Kacper Woźniak 151118

Projekt Reversi SK2

1. Temat

Tematem projektu, było zaimplementowanie serwera obsługującego logikę do gry Reversi w systemie linux i języku C++, oraz klienta z GUI do gry, który komunikuje się z serwerem za pomocą protokołu TCP. Serwer powinien móc obsługiwać pokoje gier.

2. Opis protokołu komunikacyjnego

W tym projekcie użyty został prosty protokół komunikacyjny oparty na przesyłaniu wiadomości tekstowych między klientem, a serwerem przy użyciu gniazd sieciowych i wątków. Każda wiadomość jest odpowiednio kodowana i dekodowana, a komunikacja odbywa się poprzez funkcje do odczytu i zapisu danych do gniazda. Gracze komunikują się w celu synchronizacji ruchów i utrzymania spójnego stanu gry pomiędzy aplikacją kliencką a serwerem. Przykładowe wysyłane dane:

- **Stan planszy**

  Plansza do gry Reversi jest przesyłana w formie tekstowej. Każde pole planszy jest reprezentowane przez jeden znak, który informuje o kolorze pionka na danym polu. Na przykład, 'e' oznacza puste pole, 'w' oznacza biały pionek, a 'b' oznacza czarny pionek. Cała plansza o wymiarach 8x8 zajmuje 64 znaki.

- **Ruchy graczy**

  Ruchy graczy są przesyłane jako liczby całkowite, gdzie dwucyfrowa liczba reprezentuje współrzędne ruchu na planszy. Na przykład, liczba 23 może oznaczać ruch na polu (2, 3) na planszy.

- **Informacje o turze**

  Informacja o tym, czyja jest aktualnie tura, jest przesyłana w formie jednego znaku. Dla przykładu, 'w' może oznaczać, że teraz tura należy do gracza grającego białymi pionkami.

- **Informacje o zakończeniu gry**

  Po zakończeniu gry przesyłana jest informacja o wyniku gry. Możliwe wartości to 'b' dla zwycięstwa gracza z czarnymi pionkami, 'w' dla zwycięstwa gracza z białymi pionkami, 'd' dla remisu.

3. Opis implementacji, w tym krótki opis zawartości plików źródłowych

Projekt ten obejmuje implementację gry Reversi w języku C++ przy użyciu biblioteki SFML do obsługi grafiki i interakcji z użytkownikiem. Klient został zaimplementowany przy pomocy wielowątkowości. Jest on rozdzielony na 2 wątki, jeden wątek obsługuje GUI i główną pętlę gry, natomiast drugi wątek obsługuje komunikację z serwerem. Dzięki takiemu rozwiązaniu można pozbyć się problemów z zawieszaniem się okna. Natomiast implementacja serwera w tym projekcie obejmuje zarządzanie połączeniami z klientami, logikę gry Reversi, obsługę ruchów graczy oraz pokoi gier, a także przesyłanie stanu planszy i informacji o turze.

4. Sposób kompilacji, uruchomienia i obsługi programów projektu

Sposób kompilacji klienta jest dosyć skomplikowany przez wykorzystywaną bibliotekę graficzną SFML. W pierwszej kolejności trzeba pobrać odpowiednią wersję SFML, która będzie pasowała do wersji c++. Sprawdzona i działająca wersja to: MINGW C++ 12.2.0 i działa ona z SFML wersji 2.5.1, przy kompilowaniu w CLI. Alternatywnym sposobem kompilacji jest uruchomienie programu w Visual Studio z odpowiednio skonfigurowanym SFML (jest to łatwiejszy sposób). Jeżeli jednak zdecydujemy się korzystać z CLI to będziemy musieli wykonać kilka kroków:

1. W folderze z projektem otworzyć terminal i wpisać polecenie:
   g++ -c klient.cpp -I”ścieżka do folderu include SFML”

2. Następnie wpisujemy polecenie:
   g++ klient.o -o main -L”ścieżka do folderu lib SFML” -lsfml-graphics -lsfml-window -lsfml-system -lws2_32 -Wall

3. Kolejnym krokiem jest przekopiowanie plików z folderu bin SFML do folderu z projektem.

4. Ostatnim krokiem jest uruchomienie aplikacji poprzez wpisanie komendy (port jest domyślnie ustawiony na wartość 1234):
   .\main.exe “adres ip”

Jeżeli kompilacja nie działa, lub są z nią problemy można skorzystać z poradnika (przed skorzystaniem z poradnika trzeba się upewnić, która wersja c++ współpracuje, z którą wersją SFML):
https://terminalroot.com/how-to-compile-your-games-in-sfml-with-gcc-mingw-on-windows/

Sposób kompilacji serwera:

1. Wpisanie do terminala polecenia:
   g++ server.cpp -o main -Wall

2. Wpisanie do terminala polecenia:
   ./main