

Specifica Requisiti Software

Morra Cinese Distribuita
Versione 1.0

Indice

1	Descrizione dei Requisiti	1
1.1	Requisiti	1
1.2	Scopo	1
1.3	Finalita' del Progetto	1
1.4	GLOSSARIO	2
1.5	Referenze	2
2	Descrizione generale	3
2.1	Requisiti Funzionali	3
2.2	Funzionamento Sistema	5
2.2.1	Rematch	5
2.2.2	New Match	6
2.3	Descrizione dei programmi	7
2.3.1	GameClient	7
2.3.2	GameServer	7
2.3.3	Enums	7
2.3.4	GameGui	7
2.3.5	Game	7
2.4	UML	8
3	Guida all'Installazione e all'Esecuzione	9
3.1	Requisiti software	9
3.2	Dipendenze Aggiuntive	9
3.3	Installazione	9
3.4	Utilizzo	9
4	Immagini dell'Applicazione	11

Elenco delle figure

1	Diagramma di sequenza - Rematch	5
2	Diagramma di sequenza - New Match	6
3	UML delle classi	8

1 Descrizione dei Requisiti

1.1 Requisiti

All'esame deve essere mostrata l'applicazione in esecuzione e deve essere presentata una relazione che descrive il progetto e deve comprendere:

- Descrizione dei requisiti, ed in particolare delle funzionalita' messe a disposizione (es. tramite SRS);
- Descrizione dell'architettura (ad es. tramite diagramma a blocchi o UML);
- Descrizione dei protocolli usati (client-server o peer-to-peer, ad es. tramite diagrammi UML).

1.2 Scopo

Lo scopo di questo progetto e' sviluppare un sistema distribuito utilizzando il framework Pyro per implementare un gioco di "Sasso, Carta, Forbice" in un ambiente online. Il sistema consente ai giocatori di registrarsi, partecipare al gioco, ricevere gli esiti delle partite, chiedere il rematch o scegliere di partecipare ad un nuovo game, gestendo in maniera robusta tutti i possibili scenari. L'obiettivo principale del progetto e' fornire un'interfaccia di gioco intuitiva e sicura all'interno di un ambiente distribuito.

1.3 Finalita' del Progetto

La finalita' del progetto e' dimostrare l'efficacia dell'uso di un sistema distribuito tramite l'implementazione del gioco "Sasso, Carta, Forbice". Utilizzando il framework Pyro, il sistema si basa su una comunicazione client-server per permettere l'interazione tra i giocatori. L'obiettivo generale e' fornire una soluzione di gioco scalabile ed efficiente nel contesto distribuito del sistema di gioco. Il progetto mira a soddisfare le esigenze di un'esperienza di gioco distribuita, garantendo allo stesso tempo un accesso sicuro e controllato alle funzionalita' del gioco.

1.4 GLOSSARIO

Termine	Definizione
Giocatore	Persona che partecipa al gioco della Morra Cinese
Mossa	Scelta effettuata da un giocatore tra sasso, carta o forbice
Partita	Insieme di mosse effettuate da due giocatori
Al meglio delle 5	Una serie di 5 partite. Vince chi ottiene piu' risultati utili.
Risultato	Esito della partita/serie determinato in base alle mosse dei giocatori
Punteggio	Valore numerico assegnato a ciascun giocatore in base ai risultati delle partite
Punteggio generale	Valore numerico assegnato a ciascun giocatore in base a quante serie ha vinto
Rematch	Richiesta di giocare una nuova partita dopo la fine di una precedente (contro lo stesso avversario)
New Match	Richiesta di giocare una nuova partita dopo la fine di una precedente (contro un altro avversario)
Server	Entita' che coordina le partite tra i giocatori e gestisce i risultati
Client	Il software utilizzato dal giocatore per interagire con il gioco
Interfaccia Utente Grafica (GUI)	L'interfaccia attraverso la quale il giocatore interagisce con il gioco
Registro dei Giocatori	L'insieme di giocatori attivi monitorati dal server
Gioco	Un'istanza specifica di una partita tra due giocatori

Tabella 1: Definizioni delle entita' nel gioco della Morra Cinese

1.5 Referenze

Riferimenti bibliografici

IEEE Computer Society, *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*, 1988.

