

## **Sprawozdanie z Projektu: "Wisielec"**

### **Cel Projektu**

Celem projektu było stworzenie sieciowej wersji gry logicznej "Wisielec", w której gracze mogą łączyć się z serwerem, aby odgadywać litery i odkrywać słowa. Gra została zaprojektowana tak, aby wielu graczy mogło współzawodniczyć ze sobą w czasie rzeczywistym, co dodaje elementy rywalizacji i współpracy.

### **Opis Techniczny**

Projekt składa się z dwóch głównych komponentów: serwera i klienta, napisanych w języku C++ wykorzystującym gniazda sieciowe dla komunikacji. Aplikacja serwera zarządza logiką gry, w tym wyborem słowa, obsługą zgadywanych liter, oceną postępów graczy oraz koordynacją sesji gry. Aplikacja klienta umożliwia użytkownikowi połączenie się z serwerem, wprowadzenie swojego pseudonimu oraz uczestniczenie w grze.

### **Implementacja**

- Serwer: Zarządza sesjami gry, akceptuje połączenia od klientów, obsługuje logikę gry (wybór słowa, ocena zgadywanych liter), zarządza stanem gry i synchronizuje interakcje między graczami. Wykorzystuje gniazda TCP do komunikacji, a także wątki i mutexy do obsługi wielu klientów jednocześnie.
- Klient: Umożliwia graczowi połączenie się z serwerem, wprowadzenie pseudonimu i uczestnictwo w grze. Komunikuje się z serwerem za pomocą gniazd TCP/IP, wysyłając zgadywane litery i odbierając aktualny stan gry.

### **Metody Uruchamiania**

#### 1. Server:

- Skompiluj kod serwera używając kompilatora C++, np. ``g++ -o server server.cpp -pthread`` (flaga ``-pthread`` jest wymagana do obsługi wątków).
- Uruchom serwer, opcjonalnie określając adres IP i port, np. ``./server -ip 127.0.0.1 -port 8080``. Jeżeli nie zostaną podane, użyte zostaną wartości domyślne.

#### 2. Klient:

- Skompiluj kod klienta podobnie, używając ``g++ -o client client.cpp``.
- Uruchom klienta, opcjonalnie określając adres IP i port serwera, np. ``./client -ip 127.0.0.1 -port 8080``.

### **Wyniki i Wnioski**

Projekt został z powodzeniem zaimplementowany i przetestowany. Umożliwia wielu użytkownikom jednocześnie łączyć się z serwerem i współzawodniczenie w odgadywaniu słów. Implementacja wielowątkowości na serwerze pozwala na efektywną obsługę wielu połączeń jednocześnie, podczas gdy zastosowanie mutexów i zmiennych warunkowych zapewnia synchronizację stanu gry między graczami.

### **Możliwe Ulepszenia**

- Obsługa Błędów: Dodanie szczegółowej obsługi błędów, np. nieudanych prób połączenia lub problemów z siecią.
- Interfejs Użytkownika: Rozwinięcie klienta o graficzny interfejs użytkownika (GUI) dla lepszej interakcji z graczem.

Lindenau Dominika  
147567

- Rozszerzenie Funkcjonalności: Wprowadzenie dodatkowych funkcji, takich jak tablice wyników.

Podsumowując, projekt "Wisielec" udanie demonstruje, jak można zaimplementować prostą grę logiczną w modelu klient-serwer, wykorzystując wiedzę z zakresu programowania sieciowego i wielowątkowości w języku C++.