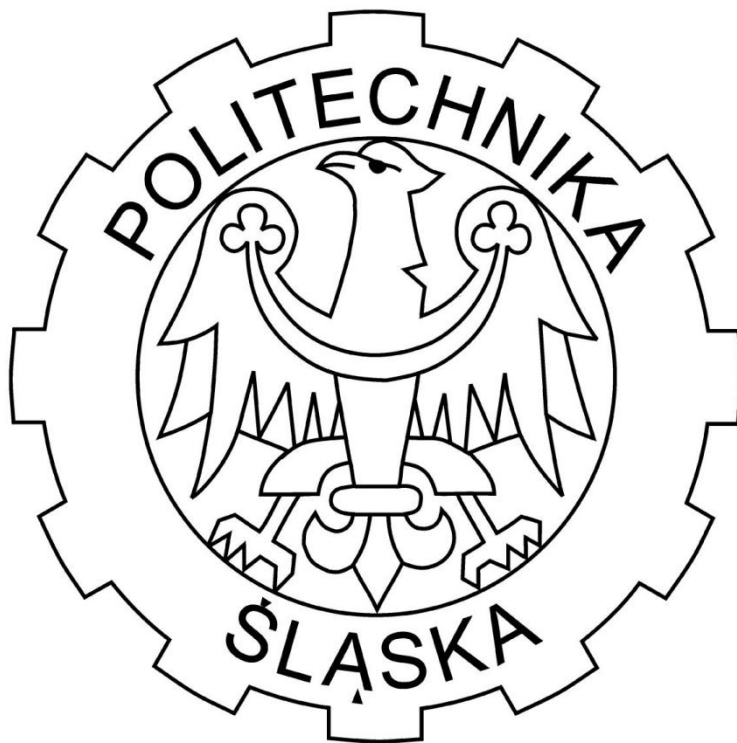


PROGRAMOWANIE W ŚRODOWISKU SIECIOWYM

DOKUMENTACJA PROJEKTU

Prosty komunikator tekstowy z interfejsem graficznym



Politechnika Śląska w Gliwicach
Wydział Elektryczny

Wykonanie:

Adam Celak, Bartłomiej Stokowy

Semestr: **VI**

Kierunek: **Informatyka**

Grupa: **E**

Data wykonania projektu: **Wrzesień 2020r.**

Spis treści

Wstęp do projektu.....	3
Cel projektu.....	3
Założenia projektu.....	3
Wykorzystane technologie.....	3
Opis projektu.....	4
Uruchomianie.....	4
Opis serwera.....	5
Opis klienta.....	6
Komunikacja.....	7
Oprogramowanie.....	9
Protokół komunikacyjny.....	9

Niniejszy dokument stanowi dokumentację projektu „Prosty komunikator tekstowy z interfejsem graficznym” stworzonego przez studentów Politechniki Śląskiej na potrzeby przedmiotu Programowanie W Środowisku Sieciowym.

1. WSTĘP DO PROJEKTU

a) Cel projektu

Celem projektu jest stworzenie prostego komunikatora tekstowego umożliwiającego wymianę informacji pomiędzy dwoma użytkownikami za pośrednictwem serwera

b) Założenia projektu

Projektowana aplikacja będzie składać się z dwóch części: klienta oraz serwera.

- Serwer nie będzie wyposażony w interfejs użytkowy będzie to jedynie program umożliwiający komunikację pomiędzy dwoma klientami. Wgląd w funkcjonowanie serwera będzie możliwy tylko za pomocą konsoli
- Kliencka część aplikacji zostanie wyposażona w prosty interfejs graficzny obsługiwany za pomocą klawiatury i myszy. Użytkownicy będą mieli możliwość nadania sobie nazwy i wysyłania wiadomości.

c) Wykorzystane technologie

Do stworzenia projektu wykorzystany zostanie język C++ wraz z bibliotekami Qt.

Środowisko w jakim zostanie napisany program to Visual Studio 2017 z dodatkami przeznaczonymi do projektowania z Qt.