2 Descrizione generale

Il sistema sviluppato e' un'applicazione di gioco "Sasso, Carta, Forbice" distribuita che funziona secondo un modello client-server. Il server gestisce il gioco, mentre i client possono unirsi, giocare e vedere i risultati del gioco. Inoltre possono richiedere il rematch dopo che la partita e' terminata, oppure possono scegliere di iniziare una nuova partita. E' importante specificare che ogni game si gioca "al meglio delle 5". I client inviano richieste al server e il server risponde a queste richieste. Ad esempio, un client invia una richiesta di registrazione al server inizialmente e poi invia una mossa da giocare. Tutte le interazioni tra il client e il server avvengono tramite richieste e risposte, utilizzando il polling per controllare lo stato del gioco. Questo rende il sistema adatto per un ambiente distribuito, poich  non c'  bisogno di mantenere una connessione persistente tra client e server. E' Pyro5 ad occuparsi della gestione della comunicazione in maniera trasparente.

2.1 Requisiti Funzionali

Questa sezione presenta i requisiti funzionali del sistema, delineando le principali funzionalita' che devono essere implementate nell'applicazione Morra Cinese Distribuita.

1. Registrazione dei giocatori

- Il sistema deve consentire ai giocatori di registrarsi per partecipare al gioco.
- Ogni giocatore deve fornire un nome univoco durante la registrazione.
- Il sistema deve verificare che il nome del giocatore non sia gia' utilizzato da un altro giocatore.
- Il sistema deve garantire che un giocatore registrato sia assegnato ad una partita con un altro giocatore, se possibile.

2. Gestione delle partite

- Il sistema deve gestire piu' partite contemporaneamente.
- Ad una partita devono essere assegnati massimo due giocatori.

3. Effettuare una mossa

- Una volta che 2 giocatori sono stati assegnati alla stessa partita, essi devono poter effettuare una mossa scegliendo tra sasso, carta o forbice.
- Il sistema deve registrare la mossa effettuata da ciascun giocatore.
- Il sistema deve impedire ai giocatori di effettuare una mossa se non hanno un avversario.

4. Determinare il vincitore

- Dopo che entrambi i giocatori hanno effettuato una mossa, il sistema deve determinare il vincitore in base alle regole del gioco "Morra Cinese".
- Il sistema deve comunicare il risultato della partita ai giocatori (e' possibile pareggiare).
- Il sistema deve gestire il risultato sia delle partite singole che compongono una serie, sia della serie stessa.

5. Richiedere un rematch

- Dopo la fine di una partita, i giocatori devono avere la possibilita' di richiedere un rematch per giocare una nuova partita.
- Se entrambi i giocatori richiedono il rematch, il sistema deve avviare una nuova partita, mantenendo il punteggio accumulato.

6. Visualizzare il punteggio

- Il sistema deve tenere traccia del punteggio di ciascun giocatore durante le partite.
- Sia il punteggio parziale che quello totale devono essere mostrati sulla GUI.

7. Interfaccia utente grafica

- Il sistema deve fornire un'interfaccia utente grafica per interagire con il gioco.
- L'interfaccia utente deve visualizzare le informazioni sullo stato attuale del gioco, il punteggio dei giocatori e il risultato delle partite.

8. New Match

- Al termine di una serie di partite, un giocatore può richiedere di effettuare una nuova partita.
- Il sistema deve gestire la richiesta e assegnare il giocatore a una nuova partita.

9. Abilitazione dei tasti di gioco

- Il sistema deve assicurarsi che i tasti per effettuare una mossa (sasso, carta, forbice) siano abilitati solo quando un match con un avversario è stato stabilito e la partita è quindi effettivamente giocabile.

10. Tempo per giocare

- Il sistema deve monitorare il tempo che ogni giocatore impiega per effettuare una mossa.
- Se un giocatore non effettua una mossa entro un tempo prestabilito (120 secondi), il sistema deve considerarlo assente.
- In caso di assenza di un giocatore, il sistema deve rimuovere tale giocatore dal gioco e assegnare la vittoria all'avversario.

