

# Progetto Client-Server Multithreading

## La ruota della fortuna de noartri©

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Regole di Gioco</b>	<b>2</b>
<b>3</b>	<b>Protocollo di Comunicazione</b>	<b>4</b>
3.1	Fase Connessione e Lobby . . . . .	4
3.2	Fase Gioco Principale . . . . .	4
3.3	Gestione Ruota . . . . .	5
3.4	Lettere . . . . .	5
3.5	Soluzione . . . . .	5
3.6	Gestione Cambio Turno . . . . .	6
3.7	Fase Finale - Ruota delle Meraviglie . . . . .	6
3.8	Gestione Eventi e Errori . . . . .	7
<b>4</b>	<b>Esempio Iterazione Completa, 2 Giocatori, 1 Turno, Nessun Errore</b>	<b>8</b>
4.1	Flowchart dell'iterazione . . . . .	12
<b>5</b>	<b>Architettura del Server</b>	<b>14</b>
5.1	Componenti del Server . . . . .	14
5.2	Gestione della Concorrenza . . . . .	14
<b>6</b>	<b>Architettura del Client</b>	<b>15</b>
6.1	Componenti del Client . . . . .	15
6.2	Gestione del Multithreading Lato Client . . . . .	15

# 1 Introduzione

Il presente progetto consiste nella realizzazione di un sistema Client-Server multithreading per l'implementazione del gioco “La ruota della fortuna de noartri”. L'architettura consente:

- 1-3 giocatori per ogni partita
- Gestione di N partite simultanee

# 2 Regole di Gioco

Il sistema implementa le seguenti regole:

- **Giocatori:** 1-3 per partita
- Gioca un giocatore alla volta, vengono mostrati degli slot vuoti, questi slot compongono una frase con le lettere “cancellate”
- Il giocatore di turno dovrà girare la ruota e successivamente nominare una lettera dell’alfabeto (le vocali hanno un costo di 500, le consonanti sono gratuite)
- Se la lettera è presente, tutte le caselle dove si troverebbe quella lettera vengono scoperte e verranno aggiunti al giocatore una quantità di soldi pari al numero sorteggiato dalla ruota moltiplicato per il numero di lettere scoperte
- Se il giocatore riesce ad indovinare la frase, aggiunge i soldi del turno al suo salvadanaio
- Se il giocatore sbaglia la frase oppure dice una lettera che non è presente nella frase, il turno passa al giocatore successivo
- Nella ruota sono presenti, oltre a soldi, uno spicchio “Passa mano” che fa passare il turno al giocatore successivo, ed uno spicchio “Bancarotta” che oltre a far passare il turno al giocatore successivo, azzera sia i soldi del turno sia quelli del salvadanaio del giocatore
- Dopo X turni il giocatore con più soldi passa alla fase finale (Il numero di turni viene scelto dal giocatore numero 1 nella lobby)



Figura 1: Esempio del tabellone di gioco

### Fase Finale:

- Il giocatore dovrà girare una ruota diversa: la ruota delle meraviglie
- Da questa ruota verranno estratte 3 buste (contengono una quantità di soldi) numerate da 1 a 3
- Il giocatore verrà messo di fronte a 3 giochi, con 3 meccaniche diverse, a ogni gioco viene assegnata una busta
- Il giocatore ha 60 secondi di tempo per completare tutti i giochi, può decidere se passare al gioco successivo
- Tra tutti i giochi che è riuscito a vincere potrà scegliere soltanto una busta, che conterrà una quantità in denaro che verrà aggiunta a quelli ottenuti in precedenza
- Viene incoronato campione

