

1. Opis projektu:

Projekt zakładał stworzenie sieciowej gry w Warcaby opartej o komunikację klient-serwer. Gra posiada możliwość zalogowania się oraz przeprowadzenia partii w Warcaby, korzysta z zaawansowanego GUI i ręcznie tworzonej grafiki. Serwer oparty o architekturę NIO umożliwia obsługę więcej niż 2 graczy jednocześnie.

Projekt został zrealizowany w języku Java, w środowisku Eclipse. Program jest spakowany w plik o rozszerzeniu .jar, a także gotowe są pliki wykonywalne o rozszerzeniu .exe.

2. Uruchomienie:

Z płyty cd należy skopiować cały folder „exe” w którym znajdują się dwa pliki

DraughtsClient.exe – Klient gry

DraughtsServer.exe – Serwer gry

oraz folder „img” w którym znajdują się wymagane pliki graficzne

By uruchomić program, należy najpierw uruchomić serwer podając ip i port na którym będzie on nasłuchiwał połączeń od klientów. Następnie należy uruchomić klientów i połączyć się z serwerem podając unikalny login.

3. Obsługa:

Po połączeniu z serwerem pokazuje się nam okienko z listą dostępnych gier, którą należy odświeżyć.

By stworzyć nową grę, należy wpisać nazwę gry w polu tekstowym i kliknąć przycisk „New Game”, by dołączyć do gry, należy wpisać w polu tekstowym jej nazwę, lub wybrać z listy dostępnych gier i kliknąć przycisk „Join Game”.

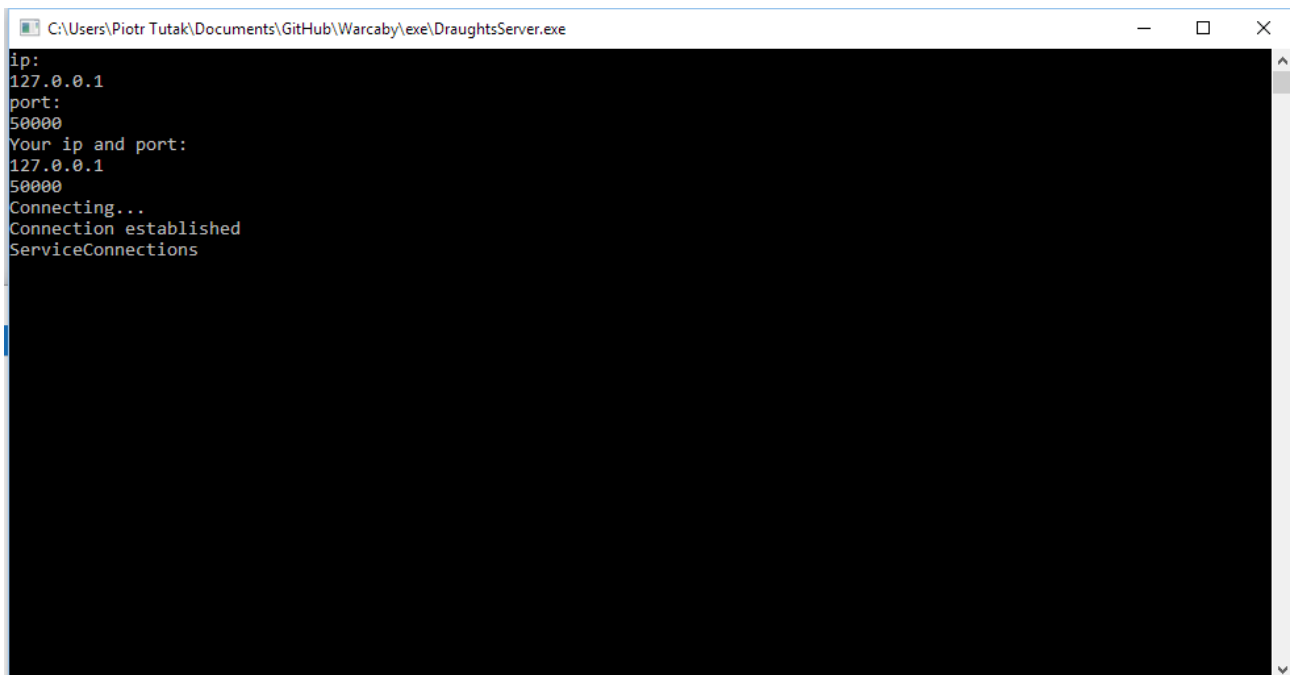
Po uruchomieniu głównego okna należy albo poczekać na drugiego gracza (jeśli stworzyliśmy nową grę) albo kliknąć przycisk Ready/Start Game. Gdy obaj gracze są gotowi, gra się uruchamia.

