Projekt gry **Gomoku** na „Sieci Komputerowe 2”

1. **Autor**

Patryk Lukaszewski 148058 I6.1

1. **Środowisko w którym program był testowany**
   1. server.c
      * + - Ubuntu 22.04
   2. client.py
      * + - Windows 11 z Pythonem 3.10
          - Ubuntu 22.04 z Pythonem 3.7
2. **O projekcie**

W projekcie zaimplementowana została gra turowa Gomoku. Serwer obsługuje równoległe rozgrywki w trzech pokojach o numerach 0, 1 i 2. Serwer waliduje poprawność ruchu, powiadamia o zakończeniu rozgrywki oraz obsługuje rozłączenie gracza.

1. **Zasady gry w Gomoku**

Dwóch zawodników rywalizuje na planszy 15x15. Gracze kładą naprzemiennie kamienie na planszy. Zwycięzcą zostaje gracz, któremu uda się ułożyć **dokładnie** 5 kamieni swojego koloru w nieprzerwanej linii (poziomej, pionowej lub ukośnej).

1. **Uruchomienie gry**
   * + **Uruchomienie serwera**
2. Na systemie linuxowym będąc w katalogu, gdzie znajduje się plik „server.c” należy wykonać polecenie:

*gcc server.c -lpthread -Wall -o server.out*

1. Następnie należy uruchomić serwer za pomocą polecenia

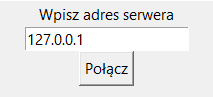
*./server.out*

1. Jeżeli na terminalu pojawił się napis „Serwer oczekuje na połączenie...” oznacza to, że serwer jest uruchomiony i oczekuje na klientów

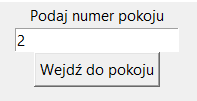
* **Uruchomienie klienta**
  1. Będąc w katalogu z plikiem „client.py” należy wykonać polecenie w terminalu

*python client.py lub python3 client.py*

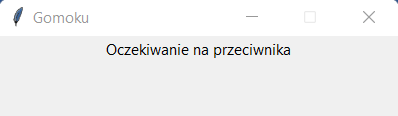
* 1. Następnie w polu „Wpisz adres serwera” należy wpisać adres ip serwera (np. „127.0.0.1”, „192.168.1.12”) i kliknąć przycisk połącz



* 1. Gdy uda się połączyć z serwerem pokaże się pole do wpisania numeru pokoju, możliwe pokoje to wyboru to 0, 1, 2. Po wpisaniu numeru pokoju należy kliknąć „Wejdź do pokoju”



1. Jeżeli udało się wejść do pokoju, ale w pokoju nie ma jeszcze rywala to pojawi się okno oczekiwania. Gdy znajdzie się przeciwnik okno automatycznie zmieni się na planszę do gry.



1. Gdy pojawi się plansza to rozgrywka się rozpoczęła.
2. **Problemy podczas tworzenia projektu**
   1. Problem z przesyłaniem int z klienta napisanego w Pythonie do serwera napisanego w C

* Rozwiązanie :
* Przed wysłaniem trzeba zamienić int na bajty za pomocą:

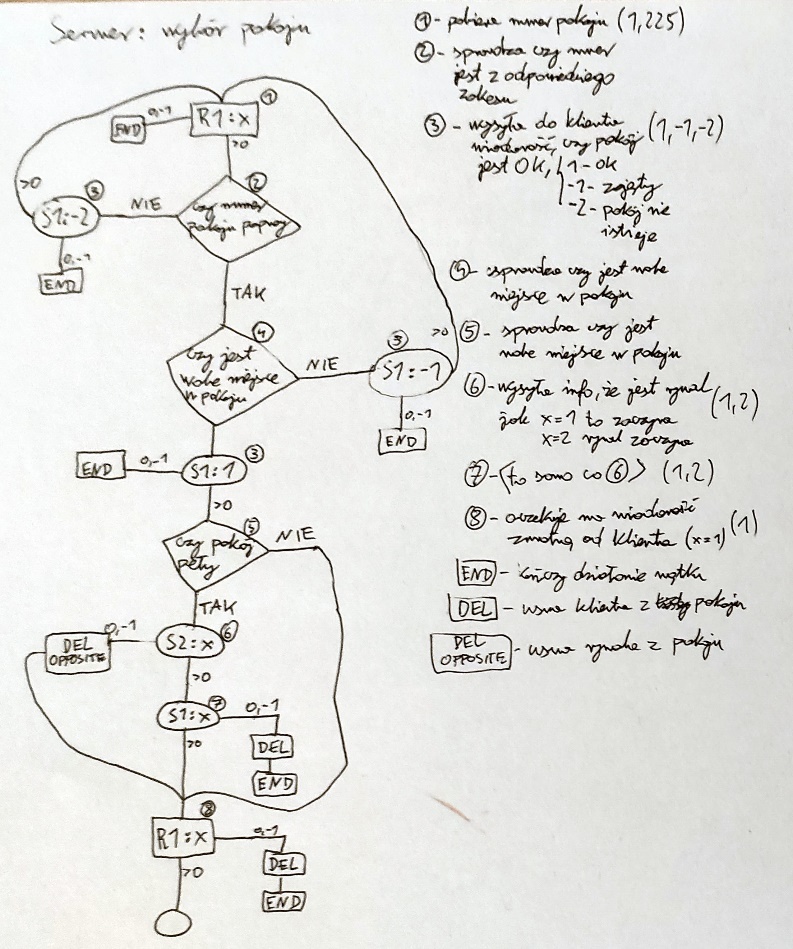
*Val\_byte = val\_int.to\_bytes(4, 'little', signed=True)*

* Po odebraniu trzeba zamienić bajty na int za pomocą kodu:

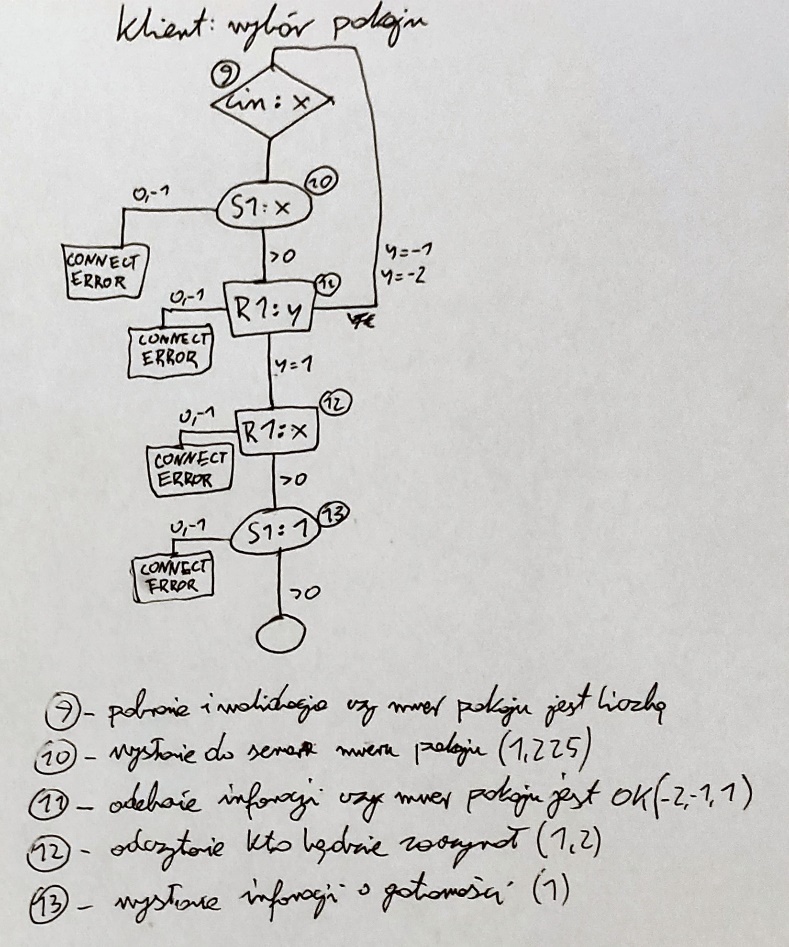
Val\_int = int.from\_bytes(val\_byte, "little", signed=True)

