

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО
АНАЛІЗУ

Проект

із дисципліни «Комп'ютерні мережі»

Виконав: студент 3-го курсу

гр. КА-71

Аблець А. В.

Прийняв: Кухарєв С.О.

Київ 2020

Загальні відомості

Мета: розробити кінцеву гру-мультиплеєр «Танки».

Мова програмування: Python 3.7.

Підключені бібліотеки: pygame, socket, time, pickle, _thread.

Середовище розробки: PyCharm Community

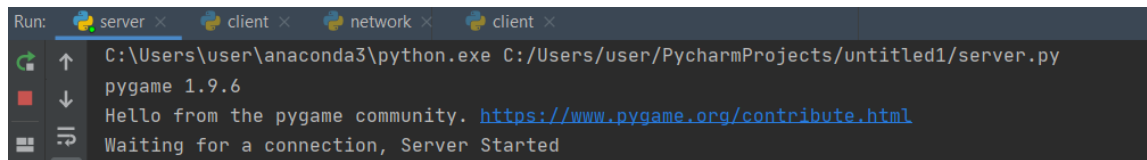
Опис гри

Гра являє собою типового представника аркадних ігор. В кожного гравця є танк (зелений або червоний), яким він керує, натискаючи на стрілочки. Також кожен гравець може стріляти снарядами за допомогою клавіши «Пробіл». В кожного гравця є по 3 життя. Перший гравець, що попав у супротивника – перемагає, відповідно інший – програє. В кінці гри впливає результат бою, згідно якому можна дізнатися який гравець переміг, а який програв, після цього гра завершується.

Етапи розробки програми

Етап I. Створення серверу.

- Спочатку було створено файл “server.py”, який представляв собою роботу сервера. Було встановлено значення вільного порту (вибрано значення 5555) та встановлено значення IPv4-адреси серверу (192.168.0.105). Далі було сполучено сервер з портом та надано дозвіл серверу отримувати зв’язки. При запуску серверу програма виводила наступну інформацію:

A screenshot of a PyCharm Run console window. The window has four tabs: 'server', 'client', 'network', and 'client'. The 'server' tab is active. The console output shows the following text: 'pygame 1.9.6', 'Hello from the pygame community. <https://www.pygame.org/contribute.html>', and 'Waiting for a connection, Server Started'. The background of the console is dark with light-colored text.

Тобто можна зрозуміти, що сервер очікує підключення клієнтів.

- Далі в сервері було створено два екземпляри класу `Player()`, про який буде розповідатися нижче. Також було створено функцію, що у циклі пакує та надсилає або приймає й розпаковує інформацію про об'єкти класу. Ця функція викликала у циклі для кожного нового клієнта.

Етап II. Створення мережі.

- Було створено файл “network.py”, що являв собою клас “Network”. У цьому класі було визначено вільний порт (5555) та значення IPv4-адреси серверу (192.168.0.105), сокет (за використанням сімейства адрес IPv4 та протоколу TCP).
- Було визначено 2 функції. Перша - “connect” зв'язувалась з сервером по заданими адресою і портом та вертала запаковане значення, яке відправив сервер. Друга - “send” відправляла серверу запаковані дані та теж вертала запаковане значення, яке відправив сервер.

Етап III. Створення файлу про гравця.

- Спочатку було ініційовано змінні, що стосуються танка (координати розташування, швидкість переміщення, поворот танку, кількість життів) та снаряду (координати розташування, швидкість переміщення, поворот снаряду, готовність снаряду, пройдена відстань)

- Оголошено допоміжні функції, такі як: `get_place()` – повертає розташування танку, `minus_hp()` – віднімає одне життя в танка, `end_game()` – ініціює кінцевий екран в клієнтів, з текстом про перемогу чи програш, `get_tank_image_file()` – повертає картинку танка.
- Оголошено функції розміщення танків та снарядів: `draw()` – промальовує танк на екрані, `rotate_and_draw()` – спочатку повертає (якщо треба) зображення танку, а потім промальовує його на екрані (ці функції знадобляться при відмалюванні танку в іншого клієнта). `draw_bullet()` – повертає зображення снаряду в залежності від положення танку в момент пострілу та промальовує його на екрані.
- Створено функцію про постріл снаряду – `fire_bullet()`. Спочатку функція в залежності від розміщення танку, що стріляв, та танку, в який стріляли, перевіряє чи снаряд попав у свою ціль. Якщо снаряд попав, вертається “hit”, що у файлі клієнта допоможе відняти одне життя у гравця в якого влучили, також змінюється стан снаряду на “ready”. Потім згідно с напрямом, в якому стріляли, снаряд відмалюється згідно своєї швидкості.
- Створена функція повороту – `rotate()`. Функція, згідно зі старим положенням танку та положенням, у який гравець хоче перемістити свій танк, визначає кут повороту, який знадобиться для визначення положення танку в другого клієнта та власне повертає танк та відмалює це. На відміну від функції `rotate_and_draw()`, функція працює з основним клієнтом.
- Визначена функція руху – `move()`. Функція зчитує всі клавіші, які натиснув гравець, та дивлячись на те, що саме за клавіша була натиснута, виконує якісь функції по відношенню до танку. На початку функція перевіряє, чи було натиснуто одночасно декілька клавіш, якщо так, то функція виконує `return`. Далі, якщо був натиснутий “Пробіл”, танк робить постріл (якщо снаряд у стані “ready”). Якщо було натиснуто клавіші по переміщенню

танка (стрілочки), то функція спочатку перевіряє чи є у заданому напрямі ворожий танк (щоб танки не проходили скрізь один одного), а потім рухається (якщо на шляху немає ворожого танку).