Aplikacja będzie działała co najmniej na systemach Microsoft Windows 2007 lub nowszych.

2. OPIS URZĄDZEŃ

Niniejszy rozdział zawiera informacje dotyczące instalacji oraz użytkowania programu oraz wyjaśnienie głównych funkcji programu.

a) Uruchomienie

Instalacja programu polega na rozpakowaniu skompensowanego folderu .rar w dowolne miejsce na komputerze użytkownika. Następnie w celu umożliwienia wymiany informacji należy w pierwszej kolejności uruchomić serwer (Katalog: Server > Server.exe) a następnie można uruchomić klienta *lub dwóch* (Katalog: Client > Client.exe).

b) Opis serwera

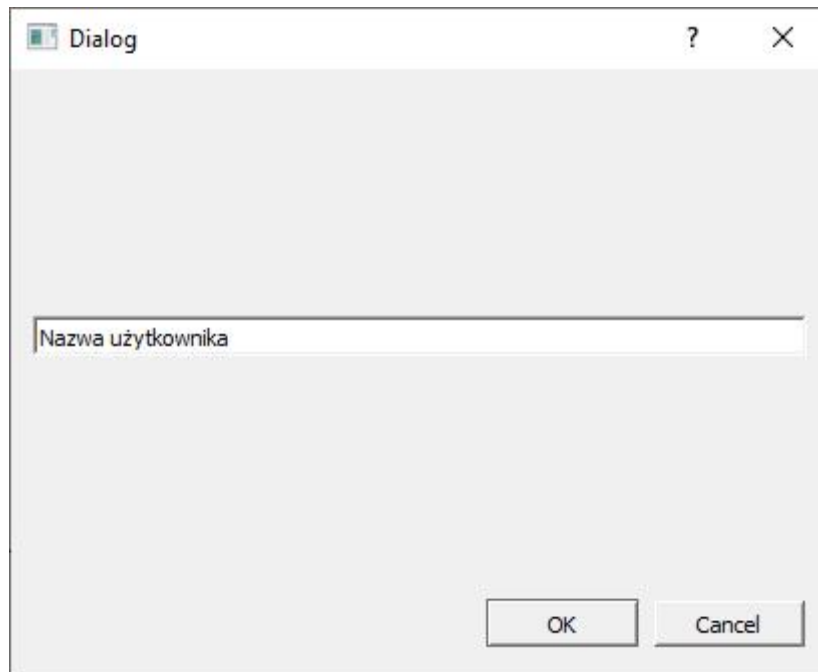
Po uruchomieniu aplikacji serwera otwiera się puste okno konsolowe (Rys.1), jest to oczekiwane działanie programu. Okna tego nie należy zamykać gdyż uniemożliwi to komunikację pomiędzy klientami (można je zminimalizować).



