

Projekt: Sieciowa turowa gra logiczna - szachy

Cel Projektu:

Celem projektu jest stworzenie sieciowej gry logicznej w szachy, w której serwer obsługuje jednocześnie wiele równoległych rozgrywek między parami graczy. Gra ma obejmować walidację ruchów po stronie serwera oraz implementację mechanizmu wykrywania i obsługi rozłączenia gracza.

Technologie i Biblioteki:

C++: Język programowania do implementacji logiki gry oraz zarządzania połączeniami sieciowymi.

Boost.Asio: Architektura klient-serwer została opracowana przy użyciu biblioteki Boost.Asio, aby zapewnić asynchroniczne operacje We / Wy i zarządzanie operacjami sieciowymi. Umożliwia równoczesne obsługiwanie wielu klientów.

ImGui: Interfejs graficzny dla narzędzi deweloperskich, umożliwiający tworzenie intuicyjnych i interaktywnych interfejsów użytkownika.

SFML (Simple and Fast Multimedia Library): Biblioteka do obsługi grafiki i okien w celu stworzenia interaktywnego interfejsu użytkownika i obsługi zdarzeń.

Architektura i design Projektu:

Projekt będzie składał się z trzech głównych komponentów: serwera, klienta oraz logiki gry. Aplikacja opiera się na architekturze asynchronicznej zbudowanej na bazie klas Session Base, Session i Listener. Interakcja komponentów jest zapewniona przez użycie gniazd TCP i wiadomości za pośrednictwem klasy Message.

Protokół przesyłania wiadomości:

Klasa Message implementuje prosty protokół przesyłania wiadomości. Używa przeciążonych operatorów do serializacji i deserializacji danych.

Obsługa zapytań:

Żądania są przetwarzane asynchronicznie w klasie Session Base. Przykład obsługi żądania i wysłania odpowiedzi znajduje się w klasie Session.

Klasy Session I Listener:

Klasy Session I Listener zapewniają podstawową architekturę do obsługi sesji i nasłuchiwanie połączeń.

Obsługa Błędów i Testowanie:

Projekt będzie zawierał system obsługi błędów, który będzie monitorował i raportował nieprawidłowości związane z połączeniem, ruchami graczy oraz innymi aspektami gry. Błędy są obsługiwane za pomocą `sys::error_code`. Do zgłaszania błędów używana jest funkcja `report Return`.

Zastosowanie Grafiki i Interfejsu Użytkownika:

SFML i ImGui będą używane do stworzenia atrakcyjnego interfejsu użytkownika, prezentującego planszę do gry, informacje o ruchach, a także umożliwiającego interakcję z graczami.

Serwer:

Obsługuje wielu klientów jednocześnie poprzez użycie Boost.Asio do asynchronicznej komunikacji. Zarządza rozgrywkami między parami graczy, przydzielając odpowiednie zasoby dla każdej gry. Realizuje walidację ruchów po stronie serwera, zapobiegając nielegalnym ruchom. Monitoruje i obsługuje przypadki rozłączenia gracza podczas trwającej rozgrywki.

Klient:

Komunikuje się z serwerem za pomocą Boost.Asio, przekazując ruchy i otrzymując aktualny stan gry. Wykorzystuje ImGui do stworzenia intuicyjnego interfejsu użytkownika, umożliwiającego graczom podejmowanie decyzji i monitorowanie stanu gry. Reaguje na sygnały od serwera dotyczące rozłączenia przeciwnika czy zakończenia rozgrywki.

Logika Gry:

Implementuje zasady gry w szachy, w tym walidację ruchów i aktualizację stanu planszy. Komunikuje się z serwerem i klientem, przysyłając informacje o ruchach, aktualnym stanie planszy i zakończeniu rozgrywki.

Podsumowanie:

Projekt ma na celu stworzenie kompleksowej sieciowej gry w szachy, która oferuje użytkownikom możliwość gry w czasie rzeczywistym. Architektura klient-serwer implementuje komunikację asynchroniczną przy użyciu biblioteki Boost.Asio. Architektura aplikacji i protokół przesyłania wiadomości umożliwiają wydajną komunikację między klientami a serwerem. Obejmuje on implementację złożonych mechanizmów sieciowych, walidację ruchów, obsługę rozłączeń oraz przyjazny interfejs użytkownika.