

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

Возняк В. З.

Прийняв: Кухарєв С.О.

Київ 2020

Загальні відомості

Мета: розробити кінцеву гру-мультиплеєр.

Мова програмування: Python 3.8.

Підключені бібліотеки: pygame, socket, pickle, _thread.

Середовище розробки: PyCharm Community Edition 2019.3.2.

Опис гри

Дана гра базується на принципі “Дилеми в’язня”. У теорії ігор дилема в’язня (ДВ) — гра з ненульовою сумою, в якій гравці прагнуть одержати вигоду, співпрацюючи один з одним або зраджуючи. Як у всій теорії ігор, передбачається, що гравець («в’язень») максимізує свій власний виграш, не піклуючись про вигоду інших. Наведемо правила нашої гри. Вам необхідно зробити вибір. Уявіть перед собою певний апарат. Якщо ви покладете монету в апарат – інший гравець отримає три монети й навпаки. Перед вами обома стоїть вибір: співпраця (покласти монету), чи обман (не класти монету). Якщо ви обираєте співпрацю та інший гравець обирає обман – ви втрачаєте монету, а він отримує три (рахунок: -1 проти +3), проте якщо ви обоє обираєте обман – ніхто не втрачає й не отримує нічого (рахунок: 0 проти 0). Якщо ж ви обоє обираєте співпрацю, то ви віддаєте монету, щоб отримати три натомість (рахунок: +2 проти +2), проте якщо ви вирішите обманути, а інший гравець вирішить співпрацювати – ви отримаєте три монети за вартістю однієї монети іншого гравця (рахунок: +3 проти -1). Саме в цьому полягає ”Дилема в’язня”. Довіра - це круто, проте вона дозволяє іншим ошукувати вас. Часом недовіра є раціональним вибором! Яким же буде результат, якщо ми зіграємо в цю гру... Відповідь на це питання Ви зможете отримати, зігравши в неї!

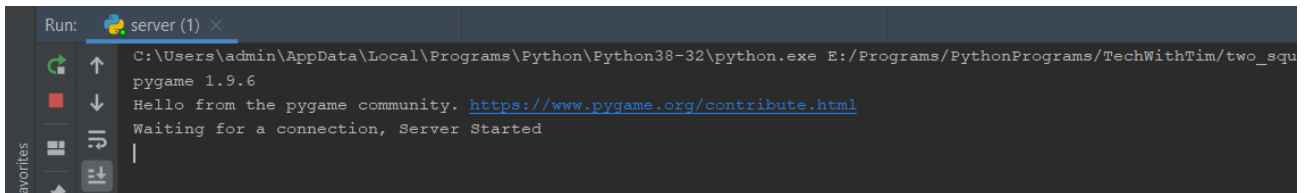
Таблиця 1.1

1 \ 2	Співпрацювати	Зрадити
Співпрацювати	Обидва отримують +2 монетки.	Перший отримує -1 монетку, а другий: +3.
Зрадити	Перший отримує +3 монетки, а другий: -1.	Обидва отримують 0 монеток.

Етапи розробки програми

Етап I. Створення серверу.

