

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Phát triển game cho 2 người chơi sử dụng Pygame

Giảng viên hướng dẫn: Từ Lãng Phiêu
Nhóm: 21
Sinh viên: Trần Tuấn Sang - 3121560077
Nguyễn Hoàng Tuấn - 3121410557
Con Kiên Huy - 3121410006

Mục lục

1	Giới thiệu	2
1.1	Python	2
1.2	Game Street Fight	2
2	Các thư viện sử dụng	3
2.1	Pygame	3
2.2	Socket	4
2.3	Threading	5
3	Street Fight	6
3.1	Cấu trúc mã nguồn	6
3.2	Cài đặt Street Fight	6
3.3	Hướng dẫn chơi game Street Fight	11
3.3.1	Vào game	11
3.3.2	Điều khiển nhân vật	12
3.3.3	Chat	13
3.4	Giao diện khác của Street Fight	15
4	Đóng góp của các thành viên	17
5	Thiếu sót của đồ án	18
6	Kết luận	18



1 Giới thiệu

1.1 Python

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Python được ra mắt lần đầu vào năm 1991, với phiên bản 0.9.0. Năm 2000, Python 2.0 ra mắt, và tương thích ngược với phiên bản 0.9.0. Python 3.0 được ra mắt vào năm 2008 với nhiều cải tiến. Phiên bản mới nhất (tính tới thời điểm viết báo cáo) là Python 3.12.3.

1.2 Game Street Fight

Street Fight là một game đối kháng 2 người chơi, được phát triển bằng ngôn ngữ Python, sử dụng thư viện Pygame.

- Một ván Street Fight sẽ bắt đầu khi có đủ 2 người chơi kết nối với nhau.
- Khi đã có đủ 2 người chơi, trò chơi sẽ bắt đầu.
 - Người chơi 1 sẽ đóng vai hiệp sĩ.
 - Người chơi 2 sẽ đóng vai pháp sư.
- Một ván đấu sẽ bắt đầu sau khi 3 giây đếm ngược kết thúc, nhiệm vụ của mỗi người chơi là phải tiêu diệt đối phương.
- Người chơi sử dụng các phím A, S, D để di chuyển nhân vật của mình.
- Mỗi người chơi sẽ có 100HP, và mỗi lần tấn công thì sẽ gây ra 10HP sát thương cho đối phương (nghĩa là nếu muốn thắng, thì cần phải tấn công đối phương 10 lần).
- Một ván đấu kết thúc khi có 1 trong 2 người chơi không còn máu.
- Và trò chơi vẫn sẽ tiếp diễn.
- Trong khi chơi, 2 người chơi có thể chat với nhau. Người chơi sẽ click vào ô chat để có thể chat. Khi đó, các phím mà người chơi gõ sẽ không có ảnh hưởng đến gameplay.
- Để tiếp tục trò chơi, người chơi click vào bất cứ đâu của màn hình (trừ ô chat) để tiếp tục trò chơi.

GitHub của đồ án: <https://github.com/ConKienHuy/Street-Fight>



2 Các thư viện sử dụng

2.1 Pygame



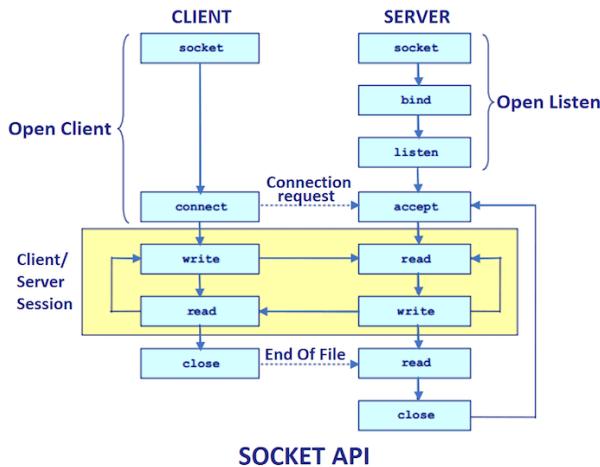
Pygame là một module Python đa nền tảng dành cho việc phát triển trò chơi điện tử. Nó tích hợp cả đồ họa máy tính và thư viện âm thanh được tạo ra đặc biệt để tương thích với ngôn ngữ lập trình Python.

Pygame giúp tạo ra các trò chơi đơn giản hoặc phức tạp bằng ngôn ngữ lập trình Python một cách hiệu quả và linh hoạt. Với sự hỗ trợ đa nền tảng và tính linh hoạt cao, Pygame đã trở thành công cụ phổ biến trong cộng đồng lập trình viên để tạo ra các ứng dụng giải trí đa dạng, hấp dẫn.

