1. Linguaggio di Programmazione Scelto

Il linguaggio scelto per questo progetto è Java per la sua gestione dell’interfaccia grafica e per la gestione di socket, la sicurezza e la creazione di applicazioni client-server

### 2. Librerie per la Comunicazione

Per la gestione della comunicazione client-server utilizzeremo le seguenti librerie Java:

* java.net.ServerSocket e java.net.Socket: Queste classi sono utilizzate per gestire la comunicazione client-server tramite il protocollo TCP. ServerSocket permetterà al server di ascoltare le connessioni in entrata, mentre Socket sarà utilizzato dai client per stabilire una connessione con il server e inviare/ricevere dati.
* JavaFX: Utilizzeremo questa libreria per sviluppare l'interfaccia grafica dell'applicazione, in quanto fornisce strumenti per creare interfacce moderne e graficamente avanzate, che miglioreranno l'esperienza dell'utente durante il gioco.
* Java Cryptography Extension (JCE): La libreria javax.crypto sarà utilizzata per implementare la crittografia dei messaggi scambiati tra client e server, garantendo così la sicurezza dei dati.

### 3. Funzionalità Utente

Le funzionalità principali dell'applicativo sono le seguenti:

* Gestione delle partite: L'applicazione supporta partite tra due giocatori, con aggiornamento dello stato di gioco in tempo reale. La griglia di gioco sarà visibile per entrambi i giocatori, e l'aggiornamento avverrà immediatamente quando un giocatore effettua una mossa.
* Validazione delle mosse: Il sistema controllerà che ogni mossa sia valida (ad esempio, impedendo che i giocatori mettano un simbolo in una cella già occupata o che agiscano fuori turno).
* Sicurezza della comunicazione: Per proteggere i dati sensibili, tutte le comunicazioni tra client e server saranno criptate end-to-end. In particolare, le mosse effettuate dai giocatori, così come i messaggi tra di loro, saranno cifrati.
* Lobby per il matchmaking: I giocatori potranno accedere a una lobby per cercare avversari e avviare una partita. La lobby permetterà la gestione delle richieste di connessione e la selezione di un giocatore con cui sfidarsi. Nella lobby i giocatori potranno decidere se avviare il matchmaking pubblico e attendere in linea che un altro giocatore si connetta alla partita oppure potranno decidere di avviare una partita privata tramite una password. Infine ci sarà la possibilità di giocare offline contro un bot con diversi livelli di difficoltà.

### 4. Protocolli di Rete

Per la comunicazione tra client e server, utilizzeremo il protocollo TCP, poiché garantisce l'integrità e l'ordine dei dati. TCP è ideale per applicazioni come i giochi, dove è essenziale che ogni mossa venga ricevuta correttamente e nell'ordine giusto.

I pacchetti inviati tra client e server conterranno dati crittografati, in modo da proteggere le informazioni trasmesse (ad esempio, le mosse dei giocatori e i messaggi di chat).

### 5. Sicurezza e Crittografia

Per garantire la sicurezza dei dati, utilizzeremo i seguenti algoritmi di crittografia:

* Algoritmo di crittografia AES (Advanced Encryption Standard): Le mosse dei giocatori, lo stato della partita, e i messaggi di chat saranno protetti utilizzando AES con chiavi a 256 bit. Questo garantirà un livello elevato di sicurezza per i dati sensibili.
* Scambio di chiavi: Per lo scambio sicuro delle chiavi AES, utilizzeremo RSA (2048 bit) per cifrare la chiave simmetrica AES. RSA garantirà che solo il destinatario del messaggio possa decifrare la chiave e, successivamente, i dati criptati.
* Dati crittografati: Le informazioni sensibili, come lo stato della partita (mosse effettuate, vincitore, ecc.), saranno tutte crittografate prima di essere inviate tra client e server, impedendo che vengano intercettate o modificate da attori malintenzionati.

6. ISTRUZIONI AGGIUNTIVE

Componenti principali:

* Server
  + Gestisce le connessioni dei client.
  + Smista le partite tra i giocatori.
  + Garantisce la correttezza delle mosse e la gestione del flusso di gioco.
  + Si occupa della crittografia e decrittografia dei dati.
* Client
  + Invia le mosse al server.
  + Visualizza lo stato della partita aggiornato.
  + Gestisce l'interfaccia utente e la comunicazione crittografata con il server.