Створимо сервер і перевіримо правильність його роботи. Запускаємо скрипт server.py, файл, де знаходиться наш сервер.



```

Run: server (1) x
C:\Users\admin\AppData\Local\Programs\Python\Python38-32\python.exe E:/Programs/PythonPrograms/TechWithTim/two_squ
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Waiting for a connection, Server Started
|

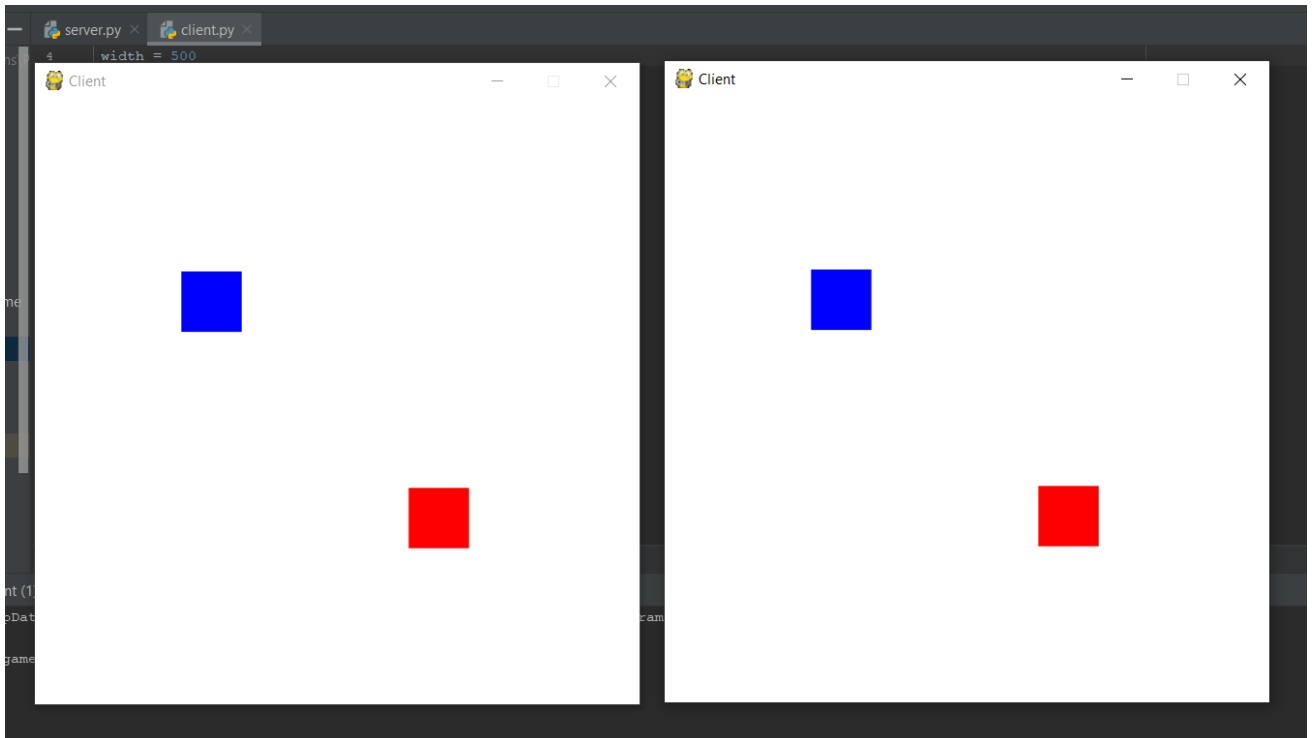
```

Бачимо відповідне повідомлення, що сервер запустився й очікує на підключення клієнтів.

Етап II. Створення клієнту й тестування синхронізації.

Створимо клієнт, підключимося до серверу за допомогою бібліотек socket та _thread і створимо простеньку гру, кінцевою метою якої стане наступне: якщо ти рухаєш свій квадрат у своєму діалоговому вікні, він рухатиметься й в іншого користувача в його діалоговому вікні.

Двічі запустимо `client.py`, таким чином створивши двох клієнтів, які гратимуть цю гру.



При цьому сервер видає нам повідомлення, хто до нього підключився.

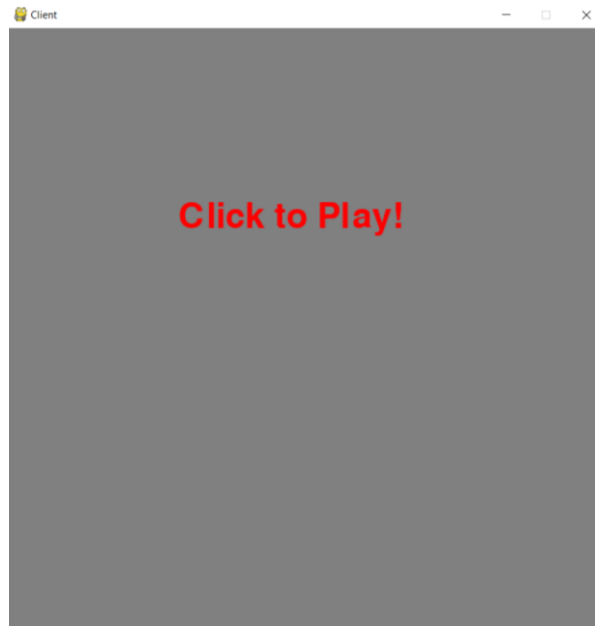
```
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Waiting for a connection, Server Started
Connected to: ('192.168.1.5', 3119)
Connected to: ('192.168.1.5', 3120)

Process finished with exit code -1
```