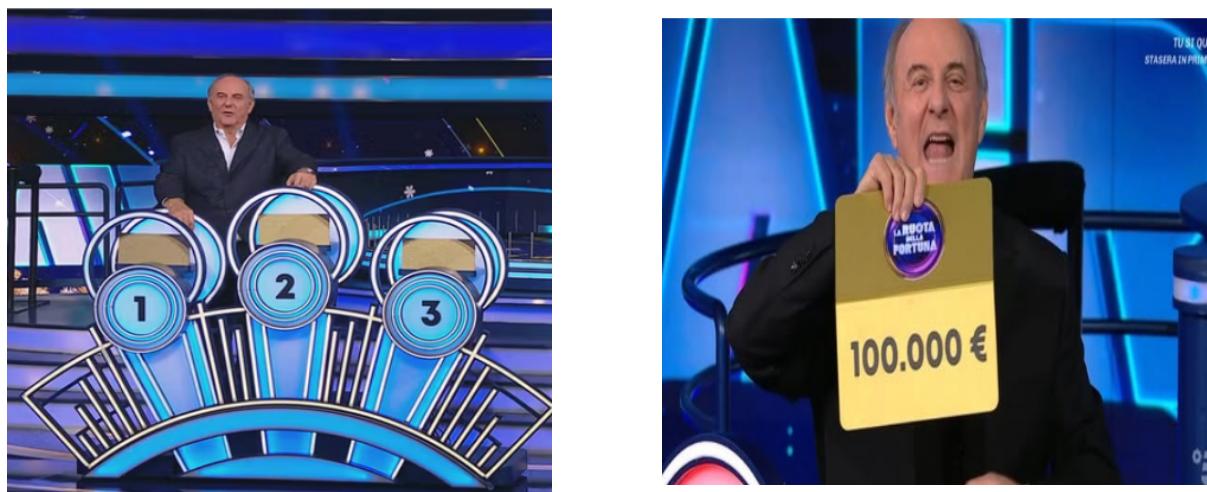


Figura 2: Fase finale: selezione buste e vincitore

### 3 Protocollo di Comunicazione

Il protocollo segue un modello Client-Server Request/Response. Il server rappresenta l'unica autorità sullo stato della partita. Tutti i messaggi utilizzano un formato strutturato con ID a 3 cifre.

#### 3.1 Fase Connessione e Lobby

##### ID 000 - Login giocatore (da client a server)

- ID (3 cifre)
- Nome (20 char)

##### ID 001 - Conferma login (da server a client)

- ID (3 cifre)
- ID giocatore assegnato (1 cifra)
- Esito (bool)

##### ID 002 - Lista giocatori (da server a broadcast)

- ID (3 cifre)
- Numero giocatori (1 cifra)
- Nome Giocatore (20 char) (Questo campo è presente tante volte quanti sono i giocatori)

##### ID 003 - Inizio partita (dal giocatore 1 a server)

- ID (3 cifre)
- Numero turni da giocare(2 cifre)

##### ID 004 - Inizio partita (da server a broadcast)

- ID (3 cifre)
- Numero totale turni (2 cifre)

#### 3.2 Fase Gioco Principale

##### ID 010 - Inizio turno (da server a broadcast)

- ID (3 cifre)
- Frase (52 char)
- Contesto (30 char)

##### ID 011 - Stato giocatori (da server a broadcast)

- ID (3 cifre)
- Numero giocatori (1 cifra)
- Per ogni giocatore: ID (1 cifra) + Soldi salvadanaio (6 cifre) + Soldi turno (6 cifre)

##### ID 012 - Timer turno (da server a client attivo)

- ID (3 cifre)
- Secondi rimasti (3 cifre)

##### ID 015 - Annuncio turno (da server a broadcast)

- ID (3 cifre)
- Giocatore (1 cifra)

### **3.3 Gestione Ruota**

**ID 020 - Girare la ruota (da client a server)**

- ID (3 cifre)

**ID 021 - Risultato della ruota (da server a broadcast)**

- ID (3 cifre)
- Risultato (4 cifre)

### **3.4 Lettere**

**ID 030 - Lettera indovinata (da client a server)**

- ID (3 cifre)
- Lettera (1 char)

**ID 032 - Conferma acquisto vocale (da server a client)**

- ID (3 cifre)
- Conferma (bool)

**ID 033 - Esito lettera (da server a broadcast)**

- ID (3 cifre)
- Lettera (1 char)
- Presente (bool)
- Volte (2 cifre)
- Frase parziale (52 char)
- Soldi guadagnati (6 cifre)

