

Sprawozdanie sk2

Gra w statki

1. Opis projektu (0.5 strony)

Projekt to sieciowa gra w statki napisana w C++ z wykorzystaniem API BSD-socket. Gra pozwala dwóm graczom rywalizować na planszy 10x10, w której umieszczają swoje statki i starają się zatopić statki przeciwnika. Gracze komunikują się z serwerem, aby ustawić swoje statki, a następnie przeprowadzają ataki na planszy przeciwnika, wprowadzając współrzędne, które określają, gdzie będą strzelać. Gra jest rozgrywana w trybie multiplayer, z obsługą wielowątkowości na serwerze.

Użyte technologie:

BSD Socket API – Użyte do komunikacji sieciowej między serwerem a klientami. Umożliwia otwieranie gniazd (sockets) i przesyłanie danych przez sieć.

Wielowątkowość – Serwer obsługuje wielu graczy w równoczesnych wątkach, co pozwala na równoległą rozgrywkę kilku meczy.

2. Opis komunikacji pomiędzy serwerem i klientem (0.5 strony, może być schemat/rysunek)

- **Połączenie TCP/IP:** Serwer oczekuje na połączenia od klientów (graczy) i akceptuje je, tworząc osobne wątki do obsługi każdej pary graczy. Serwer działa w trybie blokującym, a po zaakceptowaniu połączenia komunikacja między serwerem a klientami odbywa się za pomocą gniazd.
- **Komunikacja:**
 1. **Ustawienie planszy:** Gracze są proszeni o umieszczenie swoich statków na planszy poprzez wprowadzenie współrzędnych i kierunku (poziomo/pionowo). Serwer wysyła komunikaty informujące gracza o wymaganiach i weryfikuje poprawność wprowadzanych danych.
 2. **Atakowanie:** Po ustawieniu plansz, gracze na zmianę wykonują ataki, wpisując współrzędne (np. A,5). Serwer sprawdza, czy atak trafił w statek przeciwnika i informuje graczy o wyniku.
 3. **Stan gry:** Po każdym ruchu, serwer wysyła obie plansze do gracza: należącą do gracza i atakowaną.
- **Zabezpieczenia:** Komunikacja jest zabezpieczona poprzez odpowiednie sprawdzenie formatu wprowadzanych danych, a także obsługę przypadków, takich jak rozłączenie gracza w trakcie gry.

3. Podsumowanie (0.5-1 strona)

Implementacja projektu składa się z trzech części – plik serwera obsługujący całą rozgrywkę oraz walidację ruchów, plik klienta obsługujący połączenie do serwera i odbieranie wiadomości oraz plik common z funkcjami i strukturami niezbędnymi do logiki gry, z których później korzysta serwer. Serwer sprawdza, czy każdy ruch gracza jest zgodny z zasadami gry oraz koordynuje przesyłanie odpowiednich wiadomości do odpowiednich klientów z racji tego, że każdy klient widzi inne plansze.

Co sprawiło trudność:

- **Poprawna walidacja stanów gry** – Serwer czasami nie dostawał poprawnie komunikatów, przez co gra zapętleła się w jednym stanie.
- **Odsyłanie poprawnych informacji do odpowiednich klientów** – Dotyczyło to głównie plansz graczy, czasem po zmianie tury plansze resetowały się lub gracze dostawali plansze swojego przeciwnika.
- **Integracja interfejsu graficznego z komunikatami serwera** – W pierwotnej wersji gry podjęta była próba wprowadzenia GUI, jednak połączenie komunikatów z interfejsem graficznym było problematyczne, komunikaty były źle przekazywane i źle łączone z interfejsem, co sprawiło, że próba naprawienia GUI byłaby bardziej pracochłonna niż cała reszta implementacji, dlatego też zrezygnowano z tej opcji na rzecz interfejsu w terminalu, który jest również przejrzysty, ale posługuje się komunikatami tekstowymi.