Rys. 1. Serwer

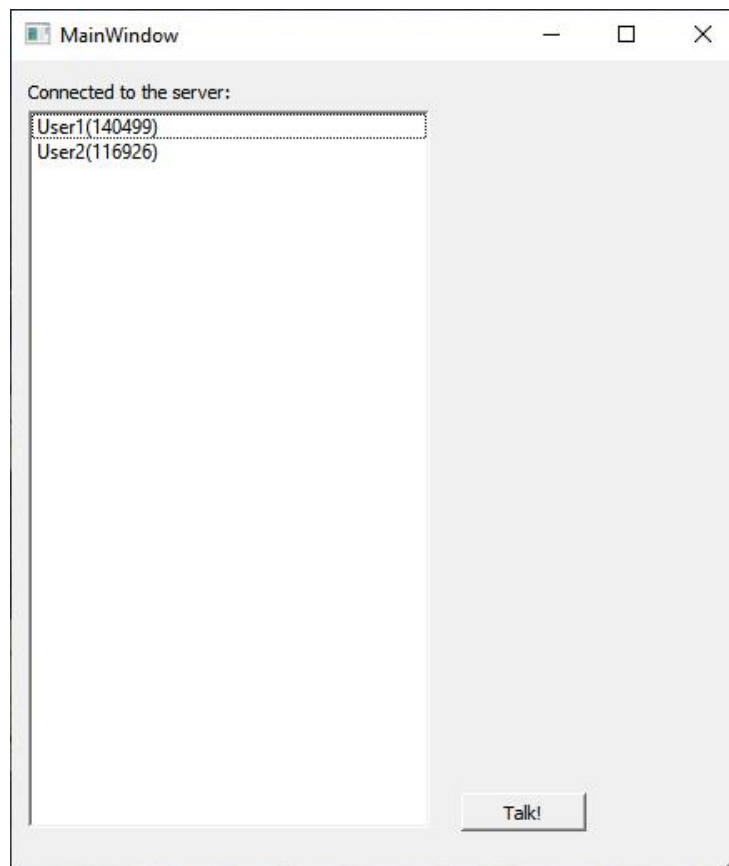
c) Opis klienta

Po uruchomieniu aplikacji klienta jako pierwsze pojawia się okno w którym należy wprowadzić nazwę użytkownika (Rys.2). Po wprowadzeniu nazwy należy nacisnąć przycisk „OK”. Nie wprowadzenie nazwy lub kliknięcie przycisku „Cancel” przerwie działanie aplikacji.



Rys. 2. Okienko wprowadzania nazwy

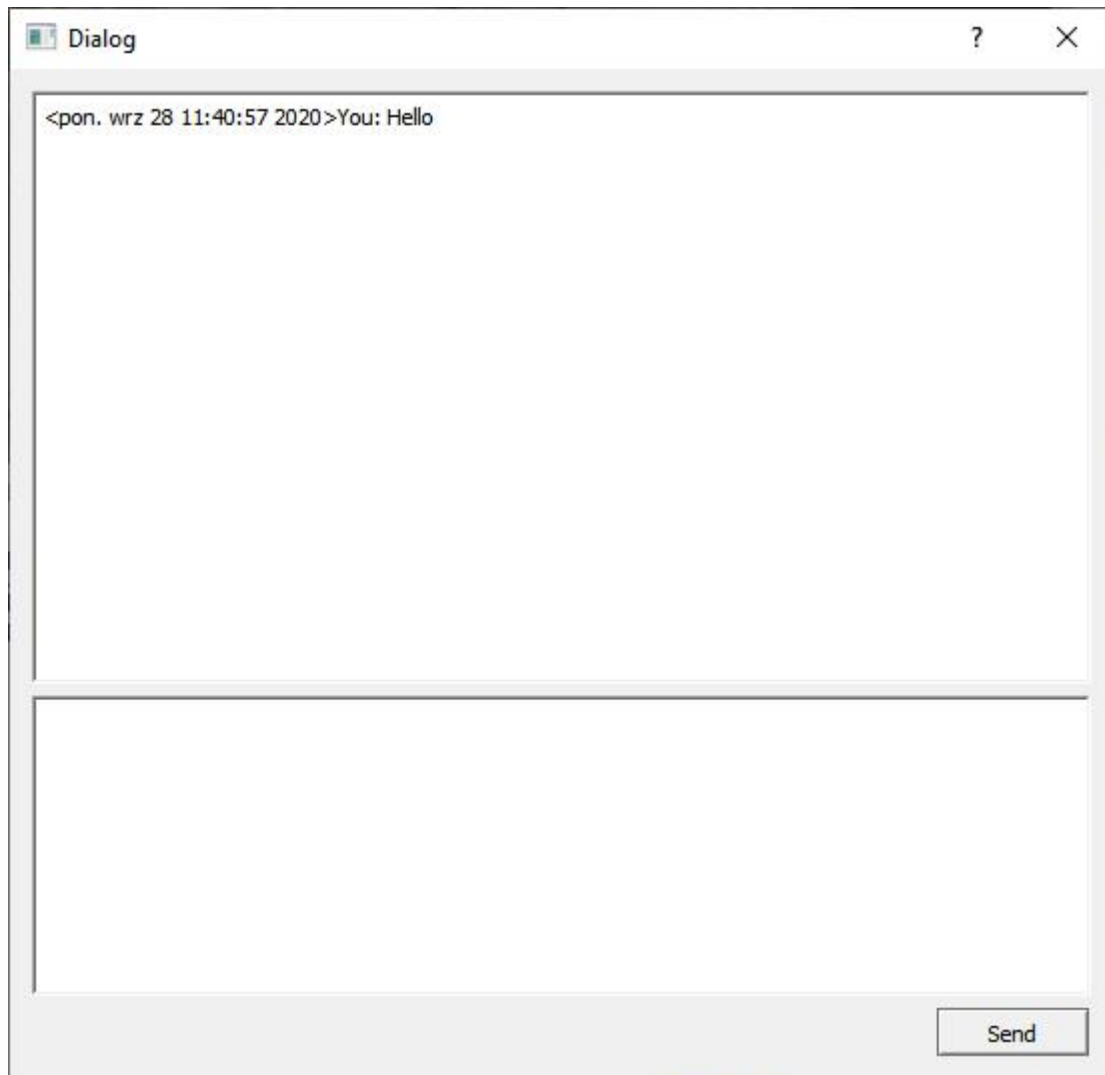
W kolejnym oknie wyświetlane są informacje o połączonych użytkownikach. Aby rozpocząć konwersację należy wybrać użytkownika z listy i nacisnąć przycisk „Talk!” (Rys. 3).



