

Sprawozdanie z projektu

1. Temat Projektu

Tematem projektu była gra szachy z wykorzystaniem protokołu komunikacyjnego TCP/IP w architekturze klient-serwer.

2. Opis protokołu komunikacyjnego TCP/IP

Teoretyczny model warstwowej struktury protokołów komunikacyjnych. Podstawowym założeniem modelu TCP/IP jest podział całego zagadnienia komunikacji sieciowej na szereg współpracujących ze sobą warstw (ang. *layers*). Każda z nich może być tworzona przez programistów zupełnie niezależnie, jeżeli narzucimy pewne protokoły według których wymieniają się one informacjami. Założenia modelu TCP/IP są pod względem organizacji warstw zbliżone do modelu OSI. Jednak liczba warstw jest mniejsza i bardziej odzwierciedla prawdziwą strukturę Internetu.

3. Opis implementacji

Klient za pomocą funkcji `SendMessageToServer()` wysyłana jest wiadomość zawierająca znak akcji, którą ma wykonać serwer. Funkcja `ReceiveMessageFromServer()` odbiera odpowiedź od serwera zezwalającą na dalsze akcje. Serwer jest oparty na wątku który czyta czy klient nie wykonał jakiejś akcji po czym przetwarza ją w zależności od otrzymanego znaku akcji. Po wykonaniu danej akcji odsyła klientowi znak akcji który określa wynik jej wykonania.

- Opis plików źródłowych
 - `Game.cpp` – obsługuje menu, wysyła informacje do serwera czy pokój został stworzony lub czy gracz chce dołączyć do już istniejącego
 - `Chess.cpp` – obsługuje grę właściwą, tworzy figury, rozstawia je, obsługuje ruchy gracza
 - `NetworkConnection.cpp` – obsługuje połączenie pomiędzy klientem a serwerem, wysyła i otrzymuje wiadomości
 - `Board.cpp` – tworzy i rysuje plansze do gry, przechowuje tablicę rozmieszczenia figur na planszy
 - `Pawn.cpp`, `Knight.cpp`, `Rook.cpp`, `Bishop.cpp`, `Queen.cpp`, `King.cpp` – klasy obsługujące figury, przypisują tekstury do sprite'ów, ustawiają dany sprite
 - `detail.h` - przechowuje kolor i typ jaki ma być nadany figurze
 - `ChessPiece.cpp` – określa poszczególne obiekty figur (np. `Pawn`, `Knight`) jako `figure`

4. Sposób kompilacji, uruchomienia i obsługi programów projektu

- Kompilacja klienta

Aby skompilować projekt należy otworzyć plik `ChessOnline.sln` i skompilować go w trybie `Debug` bądź `Release`. Konieczne jest ustawienie konfiguracji `x86`.
- Kompilacja serwera

Aby skompilować serwer należy przejść do folderu z plikiem `server.c` i użyć komendy
„`gcc -Wall -pthread server.c -o Server`”

- Uruchamianie serwera
Aby uruchomić serwer należy użyć komendy ./server.
- Uruchamianie klienta
UWAGA!!! Przed uruchomieniem klienta należy włączyć serwer.
Aby uruchomić program należy przejść do folderu Debug/Release (w zależności w jakim trybie skompilowaliśmy projekt) i uruchomić plik ChessOnline.exe

Do obsługi programu należy używać myszki. Klikając na napis „Create Room” tworzymy pokój (grę) po czym czekamy na drugiego gracza. Klikając na napis „Join Room” dołączamy do już istniejącego pokoju. Po dołączeniu do pokoju (gry) rozpoczyna się gra. Zaczyna gracz, który stworzył pokój (grę). Figury „podnosimy” figury. Aby je „położyć” należy kliknąć na pole planszy. Po położeniu następuje zmiana tury (gracz, który dołączył wykonuje ruch).