Ưu điểm sử dụng Pygame:

- Pygame sử dụng Simple DirectMedia Layer (SDL), một thư viện phát triển đa nền tảng cho phép các nhà phát triển có thể truy cập vào phần cứng máy tính như đồ họa, âm thanh và thiết bị đầu vào.
- Xây dựng các trò chơi trên nhiều nền tảng khác nhau như Windows, MacOS, Linux thậm chí là cả các thiết bị di động.
- Nhà phát triển có thể quản lý tất cả các yếu tố trong quá trình phát triển trò chơi. Đó có thể là các chức năng như xuất đồ họa, xử lý sự kiện, hoạt ảnh, hiệu ứng âm thanh và phát lại nhạc.
- Cung cấp nhiều chức năng mở rộng hỗ trợ nhà phát triển tập trung phát triển trò chơi.
- API trực quan và dễ hiểu, hỗ trợ người mới sử dụng hay cả những nhà phát triển có kinh nghiệm đều có thể truy cập được.
- Nguồn tài nguyên và tài liệu phong phú, các nhà phát triển có thể sử dụng các mã nguồn mở miễn phí để phát triển dự án của mình.
- Tính đa phương tiện giúp nhà phát triển có thể ứng dụng để xử lý hình ảnh hay video, mô phỏng, công cụ giáo dục.

2.2 Socket



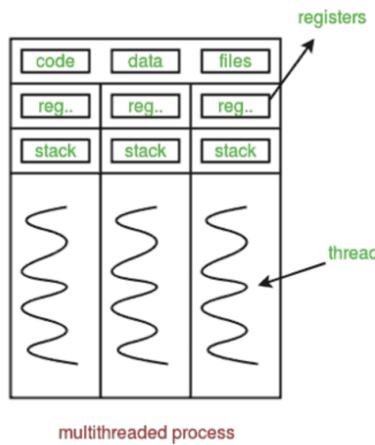
Thư viện Socket trong Python là một công cụ giúp cho các lập trình viên có thể lập trình các ứng dụng mạng, như ứng dụng chat... bằng cách giao tiếp với các socket mạng.

Lập trình socket là cơ chế của các trình duyệt hiện nay. Có một server và một hay nhiều client. Mục tiêu của lập trình socket nói chung và sử dụng thư viện Socket trong Python nói riêng là nhằm tạo ra kết nối giữa server và client, hay giữa các client với nhau.

Các hàm cơ bản của thư viện Socket:

- `socket.socket()`: tạo ra socket.
- `socket.listen()`: lắng nghe sự kết nối của các client.
- `socket.connect()`: kết nối với server
- `socket.send()`: gửi dữ liệu. Dữ liệu được gửi phải ở dạng bytes, thông thường đi kèm với câu lệnh `encode()`. VD: `socket.send("Hello World".encode())`. Mặc định chuẩn encode là UTF-8.
- `socket.recv()`: nhận dữ liệu. Dữ liệu được nhận ở dạng bytes, nên nếu muốn đọc dữ liệu thì phải decode dữ liệu bằng câu lệnh `decode()`. VD: `socket.recv(1024).decode()`.
- `socket.close()`: đóng socket.

2.3 Threading



Thư viện Threading trong Python giúp chương trình thực thi nhiều tác vụ song song (multithreading), nâng cao hiệu suất của phần mềm.

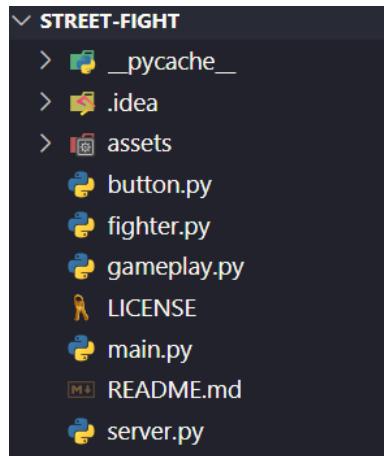
Lợi ích của lập trình đa luồng:

- **Sử dụng tối ưu tài nguyên CPU:** Đa luồng cho phép nhiều nhiệm vụ được thực thi đồng thời trên nhiều lõi CPU, giúp cải thiện hiệu suất tổng thể của chương trình.
- **Tăng tốc độ xử lý:** Các tác vụ I/O (như đọc/ghi file, truy cập mạng) thường bị chặn chờ đợi phản hồi. Với đa luồng, các tác vụ khác có thể tiếp tục thực thi trong khi chờ đợi, giúp giảm thời gian chết và tăng tốc độ xử lý chung.
- **Cải thiện khả năng phản hồi:** Khi một luồng bị chặn, các luồng khác vẫn có thể tiếp tục thực thi, giúp chương trình phản hồi nhanh hơn với người dùng.
- **Chia nhỏ các nhiệm vụ phức tạp:** Đa luồng giúp chia nhỏ các nhiệm vụ phức tạp thành các luồng nhỏ hơn, dễ quản lý và dễ dàng thực thi đồng thời.
- **Tăng tính linh hoạt:** Việc sử dụng đa luồng cho phép điều chỉnh dễ dàng số lượng luồng được thực thi đồng thời, phù hợp với nhu cầu và tài nguyên hệ thống.
- **Tạo giao diện người dùng mượt mà:** Đa luồng giúp đảm bảo giao diện người dùng mượt mà và đáp ứng nhanh chóng, ngay cả khi đang thực hiện các tác vụ nặng.



3 Street Fight

3.1 Cấu trúc mã nguồn



Một số file chính:

- File server.py để tạo server cho màn chơi, là file cần phải chạy đầu tiên.
- File gameplay.py để chứa các xử lý trong màn chơi (như đếm ngược, tiếp nhận input từ bàn phím).
- File fighter.py để chứa các xử lý liên quan đến người chơi (fighter), như là di chuyển, các hoạt hình, âm thanh...
- File main.py để chứa màn hình chính, là file sẽ chạy sau khi chạy file server.py
- File button.py cần thiết cho việc tạo nút bấm (button).

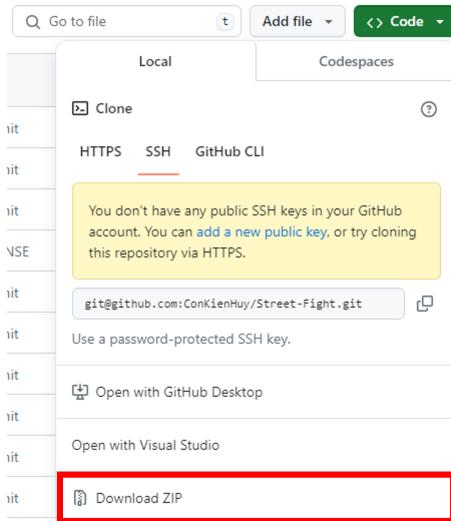
3.2 Cài đặt Street Fight

Yêu cầu tối thiểu:

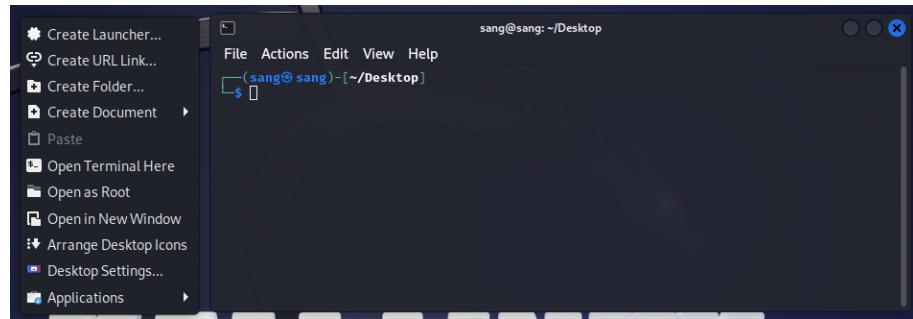
- Python 3.6 trở lên.
- Giao diện màn hình từ 800×600 trở lên.
- Có kết nối với internet để có thể chơi game.

