

Курсовий проект

Дисципліна “Розробка клієнт-серверних застосунків”

Виконали: Алексєєнко Анастасія та Братюк Катерина (КН-2)

1. Розподіл ролей:

Репозиторій: <https://github.com/katernabratiuk/The-Game>

Анастасія: передача даних по мережі, ігрова логіка, налаштування середовища (база даних на віддаленому сервері, тестування на емуляторі локальної мережі), графічний інтерфейс, unit-тести.

Катерина: доступ та робота з базою даних, графічний інтерфейс, unit-тести.

2. Опис:

Ідея: іо-гра на кшталт Slither.io. Користувачі, зареєстровані, під'єднуються до сервера. Перед ними стоїть вибір в трьох категоріях: персонаж, зброя, паверап (powerup). Задача користувача вбити якомога більше противників і не померти. Справа зверху користувачі можуть побачити рейтинг. Рейтинг гравців зберігається весь час життя сервера.

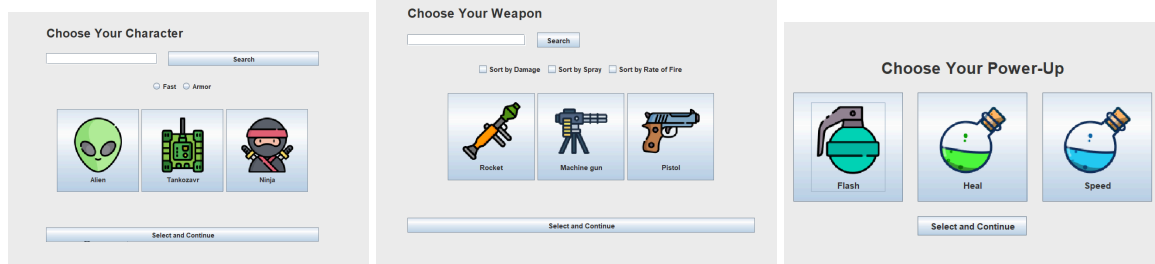
Графічний інтерфейс реалізований на бібліотеці Swing.

1) Вхід/Реєстрація

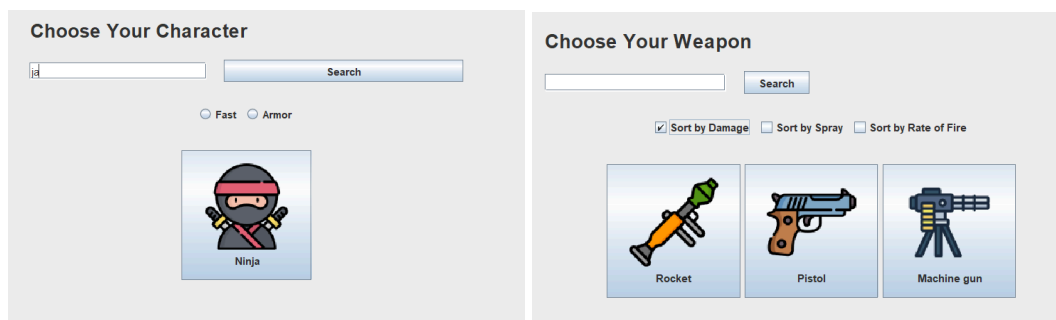
Користувач не може рєструватись під чужим юзернеймом, не може вводити пусті/короткі поля.

The image displays three sequential screenshots of a Java Swing application's login and registration interface. The first screenshot, titled "The Game", shows a simple layout with a "Login" button and a "Register" button, with a link "Not registered yet?" between them. The second screenshot, titled "Login to The Game", features input fields for "Username:" and "Password:", followed by "Login" and "Register" buttons, and a "Not registered yet?" link. The third screenshot, titled "Register", includes input fields for "Username:", "Password:", and "Confirm Password:", along with "Register" and "Back" buttons.