* 1. Problem z zawieszającym GUI klienta podczas oczekiwania na drugiego klienta
* Rozwiązanie:
* Użycie wątków w kliencie, jeden wątek zarządza GUI a inny odpowiada za komunikację

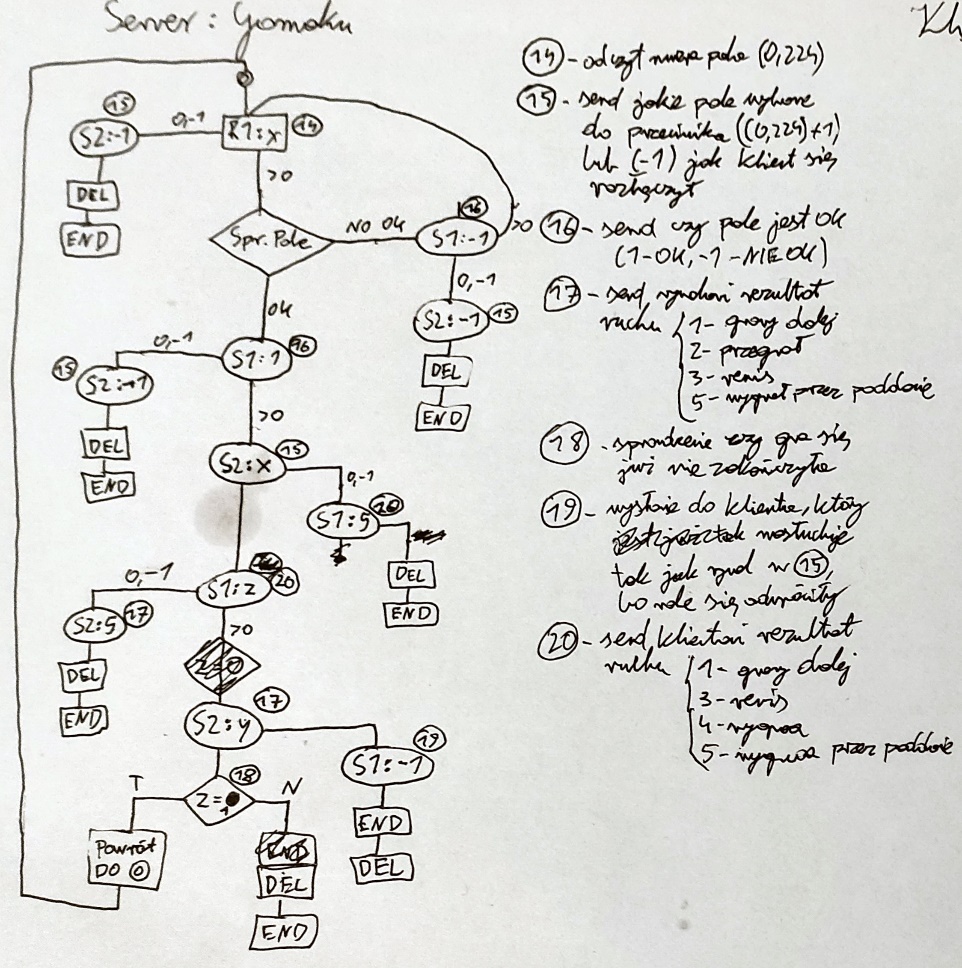
1. **Schematy blokowe funkcji**
   1. **Wybieranie pokoju**
      * + - Serwer



* Klient



* 1. **Rozgrywka**
* Serwer



* Klient