Rys. 3. Okno główne

Po otwarciu okna konwersacji widzimy dwa pola tekstowe (Rys.4). Górne, większe pole wyświetla historię czatu, natomiast dolne przeznaczone jest na wpisanie treści wiadomości. Aby wysłać wiadomość należy kliknąć na przycisk „Send” znajdujący się na dole okna aplikacji (*przycisk Enter wciśnięty na klawiaturze nie powoduje wysłania wiadomości*). Przy wszystkich wysłanych i odebranych wiadomościach wyświetlana jest data ich wystąpienia.

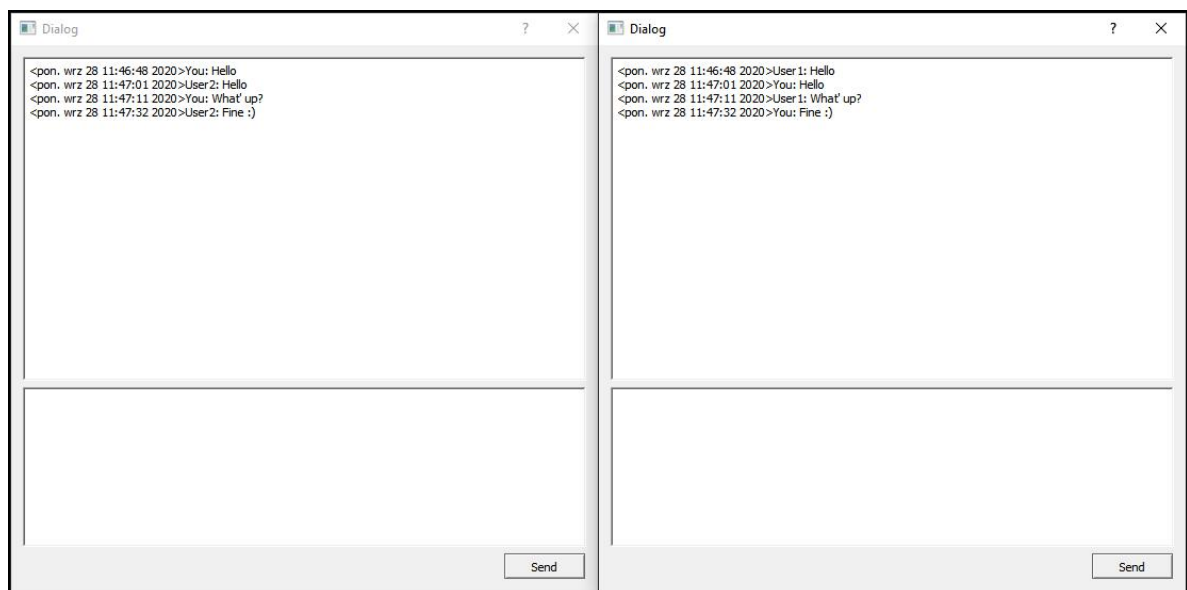
Okna aplikacji są w pełni skalowalne.