Các bước cài đặt

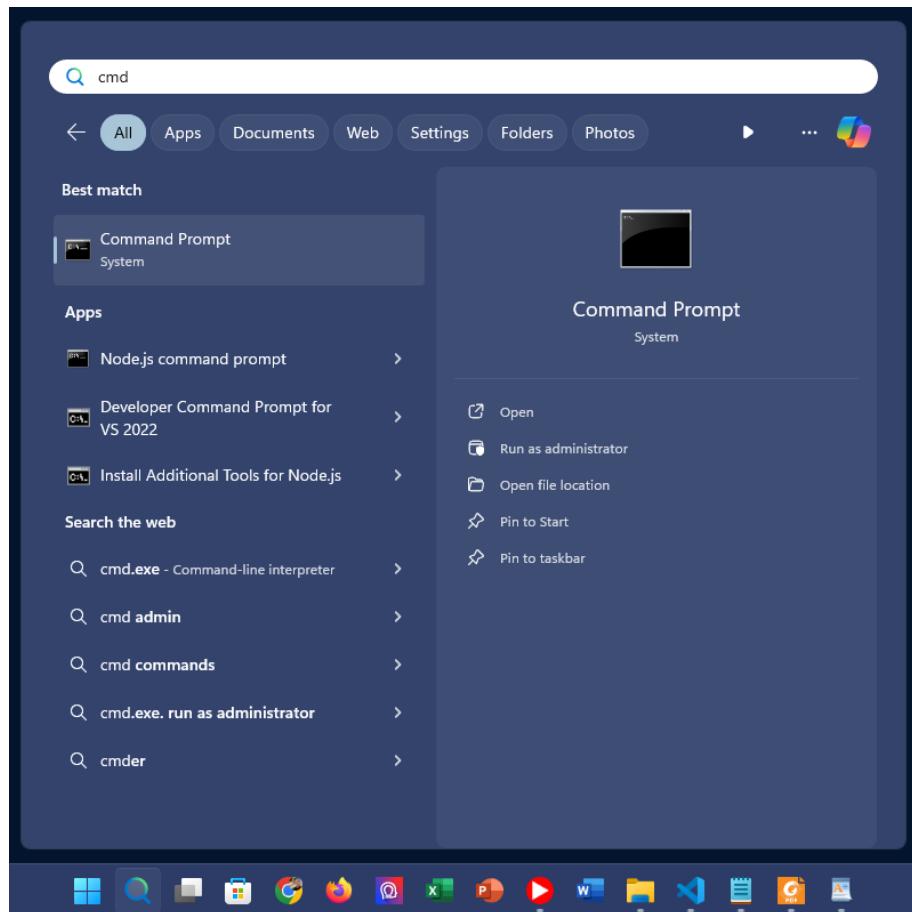
- **Bước 1:** Vào GitHub của nhóm, chọn “Code”, rồi nhấn vào “Download ZIP”.



- **Bước 2:** Vào màn hình console (CLI)



- Ở Windows, gõ Windows + R rồi gõ cmd, nhấn Enter để chạy, hoặc search cmd (Windows 8 trả lên)



- **Bước 3:** Tải thư viện Pygame

```
C:\Windows\system32\cmd.exe: + - ×
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sang>pip install pygame
Collecting pygame
  Using cached pygame-2.5.2-cp312-cp312-win_amd64.whl.metadata (13 kB)
  Using cached pygame-2.5.2-cp312-cp312-win_amd64.whl (10.8 MB)
Installing collected packages: pygame
Successfully installed pygame-2.5.2

C:\Users\Sang>
```

- Hầu hết các distro của Linux đều đã có sẵn Pygame



```
sang@sang: ~/Desktop
File Actions Edit View Help
└─(sang㉿ sang)-[~/Desktop]
$ pip install pygame
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pygame in /usr/lib/python3/dist-packages (2.5.2)
└─(sang㉿ sang)-[~/Desktop]
$
```

- **Bước 4:** Di chuyển tới folder của game sử dụng câu lệnh *cd* (cả Windows và Linux).

```
sang@sang: ~/Desktop/Multiplayer_Python
File Actions Edit View Help
└─(sang㉿ sang)-[~/Desktop]
$ cd Multiplayer_Python
└─(sang㉿ sang)-[~/Desktop/Multiplayer_Python]
$
```

- **Bước 5:** Kiểm tra IP của máy sẽ chạy server.py

- Ở Windows, sử dụng câu lệnh *ipconfig*, copy IPv4 ở ethernet (nếu đang kết nối ethernet), hoặc wifi (khuyến khích).

```
C:\Windows\system32\cmd.e... + - 
Ethernet adapter Ethernet 2:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::497e:366a:50da:c11%17
IPv4 Address . . . . . : 192.168.56.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :

Wireless LAN adapter Local Area Connection* 1:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 2:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . :
IPv6 Address. . . . . : 2405:4802:93c1:1a75:36f0:db88:4ec6:886d
IPv6 Address. . . . . : 2405:4802:93c1:1a75:ffff:ffff:ffff:ffca
Temporary IPv6 Address. . . . . : 2405:4802:93c1:1a75:d9bf:5744:2e63:f5e7
Link-local IPv6 Address . . . . . : fe80::4d7e:6c9a:beb1:63e9%7
IPv4 Address. . . . . : 192.168.1.225
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1

C:\Users\Sang>
```

- Ở Linux, sử dụng câu lệnh *hostname -i*.



```
sang@sang: ~/Desktop
File Actions Edit View Help
(sang@sang)-[~/Desktop]
$ hostname -i
127.0.1.1
```