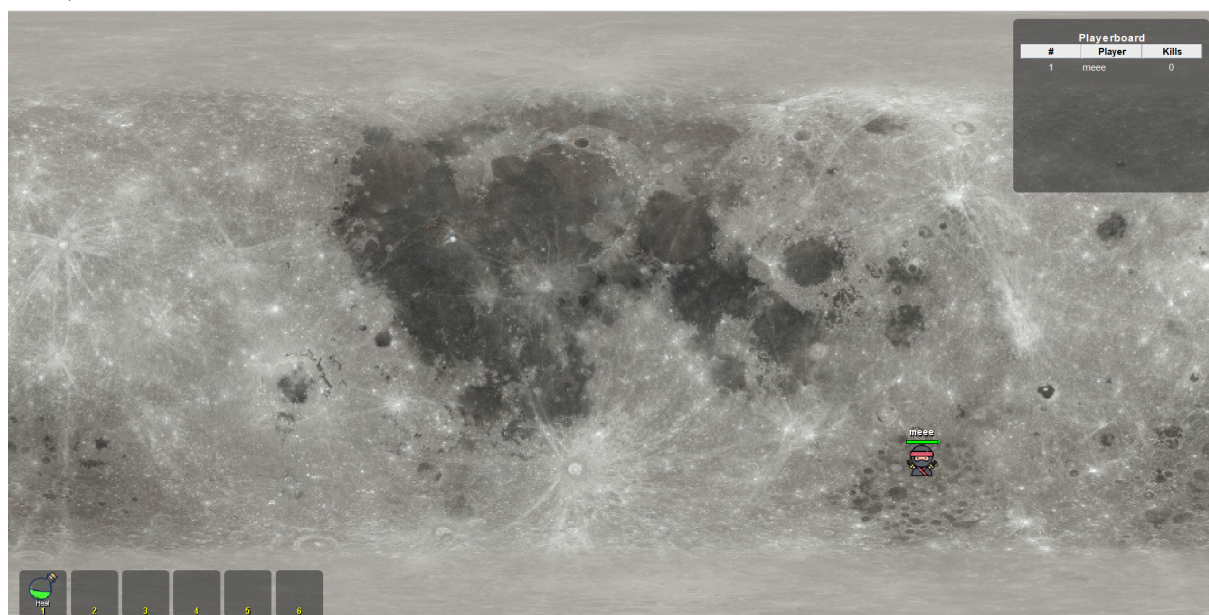
- 2) Користувач має обрати персонажа, зброю і power-up. Він також може здійснити пошук по імені/посортувати їх за певними ознаками.



Пошук:



- 3) Початок гри

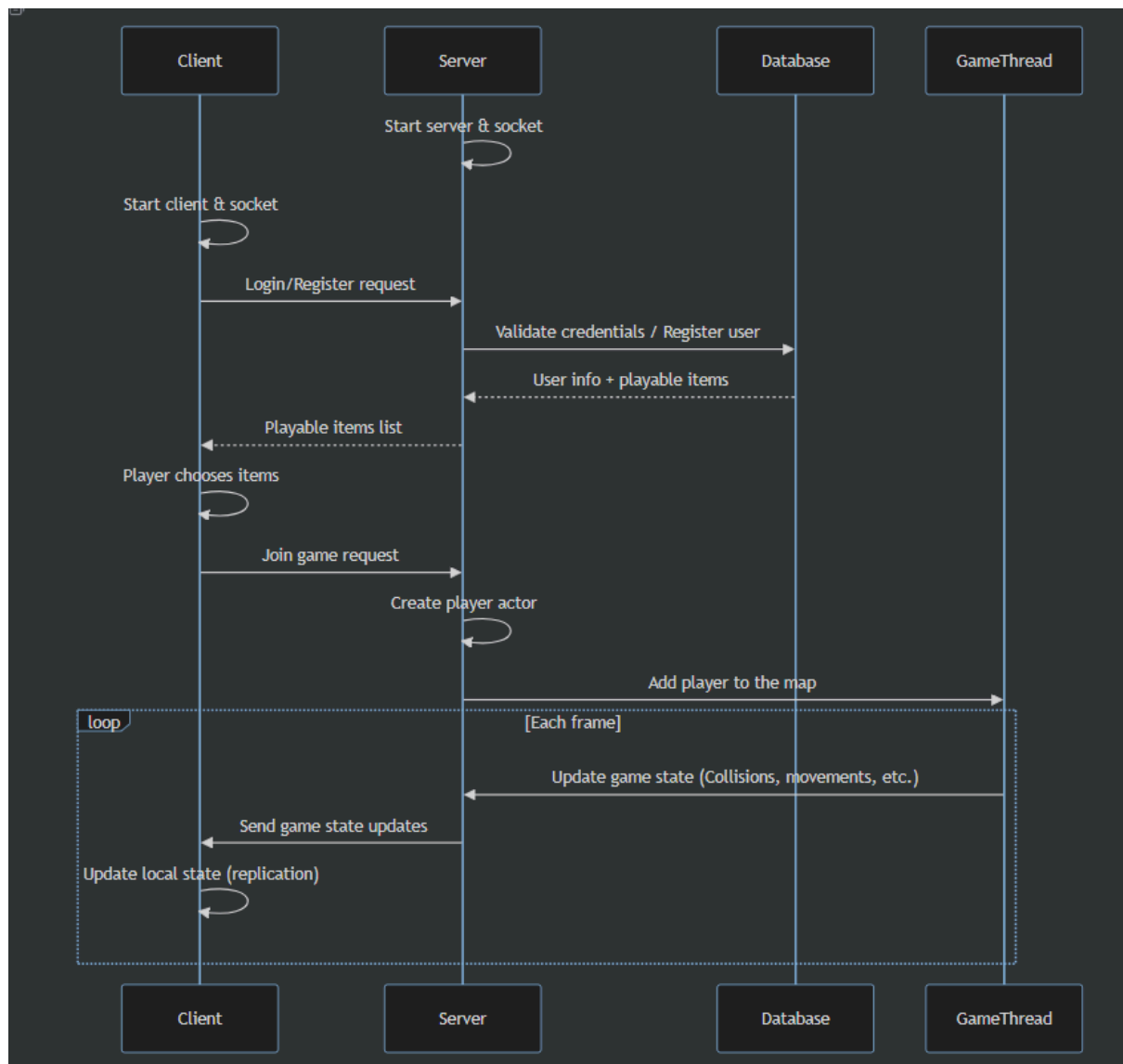


May the odds be ever in your favor!