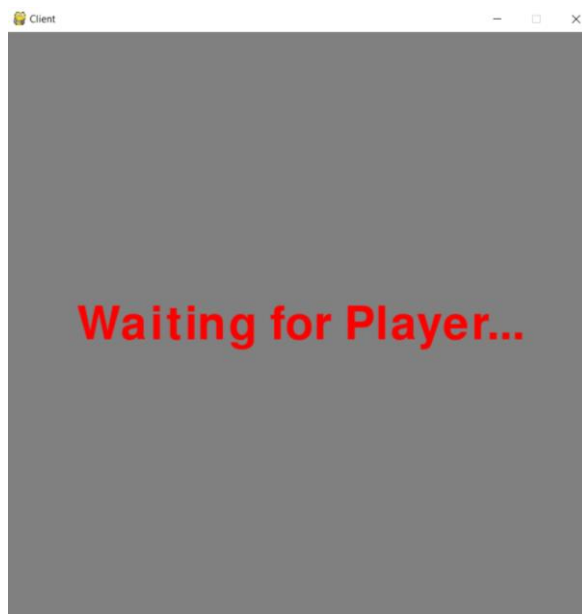
Етап III. Розробка безпосередньо кінцевої гри.

Розробимо безпосередньо кінцеву гру із застосуванням такого ж підходу, як і в попередньому етапі, сервер-клієнт. На відміну, від попередньої гри, де могли підключитися лише двоє гравців, тут розроблений алгоритм дозволяє підключатися необмеженій кількості охочих пограти її. Для кожної нової пари

гравців створюється нова гра, яка існує незалежно від інших. Коли один із гравців підключається до сервера (запускає скрипт client.py) він отримує наступне діалогове вікно. (При цьому зрозуміло, що скрипт server.py уже запущений).

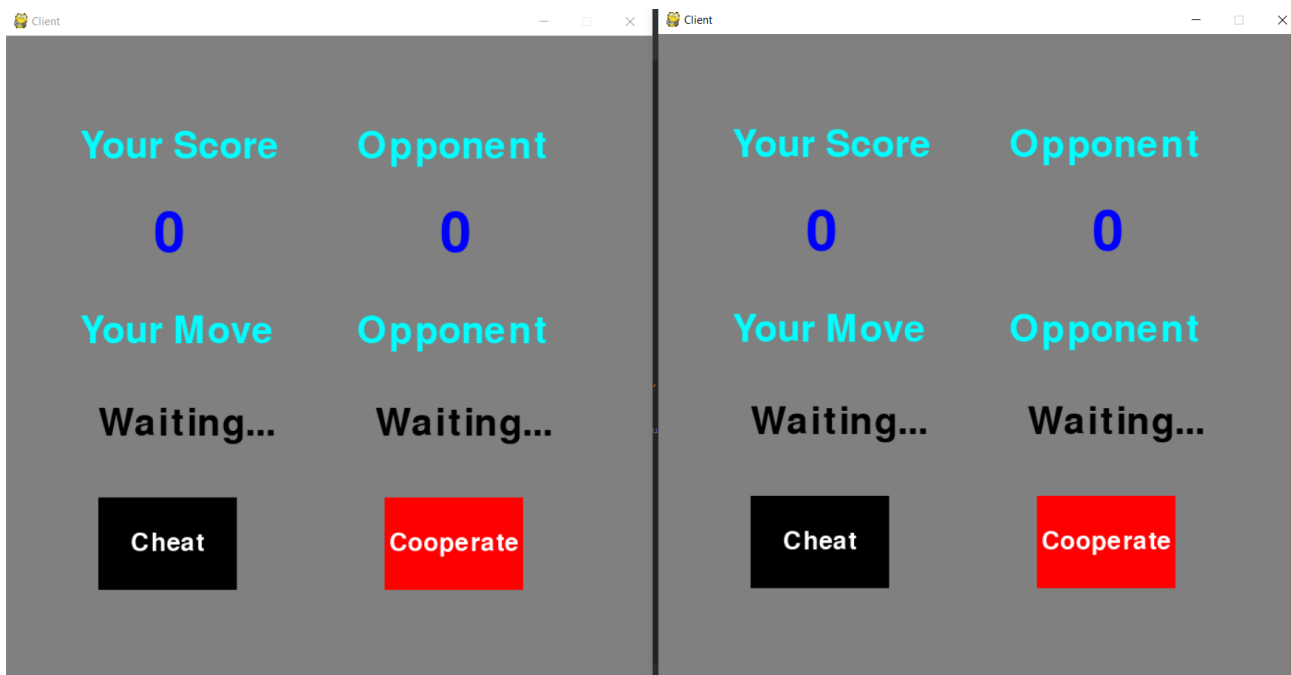


Коли він натисне на будь-яке місце цього поля, створиться нова гра, а гравець отримає наступне повідомлення.