- **Bước 6:** Ở máy server, copy IPv4 mới kiểm tra ở bước 5 vào file *server.py*. Ở các máy client, copy IPv4 của server vào *main.py*.

```
server.py > ...
1 import socket, threading, gameplay, pygame
2
3 #HOST = socket.gethostname()
4 HOST = "192.168.1.225"
5 PORT = 5555
6 MAX_CLIENTS = 2
7
8 clients = []
9
10 def broadcast(key):
11     for client in clients:
12         client.send(key)
13
14 def handle_client(client_socket):
15     while True:
16         try:
17             key = client_socket.recv(1024)
18             if not key:
19                 break
20             print("Received:", key.decode())
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74 def connect_to_server(pPlayer_name):
75     # Kết nối đến server
76     HOST = "192.168.1.225"
77     PORT = 5555
78
79     try:
80         client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
81         client_socket.connect((HOST, PORT))
82         client_socket.send(pPlayer_name.encode())
83         print("Connected to the server as Player")
84         start_game(client_socket, pPlayer_name)
85     except Exception as e:
86         print("Error:", e)
87         traceback.print_exc()
88
89 def main_menu():
90     while True:
91         SCREEN.blit(BG, (0, 0))
```

- **Bước 7:** Chạy file *server.py*.

- Lưu ý: Câu lệnh có thể là *py server.py* hoặc *python3 server.py*.
- Trong trường hợp kiểm thử của nhóm, Windows sử dụng câu lệnh *py*, còn Linux sử dụng câu lệnh *python3*.
- Nên nếu 1 trong 2 không hiệu quả thì thử đổi câu lệnh.
- Câu lệnh trong Windows 11:

```
PS D:\SGU\DoAnCode\Multiplayer_Python> py server.py
pygame 2.5.2 (SDL 2.28.3, Python 3.12.1)
Hello from the pygame community. https://www.pygame.org/contribute.html
Server is listening on 192.168.56.1:5555
```

- Câu lệnh trong Kali Linux:



```
sang@sang: ~/Desktop/Multiplayer_Python
File Actions Edit View Help
└─(sang@sang)-[~/Desktop]
$ cd Multiplayer_Python
└─(sang@sang)-[~/Desktop/Multiplayer_Python]
$ python3 server.py
<frozen importlib._bootstrap>:241: RuntimeWarning: Your system is avx2 capable but pygame was not built with support for it. The performance of some of your blits could be adversely affected. Consider enabling compile time detection with environment variables like PYGAME_DETECT_AVX2=1 if you are compiling without cross compilation.
pygame 2.5.2 (SDL 2.28.5, Python 3.11.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
Server is listening on 127.0.1.1:5555
```

- **Bước 8:** Chạy *main.py* để khởi tạo game.

```
sang@sang: ~/Desktop/Multiplayer_Python
File Actions Edit View Help
└─(sang@sang)-[~/Desktop]
$ cd Multiplayer_Python
└─(sang@sang)-[~/Desktop/Multiplayer_Python]
$ python3 main.py
<frozen importlib._bootstrap>:241: RuntimeWarning: Your system is avx2 capable but pygame was not built with support for it. The performance of some of your blits could be adversely affected. Consider enabling compile time detection with environment variables like PYGAME_DETECT_AVX2=1 if you are compiling without cross compilation.
pygame 2.5.2 (SDL 2.28.5, Python 3.11.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

- **Bước 9:** Ở máy thứ 2, chạy *main.py* để khởi tạo game.

```
PS D:\SGU\DoAnCode\Multiplayer_Python> py main.py
pygame 2.5.2 (SDL 2.28.3, Python 3.12.1)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