3. Діаграми програмної логіки:

- *Загальна логіка:*

Клієнти та сервер обмінюються даними за допомогою UDP. Відображення даних на кожному з клієнтів відбувається за принципом реплікації - клієнт тільки надсилає події на сервер та рендерить оновлення, які прийшли від сервера. Вся логіка гри відбувається на сервері - тому усі користувачі отримують однакові постійні дані. База даних розгорнута на віддаленому сервері (AWS EC2 Ubuntu micro instance).

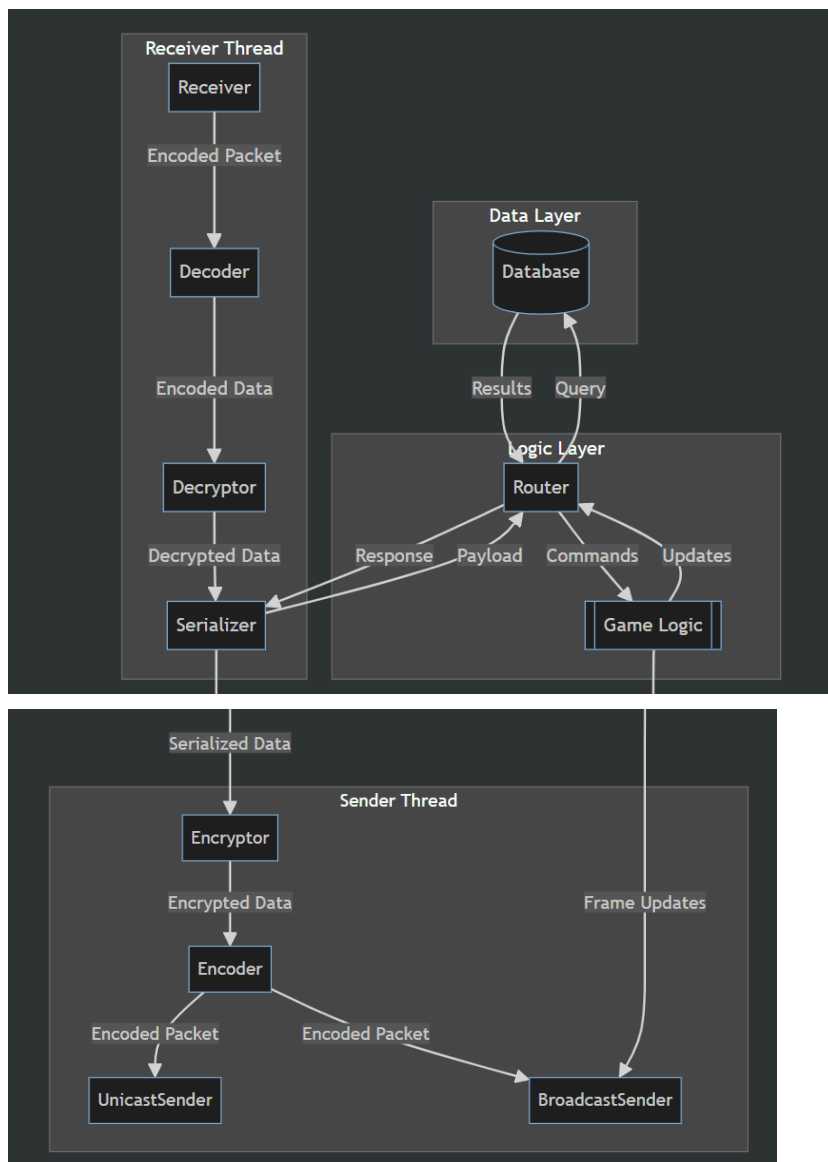


- Сервер:

Ми знайшли схожості та реалізували інтерфейс взаємодії клієнта та сервера по UDP - сокетам, який потім перевикористали і на клієнті, і на сервері. Таким чином, у сервера є Receiver Thread(слухає, отримує, декодує, розшифровує, десеріалізує) та Sender Thread(серіалізує, шифрує, кодує, відправляє). Sender може приймати різні стратегії та бути як Broadcast, так і Unicast.

Направленням отриманих даних на обробку займається Router, який викликає потрібну логіку для отриманого, і, якщо потрібно, відправляє відповідь.

Ще одна важлива частина - Game Thread. Це потік, який кожен кадр оновлює стан гри та відправляє Broadcast усім активним клієнтам. Оновлення включає в себе розрахунок колізій, моніторинг результатів гри, вбитих персонажів, прибирання куль тощо.



- Клієнт:

Клієнт має такий самий механізм взаємодії по UDP-сокету, тільки він може бути тільки Unicast, бо отримувач у нього завжди один. Клієнт тільки отримує дані з сервера, відмальовує їх у компонентах. Якщо у компонентах відбувається якась подія (наприклад, натискання на кнопку), клієнт відправляє відповідний зашифрований пакет серверу, який його обробить.

