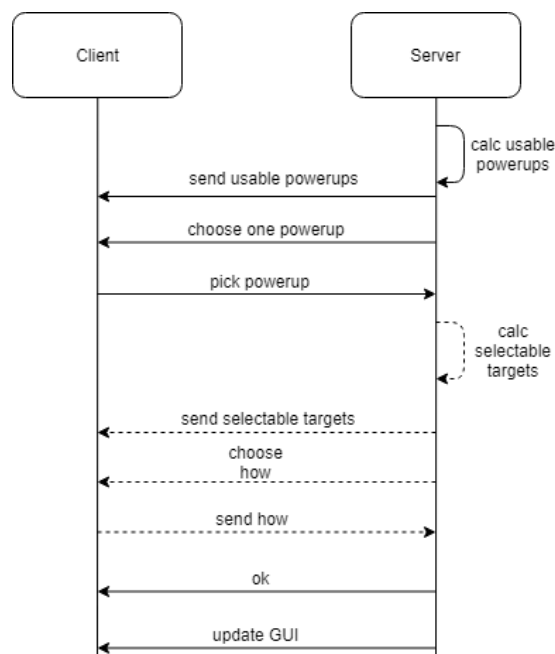
Ha così inizio il vero e proprio turno.



Il server chiede al client se vuole usare un powerup o fare la prima azione.

In base alla risposta del client si apriranno due diversi scenari.

Powerup scenario

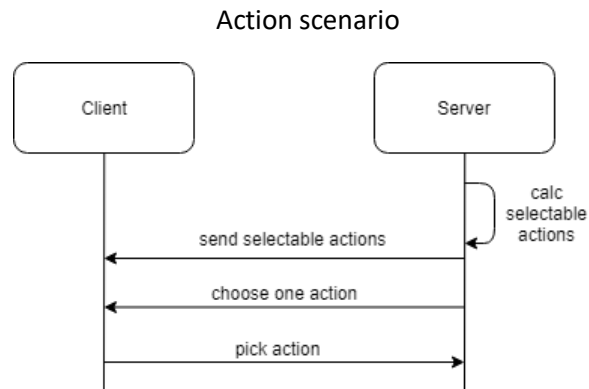


Nel caso scelto il server controlla quali powerup sono utilizzabili e li invia al client che ne sceglie uno.

Nel caso di Mirino il server calcola i giocatori colpibili e li manda al client.

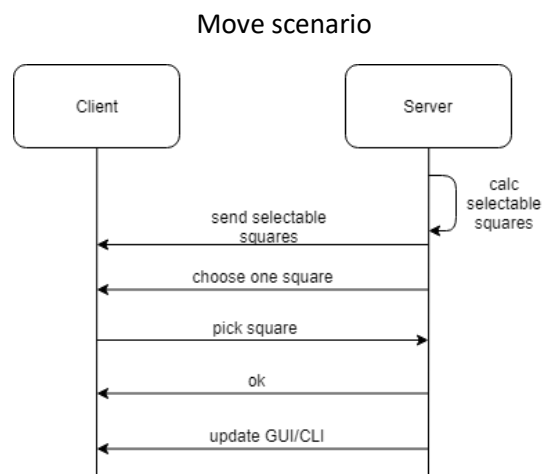
Nel caso di Mirino, Raggio Cinetico e Teletrasporto il server chiede al client come vuole usare il powerup.

Dopo la risposta del client il server manda un messaggio di accettazione e aggiorna la GUI/CLI.



Nell'altro scenario il server calcola le azioni selezionabili dal giocatore e gliela invia chiedendogli di sceglierne una.

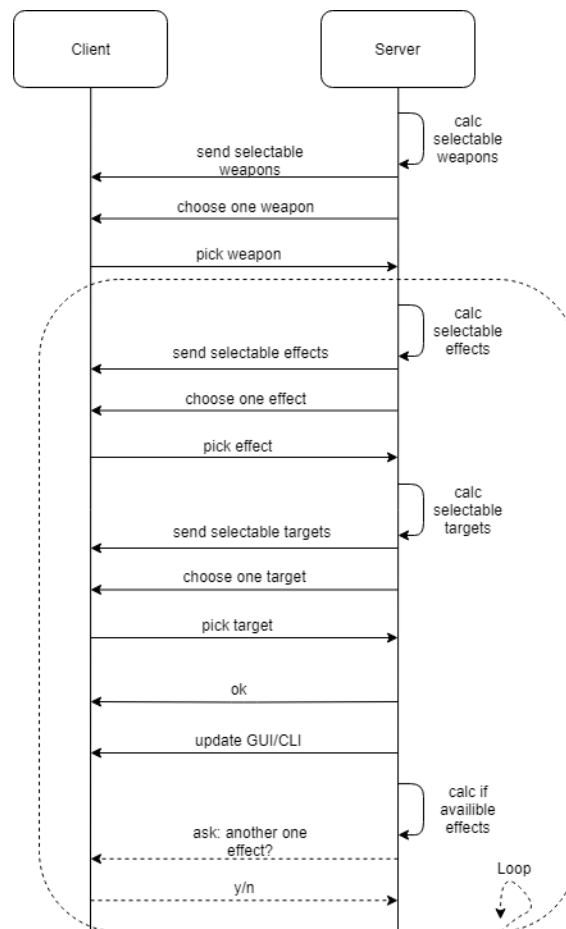
In base alla scelta del client si apriranno 3 diversi sotto scenari.



Nel caso il giocatore scelga di muoversi, il server calcola i quadrati in cui può spostarsi, glieli invia e gli chiede di sceglierne uno.