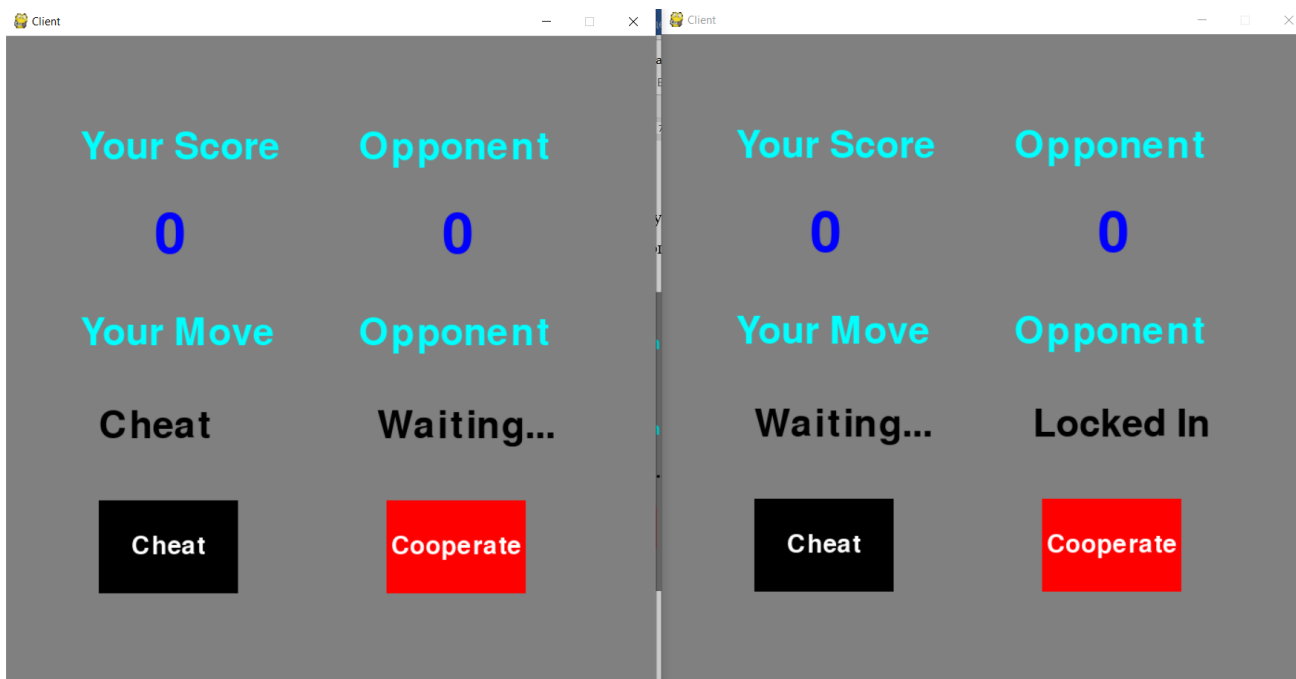


Підключимо другого гравця до серверу, ще раз запустивши скрипт client.py. Після того, як другий гравець підтвердить свою готовність до гри