### **3.5 Soluzione**

**ID 040 - Dare soluzione frase (da client a server)**

- ID (3 cifre)
- Frase (52 char)

**ID 041 - Soluzione corretta (da server a broadcast)**

- ID (3 cifre)
- Esito (bool)
- Soldi vinti (6 cifre)

**ID 042 - Annuncio frase (da server a broadcast)**

- ID (3 cifre)
- Frase (52 char)

### **3.6 Gestione Cambio Turno**

**ID 050 - Passo turno (da client a server)**

- ID (3 cifre)

**ID 051 - Cambio turno (da server a broadcast)**

- ID (3 cifre)
- Nuovo giocatore (1 cifra)

**ID 052 - Fine round (da server a broadcast)**

- ID (3 cifre)
- Round corrente (2 cifre)
- Round totali (2 cifre)
- Classifica: ID giocatore (1 cifra) + Soldi (6 cifre) per ogni giocatore

### **3.7 Fase Finale - Ruota delle Meraviglie**

**ID 100 - Inizio fase finale (da server a vincitore)**

- ID (3 cifre)
- ID giocatore vincitore (1 cifra)
- Soldi totali (6 cifre)

**ID 101 - Girare ruota meraviglie (da client a server)**

- ID (3 cifre)

**ID 102 - Inizio minigioco (da server a client)**

- ID (3 cifre)
- Numero gioco (1 cifra) (1, 2, o 3)
- Tipo gioco (2 cifre) (tipo di meccanica)

**ID 103 - Timer minigiochi (da server a client)**

- ID (3 cifre)
- Secondi rimasti (3 cifre)

**ID 104 - Selezione busta (da client a server)**

- ID (3 cifre)
- Numero busta (1 cifra)

**ID 105 - Premio finale (da server a client)**

- ID (3 cifre)
- Importo busta (6 cifre)
- Soldi totali finali (6 cifre)

**ID 106 - Vincitore finale (da server a broadcast)**

- ID (3 cifre)
- ID giocatore (1 cifra)
- Nome giocatore (20 char)
- Soldi finali (6 cifre)

### **3.8 Gestione Eventi e Errori**

**ID 901 - Disconnessione giocatore (da server a broadcast)**

- ID (3 cifre)
- ID giocatore (1 cifra)
- Nome giocatore (20 char)

**ID 902 - Avviso timeout (da server a client)**

- ID (3 cifre)

**ID 903 - Controllo Connessione (bidirezionale)**

- ID (3 cifre)
- Timestamp (10 cifre)

**ID 904 - Bancarotta (da server a broadcast)**

- ID (3 cifre)
- ID giocatore (1 cifra)
- Soldi persi (6 cifre)

## **4 Esempio Iterazione Completa, 2 Giocatori, 1 Turno, Nessun Errore**

### **1. Fase Connessione e Lobby**

1. Client1 effettua login al server
  - 000;Francesco
2. Server conferma login e assegna ID giocatore 1
  - 001;1;1
3. Server invia lista giocatori corrente (broadcast)
  - 002;1;Francesco
4. Client2 effettua login al server
  - 000;Ragazzo
5. Server conferma login e assegna ID giocatore 2
  - 001;2;1
6. Server aggiorna lista giocatori (broadcast a tutti)
  - 002;2;Francesco;Ragazzo
7. Server avvia la partita con 1 turno totale (broadcast)
  - 003;01

### **2. Turno Unico – Client1**

8. Server invia frase mascherata e contesto del turno (broadcast)
  - 010;\*\* \*\*\*\*\* \* \* \* \*\*\*\* \*\*\* \*\*\* \* \* \* \*\*\*\*\* \*\*\*\*\*;  
Insegnamento musicale\*\*\*\*\*
9. Server invia stato corrente di tutti i giocatori (broadcast)
  - 011;2;1;000000;000000;2;000000;000000
10. Server annuncia che è il turno del giocatore 1 (broadcast)
  - 015;1
11. Server invia timer per il turno a Client1 (60 secondi)
  - 012;060
12. Client1 richiede di girare la ruota
  - 020;