11. Gestione delle disconnessioni

- Se un giocatore chiude la finestra di gioco, il sistema deve considerarlo come disconnesso.
- In caso di disconnessione di un giocatore, il sistema deve rimuovere tale giocatore dal gioco e assegnare la vittoria all'avversario.

2.2 Funzionamento Sistema

2.2.1 Rematch

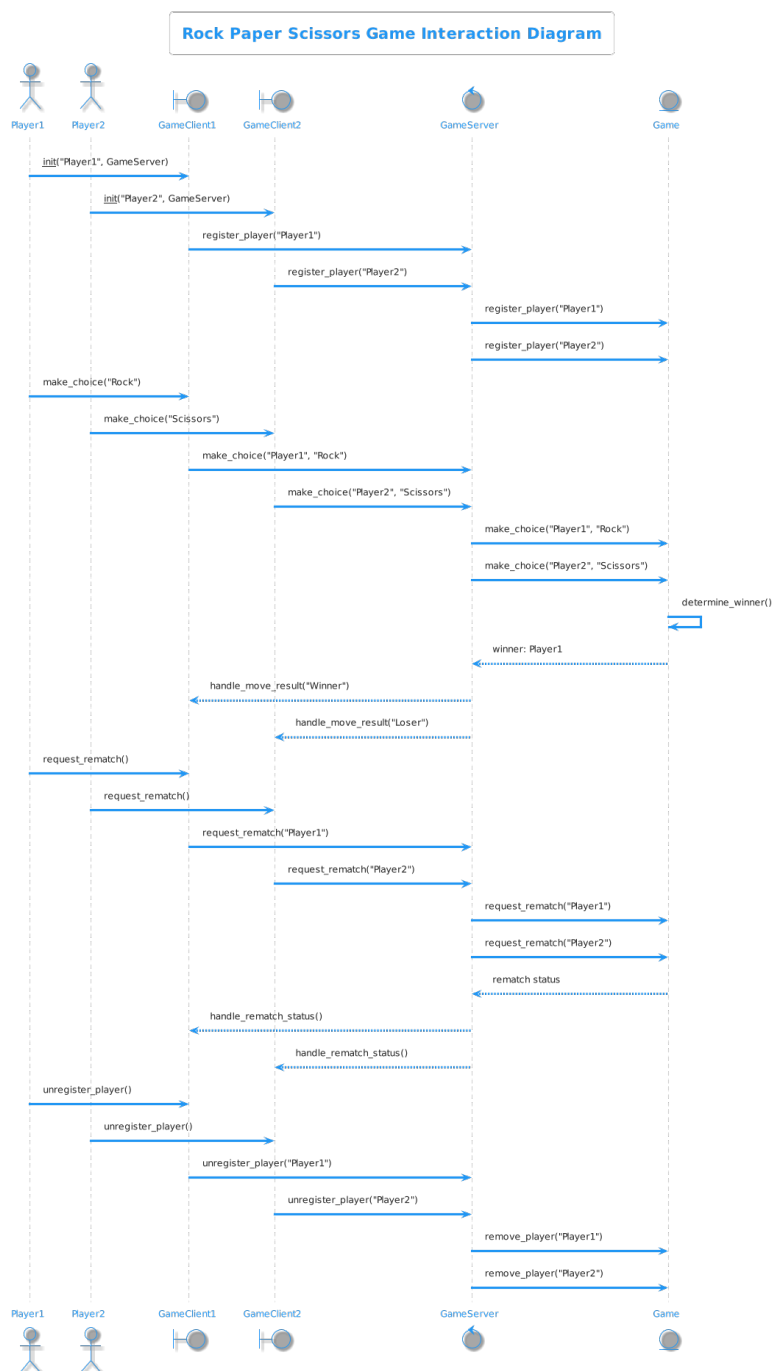


Figura 1: Diagramma di sequenza - Rematch

2.2.2 New Match

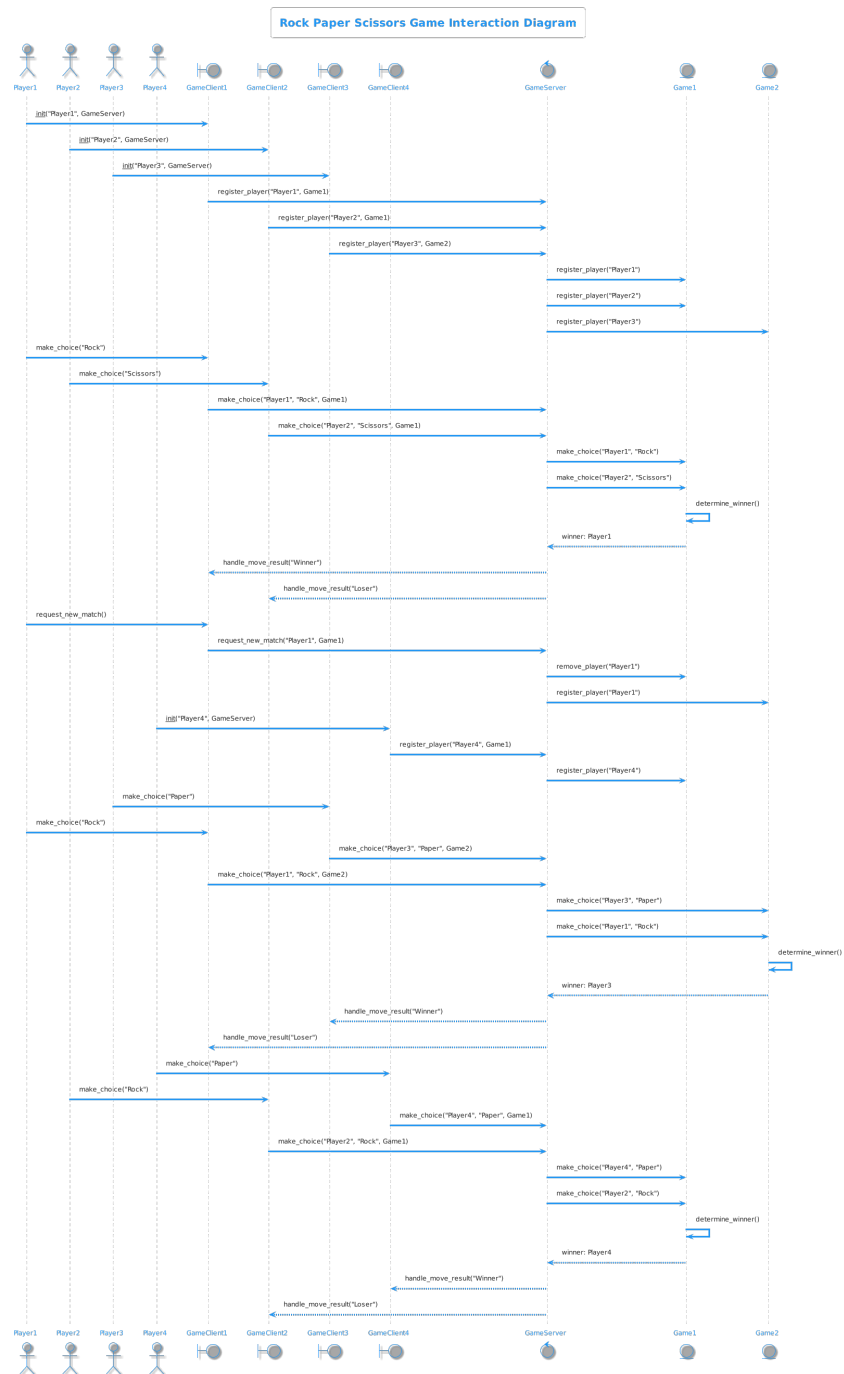


Figura 2: Diagramma di sequenza - New Match

2.3 Descrizione dei programmi

2.3.1 GameClient

E' uno script python denominato "gameclient.py" che:

- Contiene l'inizializzazione del proxy Pyro: ovvero il proxy del server del gioco;
- Gestisce la registrazione dell'utente;
- Fornisce un'interfaccia utente grafica per interagire con il gioco, permettendo all'utente di fare una mossa o richiedere un rematch/nuovo gioco;
- Trasmette le mosse dell'utente al server del gioco e riceve gli aggiornamenti sullo stato del gioco.