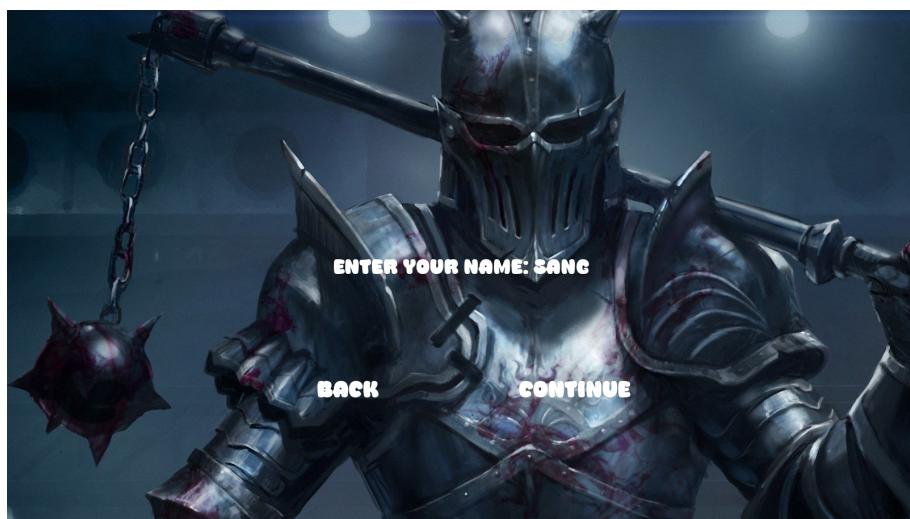
3.3 Hướng dẫn chơi game Street Fight

3.3.1 Vào game

Bước 1: Từ màn hình Main menu, nhấn *Play*.



Bước 2: Nhập tên người chơi. Ở đây người chơi này đang nhập tên *Sang*.



Bước 3: Nhấn Continue hoặc Enter để bắt đầu màn chơi.

Bước 4: Làm tương tự với máy thứ 2

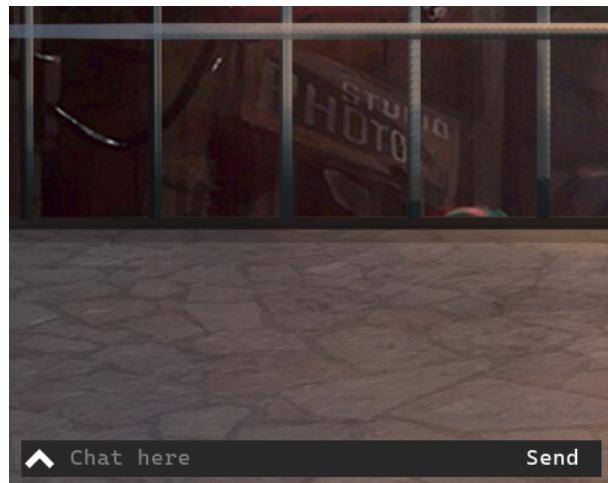
3.3.2 Điều khiển nhân vật

A	Di chuyển nhân vật về bên trái
D	Di chuyển nhân vật về bên phải
W	Nhảy
R hoặc T	Tấn công

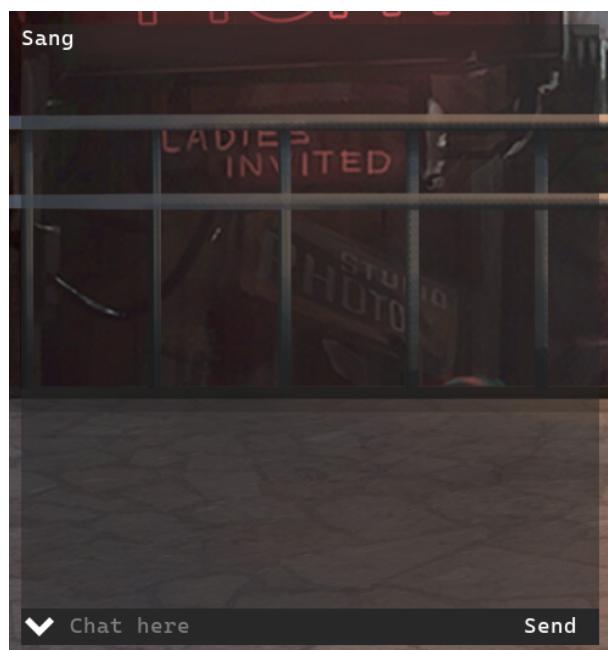


3.3.3 Chat

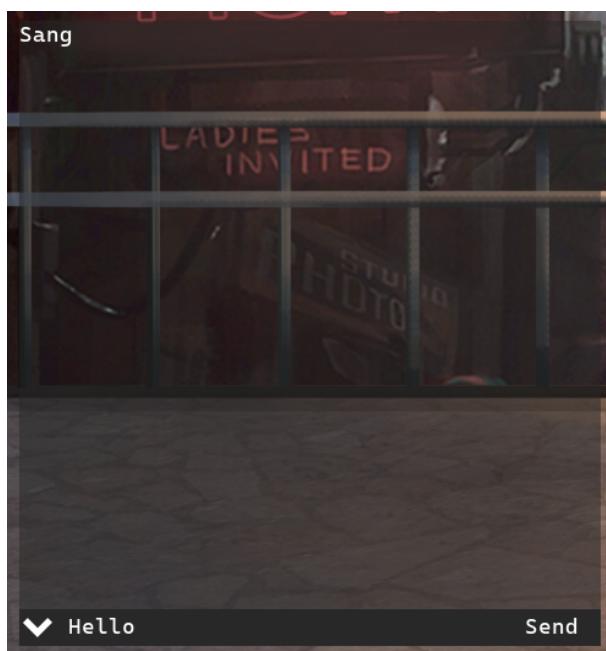
Mặc định khi bắt đầu trò chơi, thì giao diện chat như thế này:



Khi chúng ta mở rộng bằng cách nhấn **▲**, thì lịch sử chat sẽ được mở rộng:

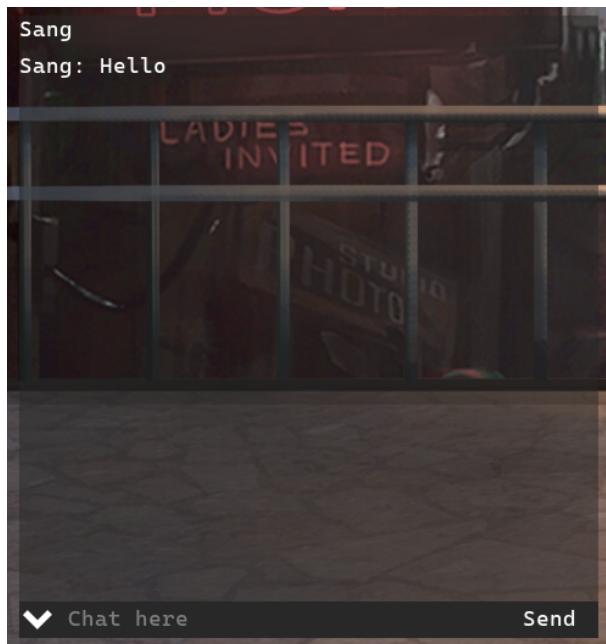


Chúng ta bắt đầu nhập nội dung bằng cách nhấn vào chữ Chat here (để game nhận biết là người chơi đang chat), sau đó nhập dòng chữ (Ví dụ bên dưới là “Hello”).

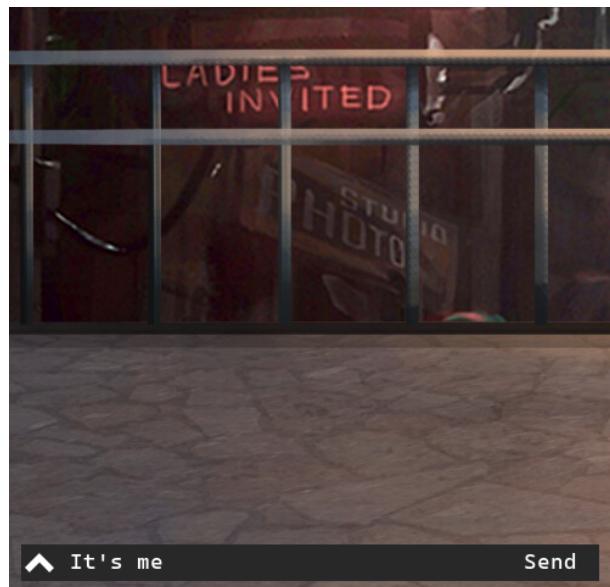


Sau đó, nhấn Enter hoặc nút “Send” để gửi.

Khi ấy “Hello” của người chơi Sang được ghi vào lịch sử chat.



Chúng ta có thể thu gọn lịch sử chat bằng cách nhấn , kết quả như hình bên dưới:



3.4 Giao diện khác của Street Fight

Giao diện đêm ngược:



Giao diện trận đấu:



Giao diện Victory (có người chơi chiến thắng):



Giao diện bắt đầu trận đấu thứ 2 trở đi:



4 Đóng góp của các thành viên

- Công việc chung:
 - Lên ý tưởng đồ án
- Công việc cá nhân:
 - Nguyễn Hoàng Tuấn:
 - * Khởi tạo đồ án. Tìm kiếm tài liệu, tài nguyên để hỗ trợ cho đồ án.
 - * Tạo gameplay, màn hình Main menu, nhập tên người chơi, kết nối người chơi về server.
 - * Tìm kiếm các tài nguyên cho trò chơi như nhân vật, âm thanh, font chữ...
 - * Review code, sửa lỗi đồ án.
 - Con Kiến Huy:
 - * Tạo gameplay, fighter.
 - * Tìm kiếm các tài nguyên cho trò chơi như nhân vật, âm thanh, font chữ...
 - * Review code, sửa lỗi đồ án.
 - Trần Tuấn Sang:
 - * Tạo khung chat.
 - * Kết nối 2 người chơi (phản gameplay).



* Viết báo cáo PDF, readme.

5 Thiếu sót của đồ án

- Chỉ có các đòn tấn công cơ bản, truyền input giữa 2 máy có lúc bị delay (trừ tấn công).
- Pygame GUI chưa thực sự có nhiều công cụ hỗ trợ tạo ra các thành phần như là Button, Textbox, Scrollbar... nên tất cả thành phần đều sẽ phải làm một cách thủ công.
- Do không có Textbox thiếu sót một số chức năng cơ bản, như:
 - Chưa thể chat bằng tiếng Việt, cũng như sử dụng emoji.
 - Cho con trỏ soạn thảo đi bất cứ đâu trong khung chat.
 - Chưa thể dùng Ctrl+A để chọn toàn bộ khung chat.
 - Chưa thể dùng phím Delete để xóa bên phải (do không thể cho con trỏ soạn thảo đi bất cứ đâu trong khung chat).
 - Không thể coi được lịch sử chat.
- Không thể liên kết được với nhiều thư viện.
 - Ban đầu, nhóm có ý định liên kết Tkinter vào giao diện chat (do Tkinter có tất cả thành phần thiếu sót nói trên).
 - Nhưng do chỉ có thể liên kết Pygame vào cửa sổ Tkinter, nhưng không thể ngược lại, nên nhóm quyết định bỏ.

6 Kết luận

Thông qua đồ án, chúng em kết luận được 1 điều "*"Chơi game là một chuyện, làm game là chuyện khác"*". Chúng ta có thể rất thích những game, như là Liên Quân, FreeFire,... nhưng chỉ khi chúng ta bắt đầu làm game, kể cả cơ bản nhất như là cờ caro, hay là bắt đầu đê tài của nhóm (game đối kháng 2 người), thì có rất rất nhiều vấn đề xảy ra.

Trong suốt quá trình của đồ án, nhóm đã tìm hiểu được quy trình làm ra một tựa game đơn giản, và tự hỏi bản thân rằng "*"Một con game đơn giản như vậy đã tốn nhiều công sức, vậy những game mà chúng ta chơi hàng ngày cần phai tốn bao nhiêu?"*"



Nhưng đó cũng là những trải nghiệm rất thú vị, vì được nhìn thấy quá trình các nhân vật từ không biết cử động cho đến có thể chiến đấu, từ màn hình đầy lỗi giao diện, cho đến một màn hình có thể thích nghi với nhiều độ phân giải khác nhau, vì được chọn nhạc, chọn nhân vật cho chính tựa game của chính mình.

Ngoài ra, đồ án môn học Phần mềm mã nguồn mở cho chúng em làm quen với socket programming, một lĩnh vực mà chưa có môn học nào giảng dạy trước đây.

Được tự tay tạo ra server, những phiên chat giữa 2 người chơi với nhau, và liên kết hành động của 2 người chơi, tất cả đều nhờ thư viện Socket và Threading của Python, cũng như là những kiến thức của môn học này.

Có thể tương lai nhóm sẽ có thể đi theo game, có thể không, nhưng chắc chắn là những bài học kinh nghiệm về lựa chọn đề tài, tài nguyên, sửa lỗi, lập trình socket, và cuối cùng là license sẽ đi theo nhóm trong suốt quá trình trở thành các kỹ sư Công nghệ thông tin trong tương lai.



Tài liệu

- [1] Python (programming language), “[link: https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))”, lần truy cập cuối: 15/04/2024.
- [2] Download Python, “[link: https://www.python.org/downloads/](https://www.python.org/downloads/)”, lần truy cập cuối: 05/05/2024.
- [3] Pygame là gì? Tất cả những gì bạn cần biết về lập trình Pygame trong Python, “[link: https://funix.edu.vn/chia-se-kien-thuc/lap-trinh-game-voi-pygame/](https://funix.edu.vn/chia-se-kien-thuc/lap-trinh-game-voi-pygame/)”, *Đặc điểm nổi bật của lập trình Pygame*, lần truy cập cuối: 23/04/2024.
- [4] socket — Low-level networking interface, “[link: https://docs.python.org/3/library/socket.html](https://docs.python.org/3/library/socket.html)”, lần truy cập cuối: 23/04/2024.
- [5] Thread trong Java, “[link: https://rikkei.edu.vn/thread-trong-java/](https://rikkei.edu.vn/thread-trong-java/)”, *Lợi ích của việc sử dụng Thread*, lần truy cập cuối: 23/04/2024.