Rys. 3. Okienko czaru

d) Komunikacja

Komunikacja odbywa się między użytkownikami za pośrednictwem serwera. Poniżej przedstawiono przykładową komunikację pomiędzy użytkownikami (Rys. 4) oraz jej reprezentację na serwerze (Rys. 5).



Rys. 4. Komunikacja między 2 klientami



Rys. 5. Komunikacja (widok serwera)

3. OPROGRAMOWANIE

a) Opis protokołu komunikacyjnego

Projekt został oparty o protokół komunikacyjny TCP. Jest to protokół komunikacyjny stosowany do przesyłania danych między komputerami w sieciach. TCP działa w trybie klient-serwer. Serwer oczekuje na połączenie na konkretnym porcie. Klient natomiast inicjuje połączenie do serwera. TCP gwarantuje dostarczenie wszystkich pakietów w całości (bez duplikatów), wraz z zachowaniem kolejności. TCP operuje w warstwie transportowej modelu OSI.

b) Opis komunikacji

Komunikacja pomiędzy użytkownikami odbywa się poprzez wydzielone w aplikacji kanały. Każdy użytkownik po podłączeniu do serwera otrzymuje swoje unikalne ID, które stanowi podstawę do jego identyfikacji. Informacje wymieniane pomiędzy klientami oraz serwerem przesyłane są za pomocą zdefiniowanych ramek. Każda z ramek obciążona jest swoim unikalnym kodem definiującym jej przeznaczenie (Rys. 6).

```
const int NAME_SEND = 150;  
const int MESSAGE_SEND = 151;  
const int ID_SEND = 152;  
const int CONNECTED_HOSTS = 153;  
const int REMOVE_HOST = 254;
```

Rys. 5. Kody ramek

NAME_SEND - ramka zawierająca utworzoną nazwę klienta, (przesyłana do serwera w celu nadania mu ID).

MESSAGE_SEND - ramka zawierająca wiadomość przekazywaną pomiędzy użytkownikami przesyłana zarówno od klienta do serwera jak i następnie od serwera do klienta

ID_SEND - wysłanie do wszystkich aktualnie podłączonych klientów informacji z nazwą i ID nowo podłączonego klienta (w celu dopisania go do listy aktualnie podłączonych klientów i umożliwienia komunikacji)

CONNECTED_HOSTS- ramka przesyłana do nowego użytkownika zawierająca informacje o wszystkich obecnie podłączonych klientach

REMOVE_HOST - ramka zawierająca informację o rozłączeniu się użytkownika, rozsyłana do podłączonych klientów (w celu usunięcia go z listy)

4. WNIOSKI

Podczas pracy nad projektem bardzo ważne jest ustalenie harmonogramu pracy już na samym początku, podczas tworzenia założeń które ma spełnić dany program. Pomoże to uniknąć pośpiechu i natłoku pracy przed wyznaczonym deadline.

Przydatną funkcją jest wyświetlanie po stronie serwera wszelkich zdarzeń i przesyłanych komunikatów, może pomóc to w identyfikacji błędów oraz umożliwi przejrzystą kontrolę nad aplikacją.

Po raz kolejny przekonano się o ogromnym wpływie małych błędów typu wstawienie operatora przypisania „=” zamiast porównania „==”. Był to jeden z głównych problemów z funkcjonowaniem aplikacji zaraz obok niewłaściwego wykorzystania wskaźników.