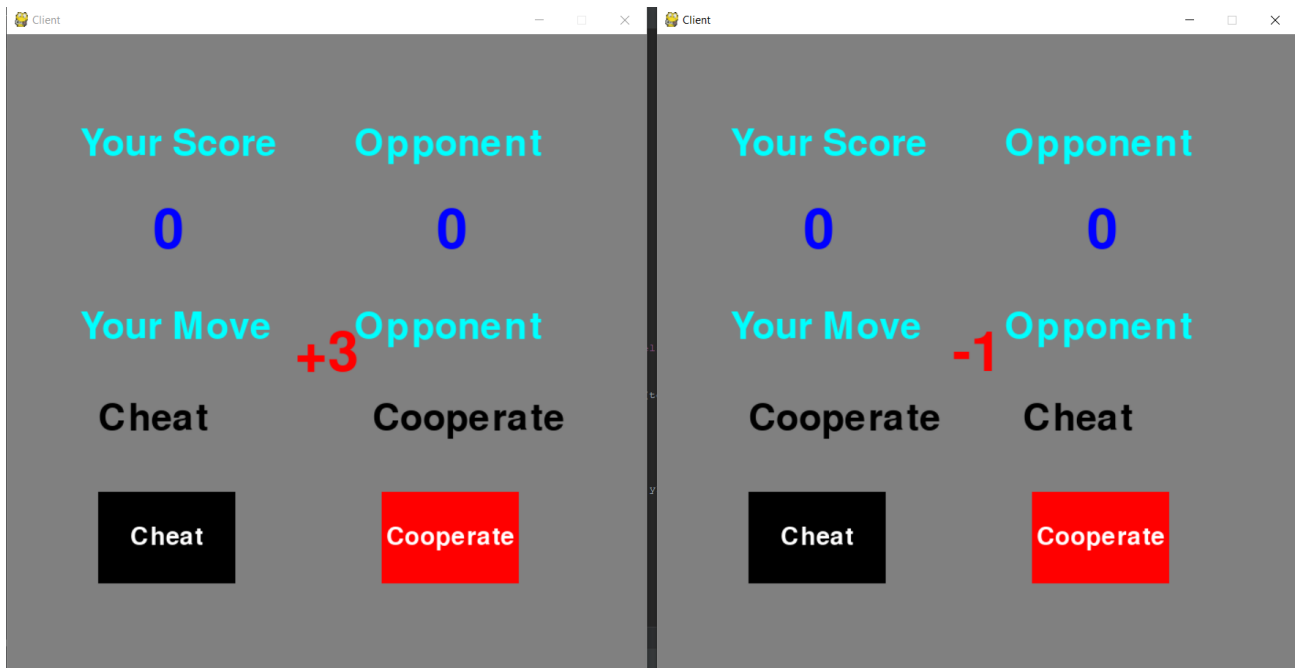
(натисне пішкою в будь-якому місці наявного діалогового вікна), одразу ж запуситься гра, де ціллю кожного із гравця стане набрати якомога більше балів.



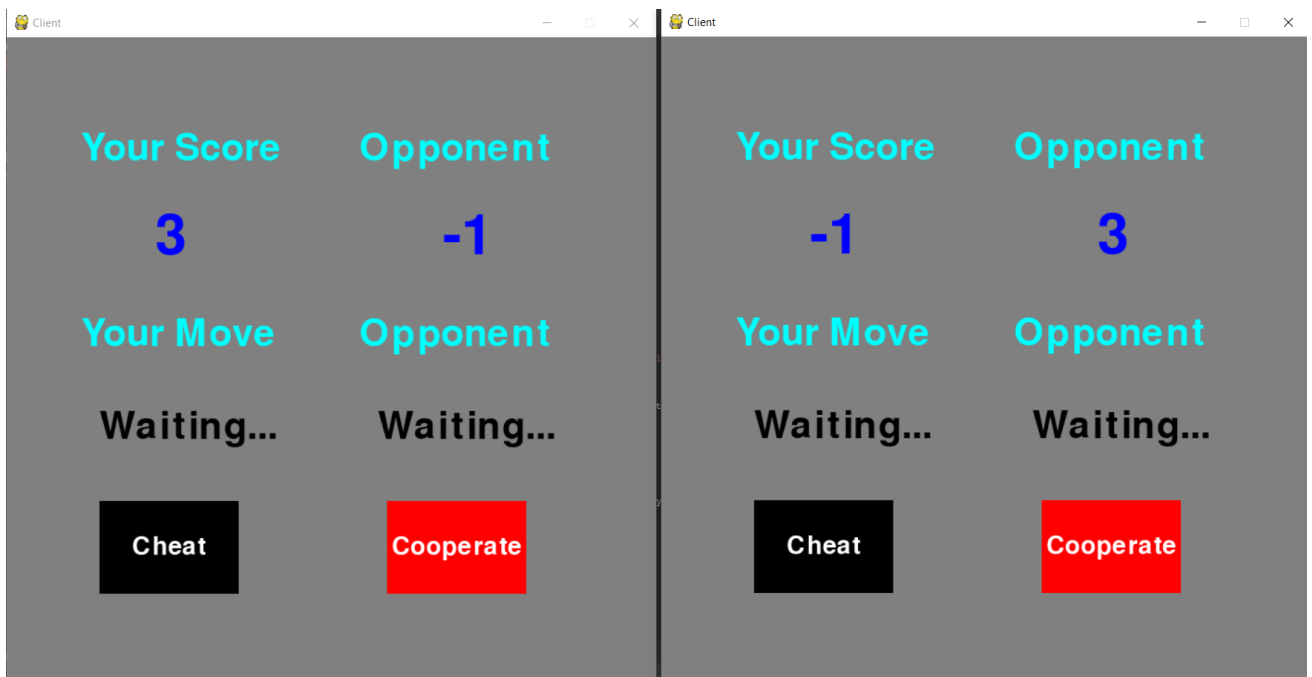
Нехай перший гравець вибирає обирає Cheat, другий отримає повідомлення, що перший гравець уже зробив свій вибір.



