

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ННК «ІІСА» НТУУ «КП ІМ. ІГОРЯ СІКОРСЬКОГО»
КАФЕДРА ММСА

Проект

З дисципліни: «Комп'ютерні мережі»

Бомбермен

Виконала: Студентка ІІІ курсу

Групи КА-77 Яцько Я. В.

Перевірів: Кухарєв С. О.

Київ 2020

Постановка задачі: реалізувати клієнт-серверну архітектуру гри «Бомberman» на транспортному рівні моделі OSI.

До гри може підключитися клієнт, що має спеціальний ключ для входу в потрібну кімнату (в нашому випадку назву хосту та потрібний порт). Одночасно сервер можна розгорнути на вільних портах – це і є окремі ігрові кімнати. До гри можна підключатися в будь-який момент гри, до її завершення (Гра завершується, коли помирає останній гравець – тоді вивільняється даний порт).

Хід виконання роботи:

1. Створити модель, де прописані логіки:
 - Розстановка персонажів
 - Пересування персонажів
 - Поповнення чи зниження hp (Health Points) персонажів
 - Знищення персонажів
 - Розстановки блоків
 - Розстановки бонусів
 - Розстановки бомб
 - Підривання бомб
2. Створити представлення, що:
 - Відображає ігрове вікно
 - Відображає фони та стіни
 - Відображає спрайти
 - Відображає вибух бомби
 - У разі смерті персонажу відображає надпис про закінчення для нього гри
3. Створити контролер введення даних клієнтами з клавіатури та їх опрацювання
4. Створити контролер сітки, що реалізує взаємодію клієнта та сервера
 - Контролер для сервера
 - ◆ Ініціалізація потоку
 - ◆ Відправка карти та розстановки всіх елементів на ній клієнтам
 - ◆ Отримання положення кожного з клієнтів та відправка
 - ◆ Отримання положення бомб, якщо вони були поставлені та відправка
 - ◆ Підключення клієнтів
 - ◆ Відключення клієнтів

- ◆ Передбачити затримку повідомлень
- Контролер для клієнта
 - ◆ Ініціалізація клієнта
 - ◆ Смерть персонажа (клієнта)
 - ◆ Отримання карти та всіх елементів на ній
 - ◆ Відправка місця положення поставленої бомби
 - ◆ Відправка свого розташування
 - ◆ Непередбачений вихід клієнта, натисненням на клавішу
 - ◆ Пересування клієнта, в залежності від натиснення кнопки
 - ◆ Розставлення бомби, в залежності від натиснення кнопки
 - ◆ Отримання даних про розстановку всіх інших об'єктів та персонажів на карті
 - ◆ Передбачити затримку повідомлень
- 5. Реалізувати сервер (для того, щоб створити «кімнату» для гри, потрібно ввести порт, що буде ключем для клієнтів)
- 6. Реалізувати клієнта (для того, щоб підключитися до гри, клієнт має ввести потрібний хост та порт(ключ) та своє ім'я)

Ігрове поле будується на основі файлу, що обраний як карта за визначенням. Файл вміщує текстові символи, кожен з яких відповідає за елемент на позиції масиву.

Приклад:

```

wwwwwwwwwwwwwwww
w11111111111w
w1w1w1w1w1w1w
w11111111111w
w1w1w1w1w1w1w
w11111111111w
w1w1w1w1w1w1w
w11111111111w
w1w1w1w1w1w1w
w11111111111w
wwwwwwwwwwwwwwww
  
```

, де w – стіна, 1 – поле

Спрайти це картинки в форматі png, розміром 30*30, що мають 4 зображення:



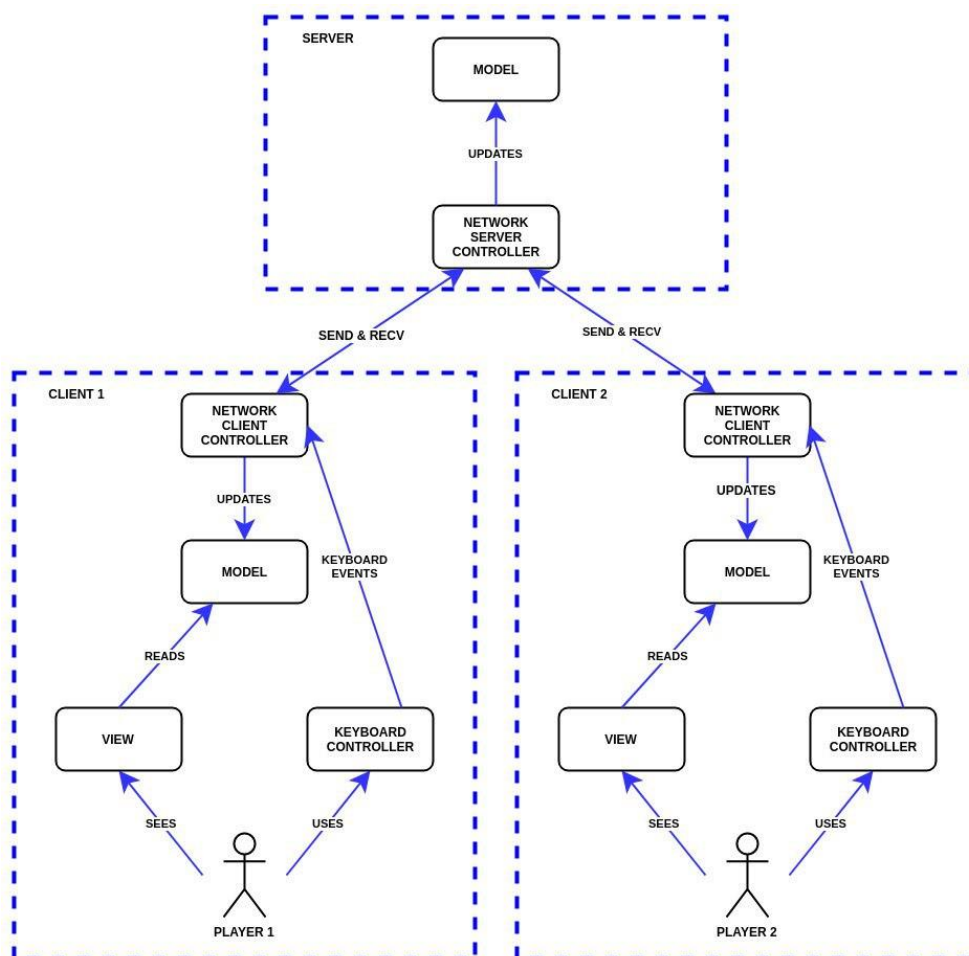
Опис готового рішення:

Рішення реалізоване мовою Python3.7 з використанням бібліотеки pygame, що допомогла реалізувати графічний інтерфейс. Інші допоміжні бібліотеки можна переглянути за посиланням.

Дане архітектурне рішення виконано за допомогою патерна MVC (Model-View-Controller), що дозволяє відділити графічний інтерфейс від логіки та логіку від даних.

- Модель (Model) забезпечує доступ до даних та містить всю логіку (пересування персонажів, розстановка бомб і тд.)
- Представлення (View) відповідає за відображення даних користувачу, реагуючи на зміну моделі.
- Контролер (Controller), в нашому випадку Network_Controller контролює дії клієнта та відповіді сервера і передає до моделі зміни в стані.

Нижче представлена схема MVC даного проекту:



Посилання на репозиторій: <https://github.com/yayatskaa/bomberman>

Тестування гри:

