Sebastian Zgoda 147919

Sprawozdanie sk2

Battleships-game

1. Opis projektu

Projekt to sieciowa gra "Statki" (Battleships) realizowana w architekturze Klient-Serwer. Aplikacja umożliwia rozgrywkę dwóch graczy za pośrednictwem sieci TCP/IP. Gra oferuje dwa tryby interfejsu: tekstowy (konsolowy) oraz graficzny (GUI). System obsługuje również czat w czasie rzeczywistym między graczami. Serwer został zaprojektowany w sposób wielowątkowy, co pozwala na jednoczesną obsługę wielu par graczy (rozgrywek) na jednym porcie. Logika gry jest zaimplementowana po stronie serwera (model Server-Authoritative), co zapobiega oszustwom (np. modyfikacji stanu planszy przez klienta).

Użyte technologie:

- C++ 17
- SFML
- Gniazda sys/socket.h, netinet/in.h
- Wielowątkowość
- Makefile

2. Opis komunikacji pomiędzy serwerem i klientem

Serwer nasłuchuje na określonym porcie. Po połączeniu dwóch klientów, tworzy dla nich osobny wątek (sesję gry). Serwer pełni rolę arbitra – przechowuje plansze obu graczy, waliduje strzały i odsyła wyniki.
Protokół (Lista najważniejszych komend):

1. Inicjalizacja:
    - SEND_BOARD – Żądanie serwera o przesłanie układu statków.
    - [Ciąg 100 cyfr] – Klient wysyła zserializowaną planszę (0-woda, 1-4 statki).
    - START [NazwaPrzeciwnika] – Serwer informuje o rozpoczęciu gry.
2. Rozgrywka:
    - TURN / WAIT – Serwer steruje turą (informuje klienta, czy może wykonać ruch).
    - SHOT [x] [y] – Klient wysyła współrzędne strzału.
    - RESULT [HIT/MISS] [x] [y] – Serwer informuje strzelającego o wyniku (Trafienie/ Pudło).
    - OPPONENT_SHOT [x] [y] – Serwer informuje drugiego gracza, że został ostrzelany w danym polu.
    - WIN / LOSE – Komunikaty kończące grę.
3. Inne:
    - CHAT [wiadomość] – Przesyłanie wiadomości tekstowych między graczami (obsługiwane asynchronicznie, niezależnie od tury).

3. Podsumowanie

Projekt został zrealizowany z naciskiem na przenośność kodu (cross-platform). Kod źródłowy kompiluje się zarówno na systemie Linux (środowisko laboratoryjne), jak i macOS. Osiągnięto to dzięki dyrektywom preprocesora (wykrywanie wersji biblioteki SFML 2.x vs 3.0.0) oraz elastycznemu plikowi Makefile.

Kluczowym elementem jest pełna separacja logiki sieciowej od interfejsu użytkownika. Klient posiada osobny wątek (networkLoop), który w tle odbiera komunikaty od serwera i umieszcza je w bezpiecznej wątkowo kolejce (std::mutex). Główny wątek renderujący GUI pobiera te zdarzenia i aktualizuje stan planszy, co zapobiega "zamrażaniu" okna aplikacji podczas oczekiwania na pakiety sieciowe.

Zastosowano mechanizm Server-Authoritative. Klient nie decyduje o tym, czy został trafiony. To serwer posiada kopię plansz i po otrzymaniu współrzędnych sam sprawdza wynik, odsyłając go do obu graczy. Zapobiega to podstawowym metodom oszukiwania.

Co sprawiło trudność:

1. Kompilacja międzyplatformowa: Największym wyzwaniem było dostosowanie kodu do różnych wersji biblioteki SFML (zmiany w API między wersją 2 a 3, np. obsługa sf::Event czy kątów rotacji) oraz różnice w ścieżkach do bibliotek (/usr/include vs /opt/homebrew).

2. Synchronizacja gniazd TCP: Problem "sklejania" pakietów (TCP stream). Konieczne było zaimplementowanie bufora po stronie klienta, który dokleja odebrane fragmenty danych i dzieli je na poprawne komendy dopiero po napotkaniu znaku nowej linii.

3. Konfiguracja środowiska testowego: Problemy z komunikacją sieciową między maszyną wirtualną (Linux) a gospodarzem (macOS) wymagały odpowiedniej konfiguracji trybu sieciowego (*Bridged Adapter*) oraz diagnostyki zapory sieciowej (Firewall), która blokowała połączenia przychodzące.

4. Błędy linkera: Problemy z wielokrotną definicją funkcji (duplicate symbol) przy współdzieleniu kodu między plikami, co rozwiązano poprzez odpowiednie użycie słowa kluczowego static lub restrukturyzację nagłówków.

Projekt zawiera tryb gry lokalnej i tryb gry online. Pliki, które są odpowiedzialne za rozgrywkę online i zostały pokazane na zajęciach to:

- server.cpp
- client.cpp
- guigame.h i guigame.cpp
- gra.h i gra.cpp
- plansza.h i plansza.cpp

Projekt można znaleźć na github: https://github.com/Seizwell/Battleships-game