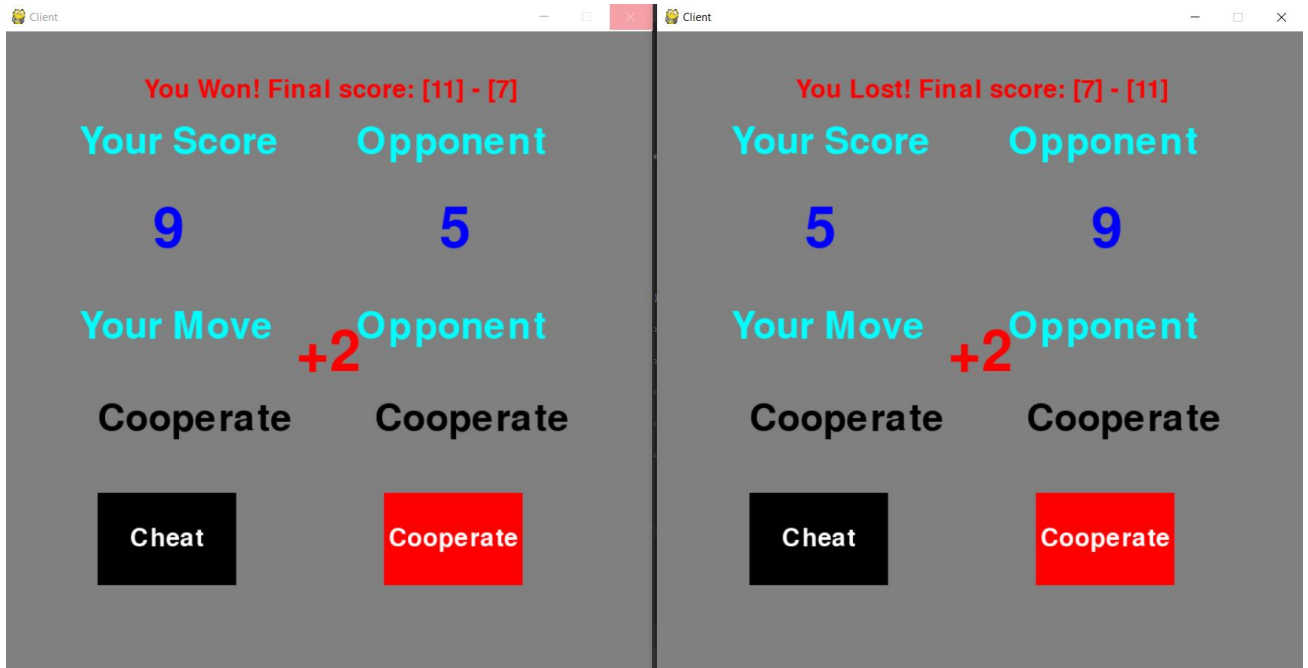
Нехай другий гравець обирає Cooperate. Це означає, що перший раунд гри зіграно, кожен гравець отримає повідомлення, скільки монеток він заробив або втратив. У даному випадку перший гравець заробив 3 монетки, а другий втратив 1.



Після цього розпочинається 2 раунд і так далі.



Гра триває 6 раундів. Після 6 раунду вверху діалогового вікна з'явиться відповідний напис для гравця, виграв він, програв чи зіграв у нічию й кінцевий рахунок гри.



Після цього кожен із гравців може розпочати нову гру, при чому не обов'язково із тим самим суперником, суперником гравця стане той, хто швидше із ним підключиться до серверу.