Il client all'avvio effettua alcuni setup:

- Individua l'identificatore dell'oggetto remoto utilizzando il Pyro Name Server;
- Crea un oggetto "Proxy" che effettua le chiamate all'oggetto remoto. Una volta ottenuto il "Proxy", il client può utilizzare gli oggetti remoti come se fossero oggetti locali.

2.3.2 GameServer

E' uno script python denominato "server.py" che:

- Gestisce le richieste di gioco da parte dei client, coordinando le partite tra i giocatori;
- Tiene traccia dello stato di ogni partita;
- Aggiorna i client sullo stato del gioco e sul risultato di ogni partita.

2.3.3 Enums

E' uno script python denominato "enums.py" che si occupa di gestire gli enumerati utilizzati nel gioco quali:

- Move: indica le mosse disponibili;
- Results: indica i risultati disponibili;
- MatchStatus: Descrive i possibili stati di un match/serie.

2.3.4 GameGui

E' uno script python denominato "gui.py" che:

- Fornisce un'interfaccia utente grafica per il client, consentendo all'utente di interagire con il gioco;
- Visualizza lo stato corrente del gioco, il punteggio dei giocatori e il risultato delle partite;
- Trasmette le azioni dell'utente al client, come la scelta della mossa o la richiesta di un nuovo gioco.

2.3.5 Game

E' uno script python denominato "game.py" ed e' una classe che incapsula la logica di una partita.

- Utilizzato da "gameserver.py" per gestire le partite.

2.4 UML



Figura 3: UML delle classi

3 Guida all'Installazione e all'Esecuzione

3.1 Requisiti software

I sistemi operativi consigliati per eseguire il gioco della Morra Cinese Distribuita sono MacOS, Linux o Windows, con Python e Pip installati. Le versioni raccomandate sono:

- Python 3.8 o superiori;
- Pip 21.0 o superiori.

3.2 Dipendenze Aggiuntive

Il gioco utilizza le seguenti librerie:

- Pyro5: una libreria Python per la programmazione di oggetti remoti, che permette di creare applicazioni distribuite sfruttando il paradigma client-server. Con Pyro5, e' possibile chiamare metodi di oggetti remoti come se fossero oggetti locali, semplificando la comunicazione tra processi su reti o sistemi diversi.
- PyQt6: una libreria Python che fornisce un set completo di moduli per la creazione di interfacce grafiche e connessioni di rete. PyQt6 e' una binding di Python per il popolare framework Qt, che consente di sviluppare applicazioni cross-platform con una vasta gamma di funzionalita', tra cui finestre, dialoghi, controlli e molto altro. Con PyQt6, e' possibile creare interfacce utente intuitive e interattive per il gioco e gestire in modo efficiente le connessioni di rete necessarie per il funzionamento del software.

Queste librerie giocano un ruolo fondamentale nel consentire al gioco di funzionare in modo efficace, fornendo le funzionalita' necessarie per la comunicazione tra i giocatori e per la gestione delle interfacce utente.

3.3 Installazione

Per creare un ambiente virtuale contenente le librerie necessarie per il programma, eseguire i seguenti comandi:

```
python3 -m venv venv
source venv/bin/activate (Linux/MacOS)
venv\Scripts\activate (Windows)
```

Per installare le librerie necessarie, e' possibile utilizzare il file "requirements.txt" che contiene l'elenco delle librerie e delle rispettive versioni utilizzate durante lo sviluppo, attraverso il comando:

```
pip install -r requirements.txt
```

Oppure si puo' procedere manualmente tramite il seguente comando:

```
pip install pyro5 PyQt6
```

3.4 Utilizzo

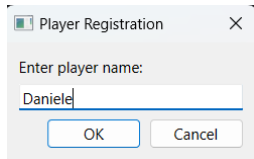
Per eseguire correttamente l'applicazione Morra Cinese, segui i seguenti passaggi:

1. Assicurati di avere installato Python, Pyro5 e PyQt6 sul tuo computer (come descritto in precedenza).
2. Clona o scarica il repository del progetto.
3. Apri il terminale o la finestra del prompt dei comandi e posizionali nella directory del progetto.