13. Server comunica risultato della ruota (broadcast - 300 euro)
  - 021;0300
14. Client1 sceglie la lettera L
  - 030;L
15. Server comunica esito della lettera L (broadcast - presente 3 volte, 900 euro guadagnati)
  - 033;L;1;03;L\* \*\*\*\*\* \* L\* \*\*\* \*\*\*\* \*\*\* L\* \*\*\* \*\*\*\*\*;000900
16. Client1 tenta di risolvere la frase
  - 040;LA TROMBA E LA VITA BEATO CHI LA SUONA\*\*\*\*
17. Server conferma soluzione corretta (broadcast - 900 euro vinti)
  - 041;1;000900
18. Server annuncia la frase completa risolta (broadcast)
  - 042;LA TROMBA E LA VITA BEATO CHI LA SUONA\*\*\*\*
19. Server chiude il round 1 di 1 con classifica finale (broadcast)
  - 052;01;01;1;000900;2;000000

### **3. Fase Finale – Ruota delle Meraviglie**

20. Server avvia fase finale per il vincitore Client1 con 900 euro
  - 100;1;000900
21. Client1 richiede di girare la ruota delle meraviglie
  - 101;
22. Server avvia minigioco 1 (broadcast - tipo meccanica 15, busta 2 associata)
  - 103;1;15;2
23. Server invia timer per i minigiochi a Client1 (60 secondi totali)
  - 105;060
24. Server avvia minigioco 2 (broadcast - tipo meccanica 23, busta 1 associata)
  - 103;2;23;1
25. Server aggiorna timer minigiochi a Client1 (40 secondi rimanenti)
  - 105;040
26. Server avvia minigioco 3 (broadcast - tipo meccanica 08, busta 3 associata)

- 103;3;08;3
27. Server aggiorna timer minigiochi a Client1 (20 secondi rimanenti)
- 105;020
28. Client1 seleziona la busta 2 (tra quelle vinte)
- 106;2
29. Server comunica premio della busta 2 a Client1 (50.000 euro) e totale finale (50.900 euro)
- 107;050000;050900
30. Server annuncia il vincitore finale (broadcast a tutti i giocatori)
- 108;1;FRANCESCO\*\*\*\*\*;050900