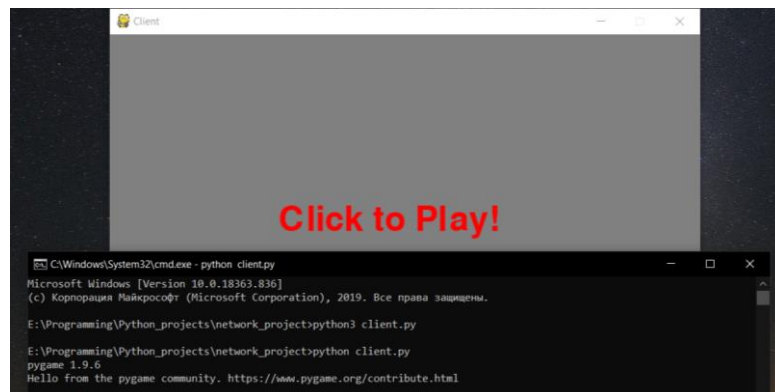
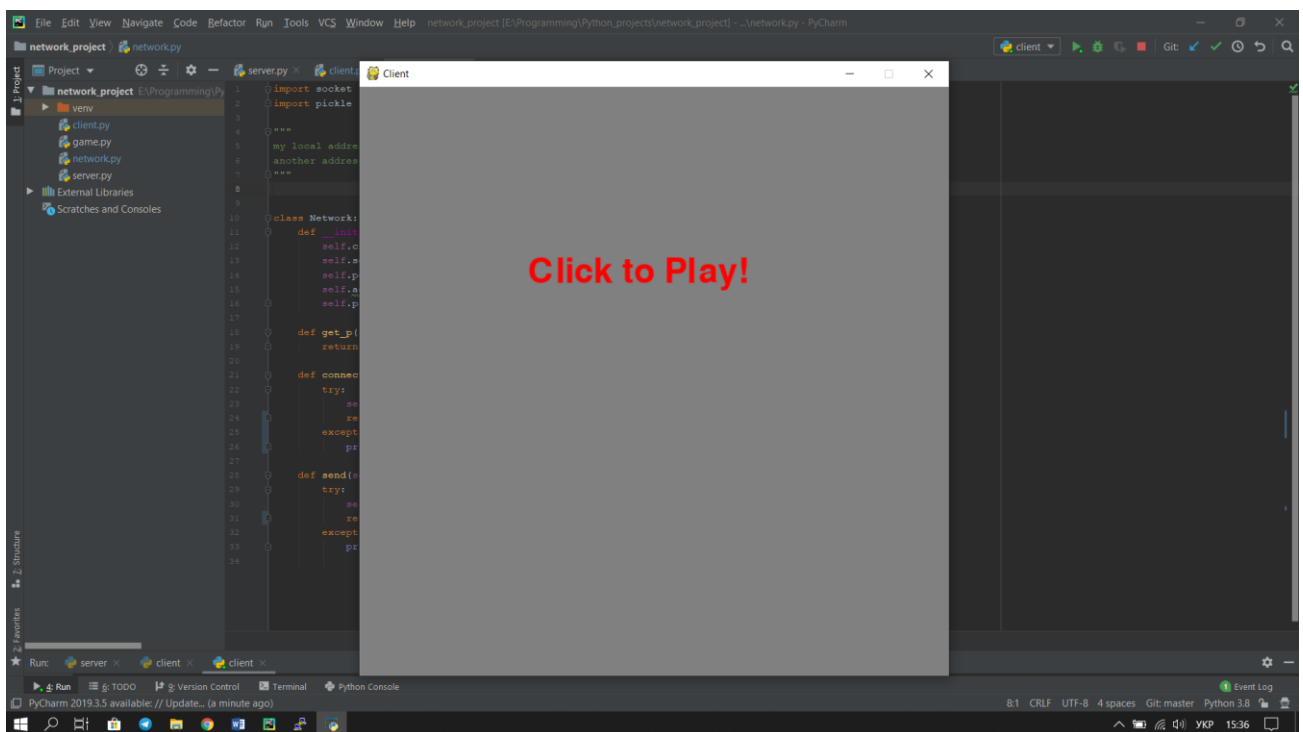
Етап IV. Створення віддаленого серверу.