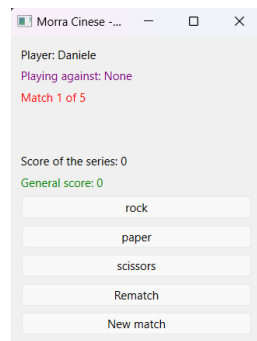
4. Esegui il file `gameserver.py` per avviare il server di gioco.
5. Per ogni giocatore che desidera partecipare al gioco, apri un nuovo terminale o finestra del prompt dei comandi e posizionali nella directory del progetto.
6. Esegui il file `gameclient.py` per avviare il client del giocatore (ricordarsi che sono necessari almeno due giocatori per iniziare una partita).
7. Segui le istruzioni visualizzate sull'interfaccia del client per giocare alla Morra Cinese.

4 Immagini dell'Applicazione

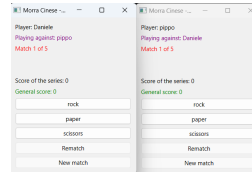
Di seguito e' mostrato il funzionamento dell'applicazione Morra Cinese tramite immagini:



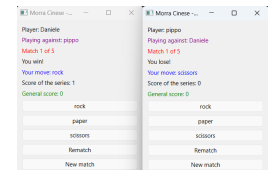
(a) Client: Registrazione 1/2



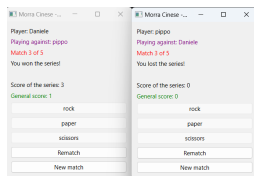
(b) Client: Registrazione 2/2



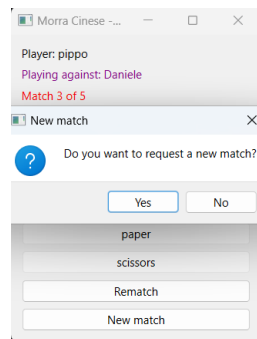
(c) Client: Accoppiamento con un altro giocatore



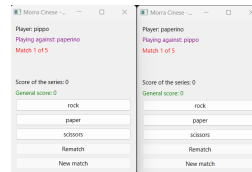
(d) Client: Mossa e risultati singola partita



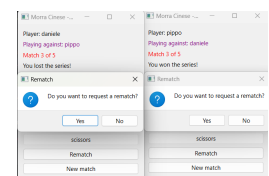
(a) Client: Risultati serie



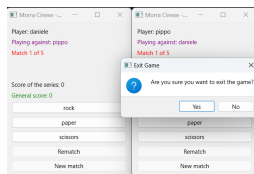
(b) Client: Richiesta nuova partita



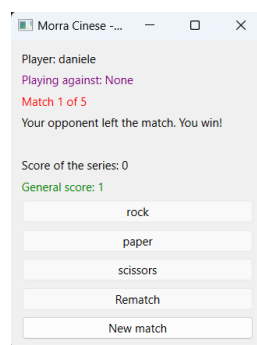
(c) Client: Accoppiamento con un nuovo giocatore (dopo richiesta nuova partita)



(d) Client: Richiesta rematch



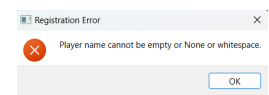
(a) Client: Chiusura client e abbandono del gioco



(b) Client: Vincita match per abbandono del gioco



(c) Client: Errore registrazione nome 1/2



(d) Client: Errore registrazione nome 2/2

Le immagini mostrano quanto descritto finora in questa documentazione. Queste schermate consentono ai giocatori di interagire con il gioco in modo intuitivo e godere di un'esperienza di gioco coinvolgente.