

# Raport techniczny projektu

## **Sieciowa gra w statki**

Projekt wykonał:

Wojciech Midura

Akademia Górniczo-Hutnicza  
Elektronika i Telekomunikacja

## **Wstęp:**

Projekt realizuje klasyczną grę w statki (Battleship) w architekturze klient-serwer. Gra umożliwia dwóm graczom rozgrywkę przez sieć, gdzie każdy z graczy ma swoją planszę z ustawionymi statkami i na zmianę próbuje trafić w statki przeciwnika. Serwer zarządza logiką gry, przesyłaniem danych między graczami oraz synchronizacją stanu gry.

## **Architektura Systemu:**

System składa się z dwóch głównych komponentów:

1. Serwer – zarządza logiką gry, przetwarza ruchy graczy, sprawdza poprawność strzałów, aktualizuje stan gry i komunikuje się z klientami.
2. Klient – odpowiada za interakcję z użytkownikiem, wyświetlanie planszy, odbieranie i wysyłanie danych do serwera.

Komunikacja między klientem a serwerem odbywa się za pomocą gniazd sieciowych (sockets) z wykorzystaniem protokołu TCP/IP.

## **Struktura Danych:**

- **Pola:**
  - int id – unikalny identyfikator gracza.
  - char username[50] – nazwa użytkownika.
  - int comand – flaga określająca typ komunikatu (np. INTRODUCTION, SHOT, BOARD, GAMEOVER).
  - char board[10][10] – plansza gry (10x10) przechowująca stan statków i strzałów.
  - char x, y – współrzędne strzału.

## **Struktura Client**

- **Pola:**
  - int socket – deskryptor gniazda klienta.
  - ClientData data – dane klienta.

## **Logika Serwera:**

### **Główne Funkcje Serwera**

1. **handleClient:**
  - Obsługuje połączenie z klientem.
  - Przetwarza komunikaty od klienta na podstawie flagi comand.
  - Zarządza logiką gry, w tym:
    - Przydzielaniem identyfikatorów graczom.
    - Generowaniem plansz z statkami.
    - Sprawdzaniem trafień i aktualizacją planszy.
    - Wysyłaniem odpowiedzi do klientów.

## 2. **set\_board:**

- Generuje losowe ustawienie statków na planszy.
- Statki mają różne rozmiary: 4-masztowiec (A), 3-masztowiec (B), 2-masztowiec (C), 1-masztowiec (D).

## 3. **is\_valid\_position i place\_ship:**

- Sprawdzają, czy statek może zostać umieszczony na planszy, oraz umieszczają go w odpowiednim miejscu.

## 4. **handleWaitforshot:**

- Monitoruje czas oczekiwania na strzał gracza. Jeśli gracz nie odda strzału w określonym czasie, serwer przypomina mu o konieczności wykonania ruchu.

### **Flagi Komunikatów**

- **INTRODUCTION** – inicjalizacja połączenia, przydzielenie identyfikatora graczowi.
- **BOARD** – przesyłanie planszy do klienta.
- **GIVESHOT** – informacja, że gracz może oddać strzał.
- **SHOT** – przesyłanie współrzędnych strzału do serwera.
- **GAMEOVER** – koniec gry, wyłonienie zwycięzcy.
- **TOOMANY** – informacja o przekroczeniu maksymalnej liczby graczy (2).

### **Logika Klienta:**

#### **Główne Funkcje Klienta**

## 1. **receiveMessages:**

- Odbiera komunikaty od serwera w osobnym wątku.
- Aktualizuje stan gry na podstawie otrzymanych danych (np. planszę przeciwnika, informacje o trafieniach).

## 2. **Drawboard:**

- Rysuje planszę gry w terminalu, pokazując zarówno stan własnych statków, jak i strzały przeciwnika.

## 3. **Główna Pętla:**

- Czeka na sygnał od serwera (ready\_for\_command), aby umożliwić graczowi oddanie strzału.
- Wprowadzone przez gracza współrzędne są walidowane i wysyłane do serwera.

### **Komunikacja Sieciowa:**

- **Port:** 8080.
- **Protokół:** TCP/IP.
- **Adresacja:** Serwer nasłuchuje na wszystkich interfejsach (0.0.0.0), a klient łączy się z podanym adresem IP serwera.
- **Synchronizacja:** Wykorzystanie mutexów (`pthread_mutex_t`) do zabezpieczenia dostępu do współdzielonych danych (np. listy klientów).

### **Obsługa Błędów i Wyjątków:**

#### **1. SIGPIPE:**

- Serwer i klient mają zdefiniowane handlers dla sygnału SIGPIPE, który jest generowany w przypadku próby zapisu do zamkniętego gniazda.

#### **2. Przekroczenie liczby graczy:**

- Jeśli próbuje połączyć się trzeci gracz, serwer odsyła komunikat TOOMANY i zamyka połączenie.

#### **3. Utrata połączenia:**

- Jeśli klient rozłączy się podczas gry, serwer wysyła pozostałemu graczowi komunikat GAMEOVER.

### **Interfejs Użytkownika:**

- Plansza gry jest wyświetlana w terminalu w formie tekstowej.
- Gracz wprowadza współrzędne strzału w formacie literowo-liczbowym (np. a0, b5)