## 4.1 Flowchart dell'iterazione

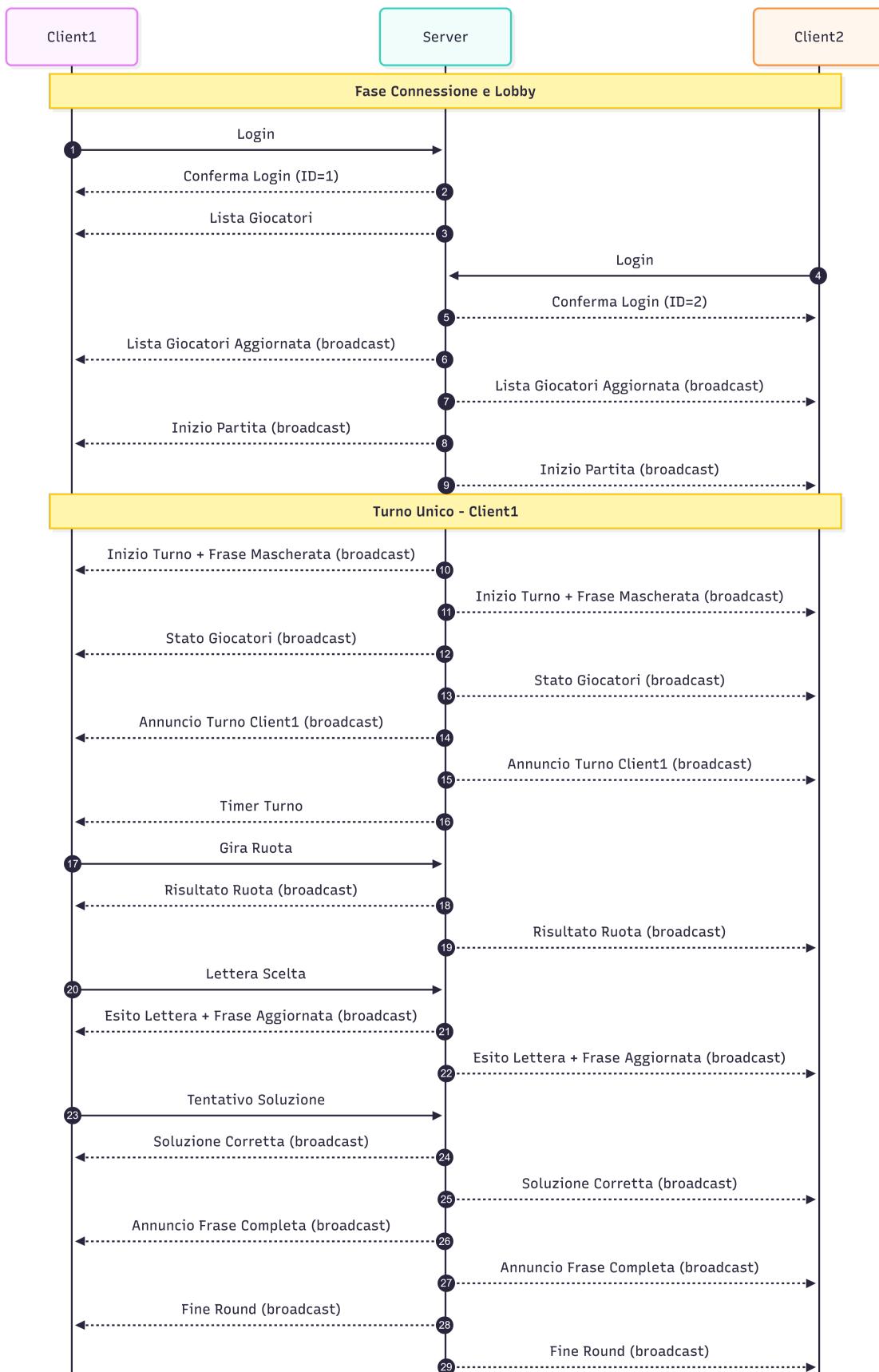


Figura 3: Flowchart: Fase connessione, lobby e turno principale

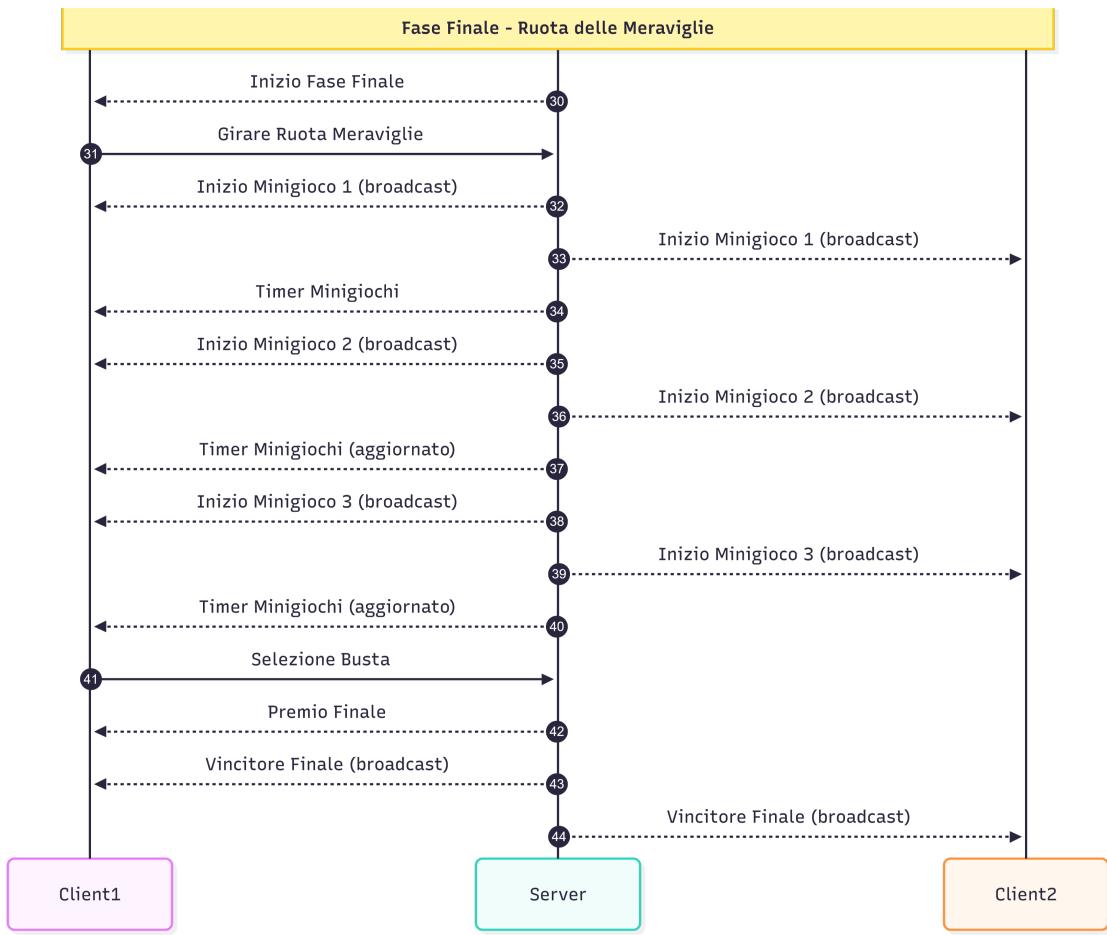


Figura 4: Flowchart: Fase finale - Ruota delle meraviglie

## 5 Architettura del Server

Il server costituisce il nodo centrale del sistema ed è progettato secondo un modello multithread per consentire la gestione concorrente di più partite.

### 5.1 Componenti del Server

Componente	Tipologia	Responsabilità	Motivazione Progettuale
Server	Classe principale	Avvia il ServerSocket e accetta nuove connessioni	Punto di ingresso del sistema
ClientHandler	Thread	Gestisce comunicazione con un singolo client	Permette multithreading e gestione concorrente
GameManager	Classe	Assegna giocatori e gestisce le partite attive	Supporta N partite simultanee
RegistroPartite	Struttura dati	Riferimento a tutte le partite attive	Tracciamento delle sessioni concorrenti
Partita	Classe	Gestione di una singola partita	Isolamento dello stato di gioco
Ruota	Classe	Gestione ruota principale e ruota delle meraviglie	Separazione logica di gioco dalla rete
Frase	Classe	Gestione frasi, lettere e valida soluzioni	Centralizzazione logica delle frasi
Minigame	Classe	Gestione dei 3 mini-giochi della fase finale	Isolamento logica fase finale
Giocatore	Classe	Rappresentazione del giocatore con soldi e stato	Astrazione degli elementi di gioco

### 5.2 Gestione della Concorrenza

Il server utilizza:

- Un thread principale per l'accettazione delle connessioni
- Un ClientHandler dedicato per ogni client
- Una Partita indipendente per ogni partita

I metodi che modificano lo stato della partita sono progettati per evitare race condition tra thread concorrenti.

## 6 Architettura del Client

Il client consente l'interazione con l'utente e la comunicazione asincrona con il server.

### 6.1 Componenti del Client

Componente	Tipologia	Responsabilità	Motivazione Progettuale
Client	Classe principale	Gestione connessione e invio messaggi	Interfaccia principale del giocatore
ServerListener	Thread	Ascolto continuo dei messaggi dal server	Aggiornamento asincrono della partita
InputHandler	Classe gestionale	Gestione input utente (lettere, soluzioni)	Validazione input lato client

### 6.2 Gestione del Multithreading Lato Client

Il client utilizza due flussi di esecuzione:

- Thread principale per input utente
- Thread ServerListener per ricezione messaggi

Questa struttura evita il blocco dell'interfaccia e garantisce aggiornamenti in tempo reale.

© Enio Parlanti