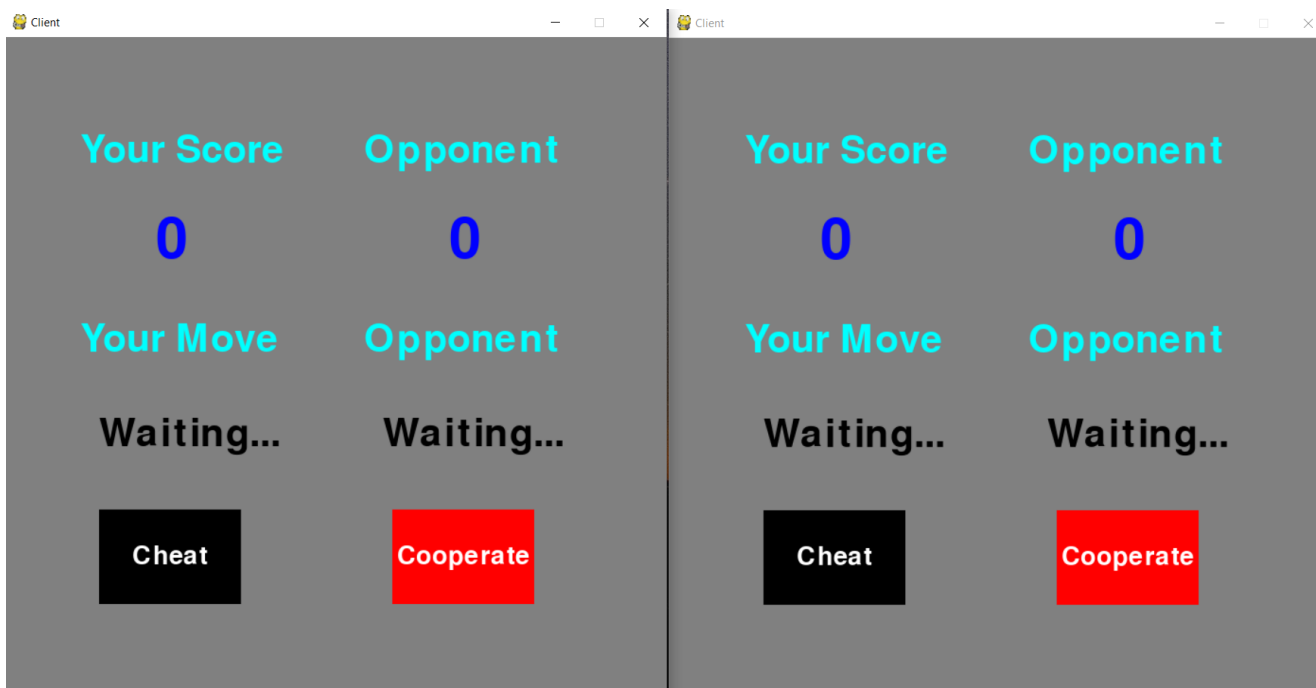
У попередньому етапі гравці могли зіграти цю гру лише по локальній IP-адресі, тобто знаходячись досить на невеликій відстані одне від одного. Кінцевою метою цієї програми було саме те, щоб підключатися можна було із будь-якої точки світу. Я зареєструвався на Linode й отримав 20\$ безкоштовно для запуску свого віддаленого серверу. Не буду вдаватися в деталі, як налаштовувати цей віддалений сервер, продемонструю лише правильність роботи програми, підключаючись до цього серверу. Його адреса: 139.162.185.251.

Запустимо наш сервер, який я закинув у віртуальне середовище Linux.

```
root@localhost:~/python-server# ls
game.py  __pycache__  server.py
root@localhost:~/python-server# python3 server.py
Waiting for a connection, Server Started
```

Запустимо один клієнт через PyCharm, а другий через консоль (попередньо змінивши IP-адресу серверу).





Як ми бачимо, усе працює доволі правильно.

Висновки

Отже, ми розробили кінцеву гру-мультиплеєр, у яку можуть грати гравці із будь-якої точки світу. Звісно, її зовнішній вигляд залишає бажати кращого, проте в майбутньому я прагну розробити кінцеву гру, яка включатиме і мультиплеєр, і гру проти різних ботів (у мене вже є закінчений програмний продукт такого плану, який я писав на 2 курсі, як курсову роботу з ООП. Там користувач мав можливість поспробувати свої сили проти різних ботів).

Посилання

Посилання на проєкт на GitHub:

https://github.com/VolodymyrVozniak/network_project