Dopo la risposta del client ne conferma la scelta e aggiorna la GUI/CLI.

Shot scenario



Nel caso il giocatore scelga di sparare, il server calcola le armi utilizzabili, gliele invia e gli chiede di sceglierne una.

Di conseguenza calcola gli effetti usabili inviandoli al client e gli chiede di sceglierne uno.

In base all'effetto scelto il server calcola i bersagli selezionabili, li invia al client e chiede di scegliere i bersagli/o.

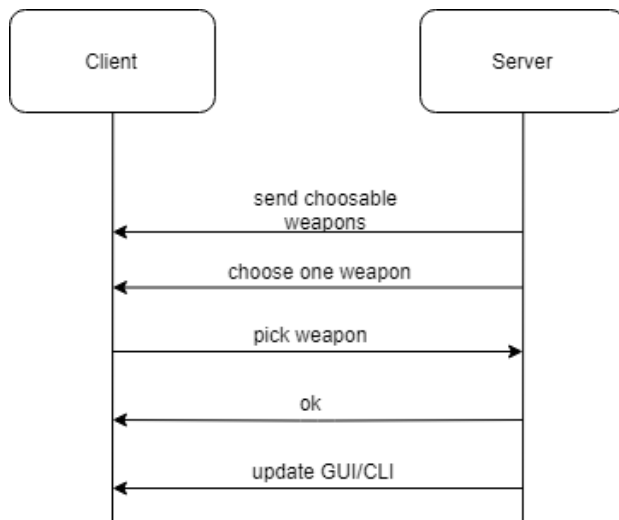
Dopo la risposta del client il server manda un messaggio di accettazione e aggiorna la GUI/CLI.

Infine in caso ci siano altri effetti fattibili viene richiesto al client se ha intenzione di eseguirli.

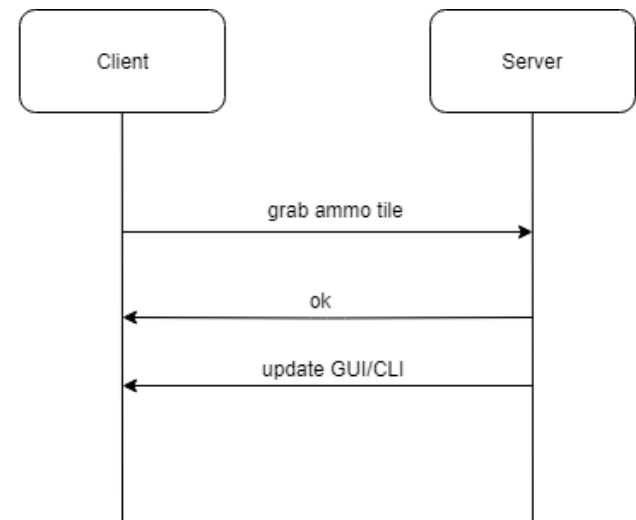
Questo processo di scelta dell'effetto e conseguenti bersagli viene ripetuto fino a che è possibile.

Grab scenario

Weapon



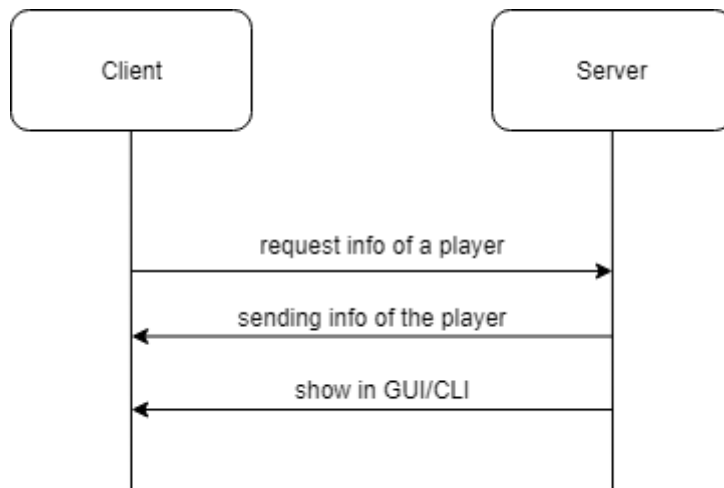
AmmoTile



Caso weapon: il server manda la lista di armi disponibili nel quadrato del giocatore e gli chiede di prenderne una. Una volta scelta l'arma il server manda un messaggio di accettazione e aggiorna la GUI/CLI.

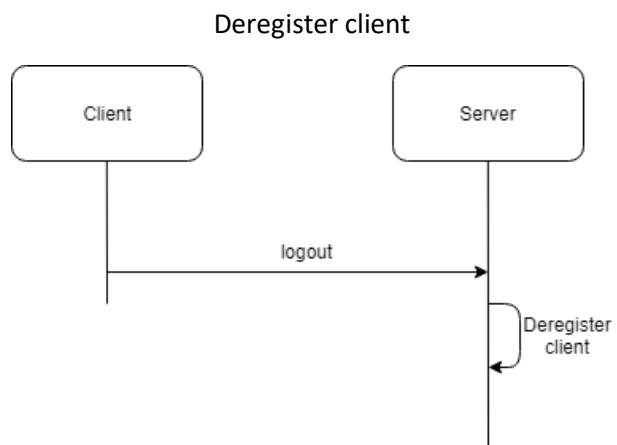
Caso AmmoTile: il client raccoglie la tessera munizioni e il server conferma e modifica la GUI/CLI.

Requesting player info



Il client manda un richiesta di informazioni su un player.

Il server risponde con le info richieste e le mostra al client.



Il client manda un messaggio di disconnessione.

Il server provvede a deregistrarlo correttamente.