

Sprawozdanie SK2

Sieciowa Turowa gra logiczna „Kółko i krzyżyk 4x4”

Informacje dotyczące Implementacji projektu:

Projekt został zrealizowany w architekturze klient-serwer, korzystając z protokołu TCP oraz interfejsu gniazd BSD. Serwer został zaimplementowany w języku C++, natomiast klient w Pythonie. Komunikacja między nimi odbywa się w trybie połączeniowym z wykorzystaniem gniazd BSD. W celu efektywnej obsługi wielu równoległych rozgrywek między parami graczy, serwer wykorzystuje wielowątkowość, obsługując każde połączenie z klientem jako osobny wątek.

Serwer odpowiada za połączenia między dwójką klientów w celu rozpoczęcia i przeprowadzenia gry „Kółko i krzyżyk 4x4”. Odpowiada również za sprawdzenie poprawności ruchów wykonanych przez klientów, powiadamianie uczestników o rezultacie rozgrywki oraz wykrywanie i obsługę sytuacji, gdy któryś z klientów zostanie rozłączony.

Klient jest odpowiedzialny za nawiązywanie połączenia z serwerem, prezentowanie bieżącego stanu gry w interfejsu wiersza poleceń (CLI) oraz umożliwianie graczowi dokonywania ruchów w trakcie rozgrywki.

Ograniczenia:

Serwer może obsługiwać maksymalnie n równoległych rozgrywek, gdzie wartość n została ustawiona na 5.

Opis działania Serwera:

Na początku działania, serwer inicjuje obiekt klasy "Games", który przechowuje tablicę składającą się z n obiektów klasy "Game". Klasa "Game" zawiera implementację metod niezbędnych do przeprowadzenia rozgrywki w grze "Kółko i krzyżyk 4x4". Natomiast klasa "Games" implementuje metody umożliwiające połączenie graczy w ramach rozgrywki.

Następnie, serwer rozpoczyna nasłuchiwanie na połączenia graczy. W momencie podłączenia się gracza do serwera, serwer sprawdza czy istnieje już gra z oczekującym graczem w kolejce. Jeśli taki gracz istnieje, uruchamiany jest wątek, który przyjmuje jako argument tablicę z gniazdem połączenia klienta oraz indeksem obiektu rozgrywki z tablicy rozgrywek.

W przypadku braku oczekującego gracza, tablica z przechowywanymi rozgrywkami jest przeszukiwana w poszukiwaniu pierwszej dostępnej gry i ustawieniu jej w kolejce. Jeśli wszystkie gry są już zajęte, klient pozostaje w oczekiwaniu, aż jakaś gra zakończy się, aby móc dołączyć do niej.

Po tym, jak gracze podłączą się do gry, serwer przesyła do każdego z graczy następujące informacje:

1. **Status ruchu:** Poprawny, Niepoprawny, Wygrany, Przegrany, Remis, Rozłączenie oponenta.
2. **Kto teraz przeprowadza ruch:** Pierwszy albo drugi gracz.

3. **Strona gracza:** Kółko albo Krzyżyk
4. **Zakodowana tablica obecnego stanu gry:** Każdy kolejny symbol reprezentuje element bieżącej tablicy rozgrywki w grze „Kółko i krzyżyk”, gdzie 'X' oznacza krzyżyk, 'O' oznacza kółko, a ' ' (spacja) oznacza puste pole.

Serwer z góry zakłada, że gracz, który pierwszy podłączył się do gry jest Krzyżykiem, a drugi gracz jest Kółkiem.

Po przesłaniu tych informacji, serwer w pętli będzie oczekiwał na kolejne ruchy od graczy, aż gra się nie zakończy lub jeden z graczy się nie rozłączy. Po otrzymaniu wiadomości od gracza, serwer sprawdzi, czy jest to dopuszczalny ruch. Jeśli ruch nie jest dopuszczalny, serwer wyśle do gracza wiadomość z prośbą o powtórny próbę ruchu. Jeżeli ruch jest poprawny, serwer dokona tego ruchu, sprawdzi, czy gracz wygrał lub doprowadził do remisu, a następnie wyśle informacje z odnowionym stanem gry do obu graczy. Informacje o zakończeniu i wyniku gry zostaną przesłane do obu graczy, a sama gra zostanie usunięta z pamięci. Połączenia z klientami zostaną również zamknięte.

W przypadku rozłączenia jednego z graczy, serwer wyśle komunikat do oponenta z tą informacją, po czym gra zostanie usunięta z pamięci, a połączenie z klientami zostaną zamknięte.

Opis działania Klienta:

Klient inicjuje połączenie z serwerem, wykorzystując określony adres IP i numer portu. Oczekuje na komunikat od serwera dotyczący rozpoczęcia gry oraz informacji specyficznych dla danej rozgrywki i gracza. Następnie, w pętli, klient odbiera od serwera aktualne informacje o stanie gry, prezentuje go i wysyła kolejne ruchy, korzystając z interfejsu wiersza poleceń (CLI). W sytuacji otrzymania komunikatu o wygranej, przegranej, remisie lub rozłączeniu przeciwnika, klient zamyka połączenie z serwerem.