4. Rozgrywka:

Zaczyna czerwony gracz jako pierwszy. Gra toczy się do momentu wyeliminowania wszystkich pionków przeciwnika z planszy.

5. Screeny:

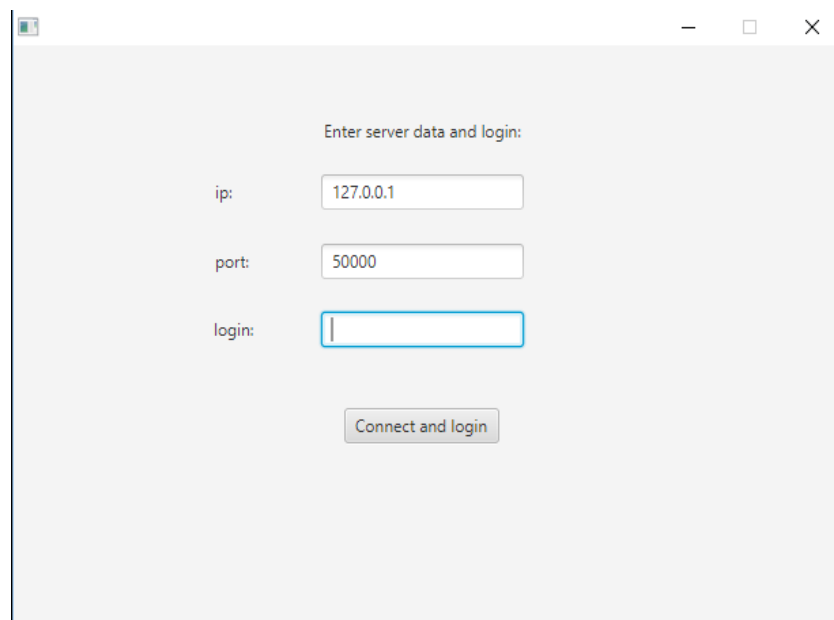
Okienko serwera, można podać adres ip i port:



```
C:\Users\Piotr Tutak\Documents\GitHub\Warcaby\exe\DrughtsServer.exe
ip:
127.0.0.1
port:
50000
Your ip and port:
127.0.0.1
50000
Connecting...
Connection established
ServiceConnections
```

The screenshot shows a Windows command prompt window titled "C:\Users\Piotr Tutak\Documents\GitHub\Warcaby\exe\DrughtsServer.exe". The text inside the terminal displays the configuration of the server, including the IP address (127.0.0.1) and port (50000), followed by a confirmation message "Connection established" and "ServiceConnections".

Okienko logowania:



The screenshot shows a login window titled "Enter server data and login:". It contains three input fields: "ip:" with the value "127.0.0.1", "port:" with the value "50000", and "login:" which is currently empty. Below the input fields is a button labeled "Connect and login".

Lista gier, wybór gry:

Game Name	Row Number	Player Red	Turn Time Limit	Game Time Limit
No content in table				

Join or create a game.

Enter Game Name:

New Game

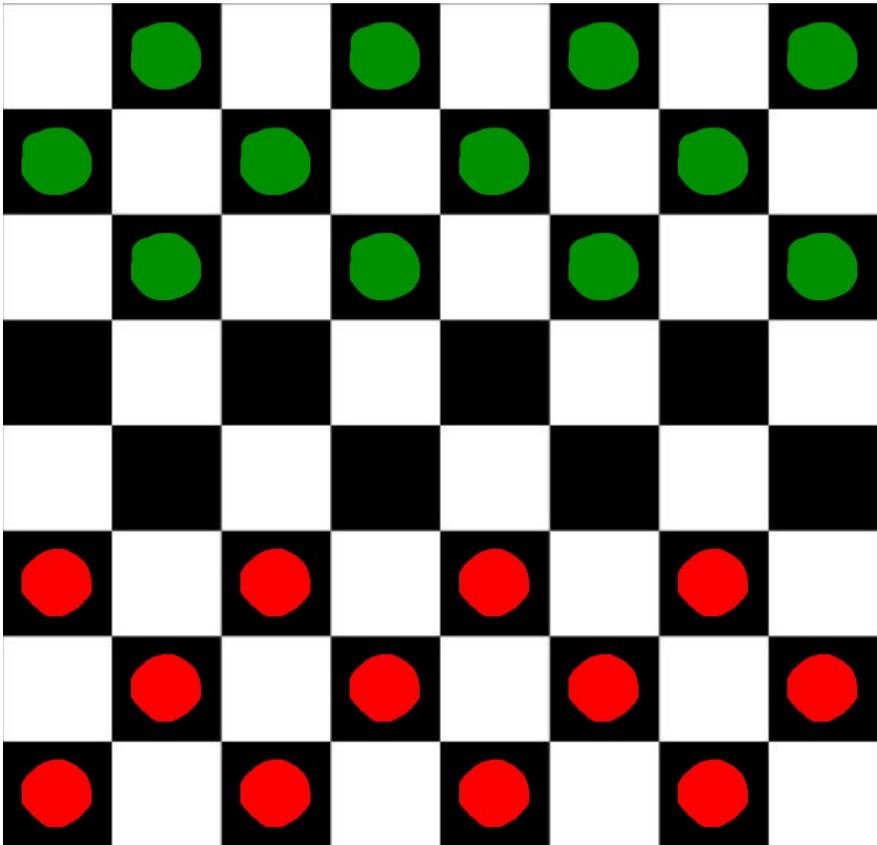
Join Game

Refresh


Quit

[illegible]

Rozgrywka:




Opposite Player:

piotr

Player NOT READY

GAME STARTED!!!

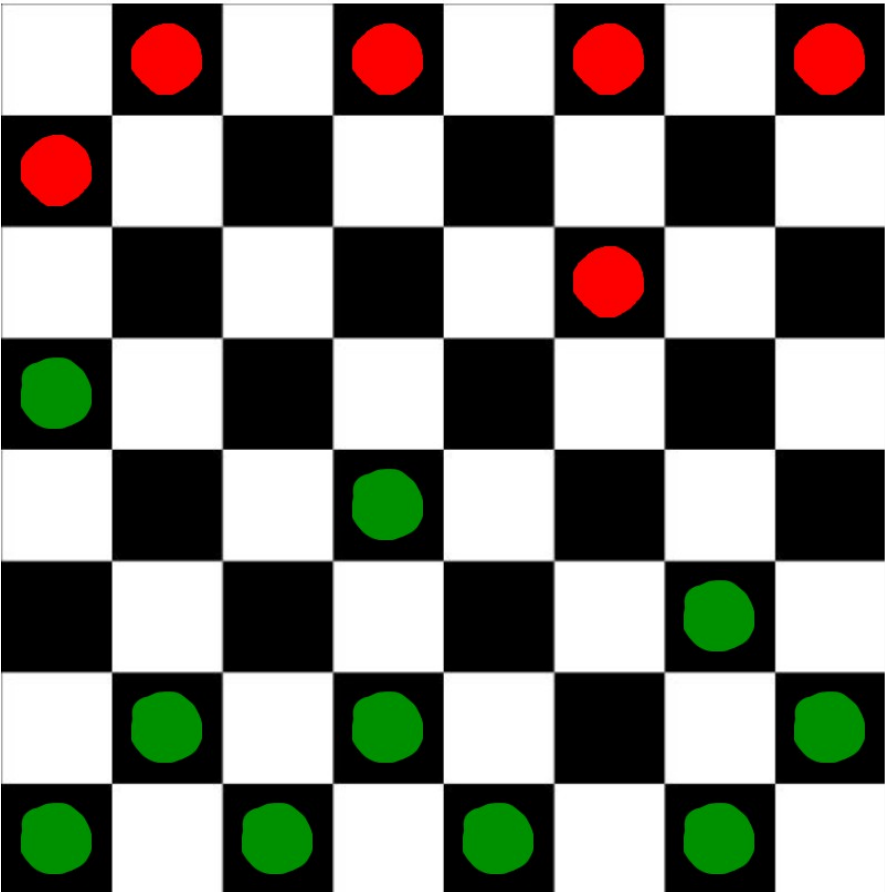
You:

adam


READY

Player Ready / Start Game

Game ListExit




Opposite Player:

adam

READY

OPPONENTS MOVE

You:

piotr

READY

Player Ready / Start Game

Game ListExit

6. Podsumowanie i wnioski:

Najwięcej problemów i czasu zajęło napisanie całego serwera NIO, szczególnie poprawne napisanie bezpiecznej funkcji odczytu pakietu, który pojawia się na kanale i obsługa wszystkich sytuacji wyjątkowych mogących wystąpić w różnych sytuacjach komunikacji sieciowej, jak na przykład nagle rozłączenie się klienta. Również dość sporo zajęło zaprogramowanie samej rozgrywki.