Етап IV. Створення файлу з клієнтом.

- Спочатку було створено вікно, на якому відображалася гра, імпортовано зображення танків, снаряду та заднього фону, імпортовано музичне супроводження гри та шрифт з розміром (для закінчення гри). Також була створена функція, що відновлювала на екрані розміщення танків – `redrawWindow()`.
- Була створена головна функція – `main()`, що викликала вкінці. В функції було створено екземпляр класу `Network()` (для отримання інформації про танк іншого клієнта), створена змінна `clock` з бібліотеки `pygame` (для створення максимального розміру FPS). Далі у циклі надсилалася та отримувалася інформація щодо стану об'єкту `Player`, робилася перевірка на закриття вікна клієнтом, перевірявся стан життя танків гравців (якщо в когось з гравців життя дорівнювало нулю, гра закінчувалась для обох клієнтів), перевірявся стан снаряду (якщо стан був "fire", то викликала функція `fire_bullet()`). Також на кожній ітерації циклу викликала функція `redrawWindow()` для промалювання танків на екрані.

Робота програми



Рис. 1. Робота двох клієнтів на одному пристрої

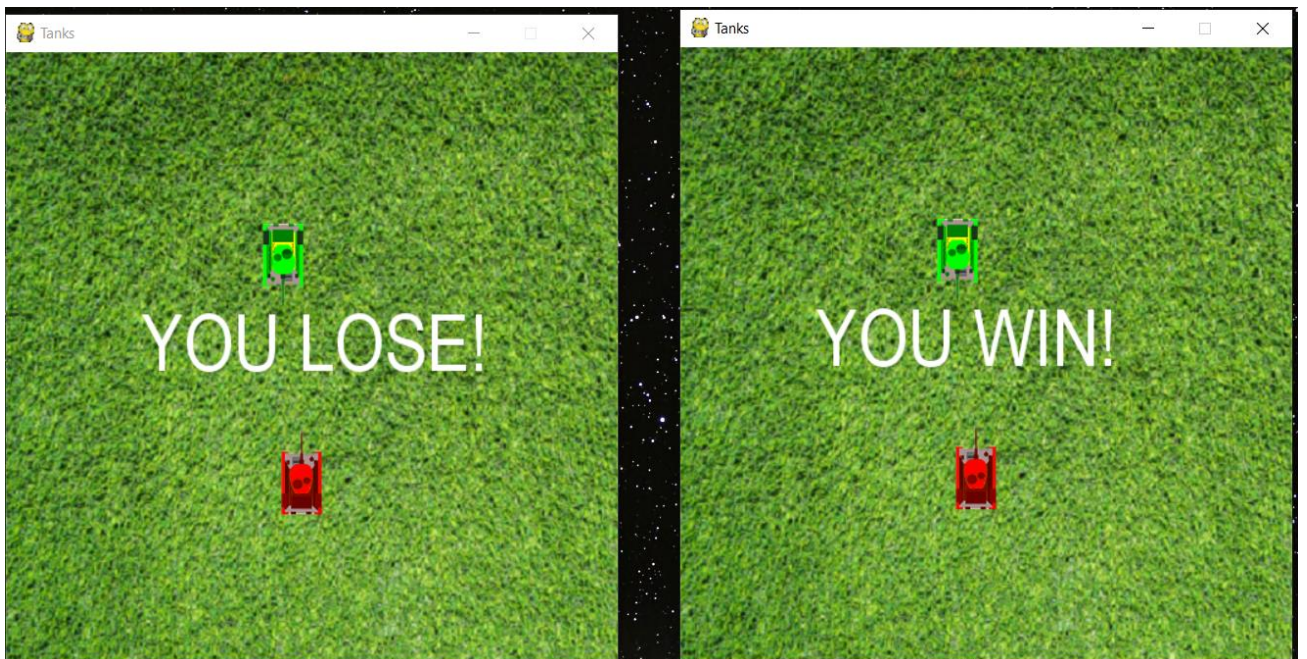


Рис. 1. Кінець гри, правий клієнт переміг, лівий програв

Висновок

Розробляючи цю гру на Python, я розібрався з бібліотекою pygame для розробки ігор та з бібліотеками _thread, socket для роботи з сервером. Взагалі, мені сподобалось розробляти цю гру, це був корисний досвід. В подальшому, я планую доробити її шляхом додавання перешкод, через які танки не зможуть проїхати, але зможуть пробити собі шлях снарядами.