**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

# **«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

# **ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу**

**Кафедра математичних методів системного аналізу**

**Дисципліна: “Комп’ютерні мережі”**

**Проект**

|  |  |
| --- | --- |
|  | Виконала: студентка  3 курсу групи КА-77  Деменкова В.В. |
|  | Прийняв: к.т.н. Кухарєв С.О. |

**Київ – 2020**

**Мета роботи:**

Реалізувати програмний продукт , що демонструє роботу клієнт-серверної багато поточної архітектури на основі гри у хрестики-нулики

**Правила гри:**

Кожен з учасників підключається до сервера по вказаному порту та починає гру з комп’ютером. При підключені клієнта до сервера обидва отримують повідомлення про підключення. Після підключення, сервер відправляє клієнту запит, де клієнт має вказати координати точки місця, на якому має стояти хрестик. Наступний крок за комп’ютером. Якщо клієнт випадково обере місця, що зайняте, сервер повідомить про це. Є три варіанта кінця гри : виграш клієнта, виграш комп’ютера або нічия.

**Ігрова логіка:**

Гра В якості полів використовуємо 3 символьні константи: SIGN\_X, SIGN\_O и SIGN\_EMPTY. Для ігрового поля використано двовимірний символьний масив table. Ігрова логіка базується на циклі while, з якого можна вийти при втраті зв’язку. Кінець ігри (виграш або програш) наступає коли всі клітинки заповнені, і за допомогою break ми виходимо з циклу та завершуємо гру.

* Метод initTable() забезпечує початкову ініціалізацію ігрової таблиці, заповнивши пустими клітинками.
* Метод showTable() відповідає за відображення поточного стану ігрової таблиці.
* Метод sendTable відповідає за відправлення ігрової таблиці клієнту.
* Метод turnHuman() дозволяє користувачу зробити свій хід.
* Метод turnAI() дозволяє комп’ютеру зробити свій хід.
* Метод checkWin() перевіряє таблицю на наявність «переможної трійки», тобто якщо три символи однакового знаку підряд стоять на вертикалі, горизонталі чи на одні з діагоналей.
* Метод isTableFull() перевірка на заповнення ігрової таблиці.

**Хід роботи:**

1. Створення класу сервера, що має в собі містити номер порту, до якого мають підключитися клієнти, список клієнтів, в який він записує нові підключення та ServerSocket, в єдиному екземплярі. Там наявний метод accept() для того, щоб визначити що з сервером хоче з’єднатися клієнт.

2. Для багато поточності необхідно створити нащадок класу Thread – ServerThreads і перевизначити метод run(). В конструкторі повинен бути проініціалізован сокет, через який можна буде звернутися до конкретного клієнта.

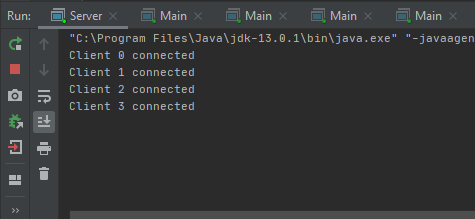
3. Створення моделі гри за ігровою логікою та занесення її в клас ServerThreads.

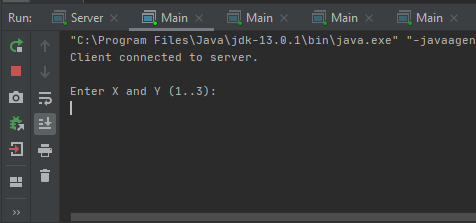
4. Створення класу Main для клієнтів. Створення сокету зв’язку на стороні клієнта з сервером і при виході з гри закриття сокету та потоків введення/виведення.

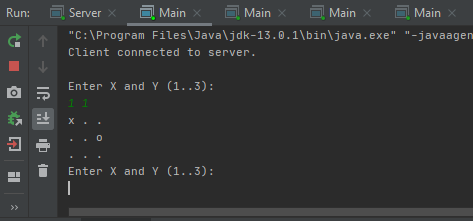
Програмний продукт реалізовано за допомогою мови програмування Java. Використання мережевого інструменту Java такого як сокет дозволяє працювати через протокол TCP/IP, так як являється достатньо низкорівневим інтерфейсом.

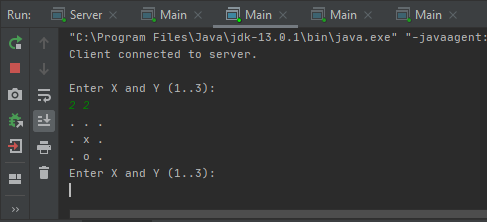
|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| 1 | . | . | . |
| 2 | . | . | . |
| 3 | . | . | . |

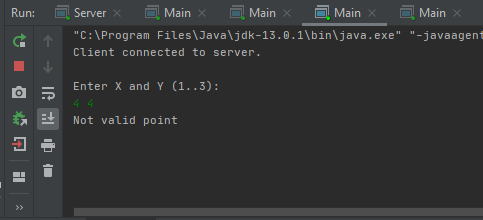
**Тестування гри:**

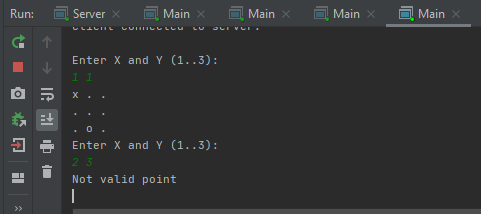
****

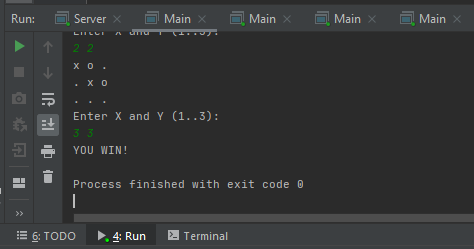


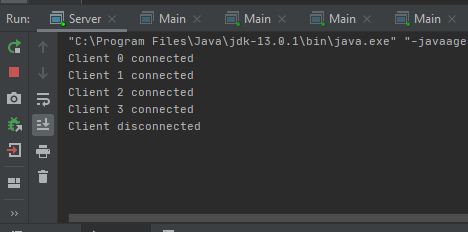












https://github.com/Demenkova/TicTacToe/