

TRƯỜNG ĐẠI HỌC SÀI GÒN  
KHOA CÔNG NGHỆ THÔNG TIN



## PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

---

Phát triển ứng dụng Game

# Cờ tướng trên PYTHON

---

GVHD: Từ Lăng Phiêu  
SV: Nguyễn Quốc An - 3120410020  
Lý Kiến Huy - 3120410207  
Lê Minh Phúc - 3120410405  
Email: kienhuy1411@gmail.com

TP. HỒ CHÍ MINH, THÁNG 5/2024

**Bảng Phân Công Công Việc**

Họ và Tên	Nhiệm vụ	Mức độ hoàn thành
Lê Minh Phúc	Thiết kế giao diện, kiểm lỗi	100%
Nguyễn Quốc An	Thiết kế giao diện, xử lý socket	100%
Lý Kiến Huy	Xử lý logic và xử lý thắng thua của game	100%



## Mục lục

<b>1</b>	<b>GIỚI THIỆU ĐỀ TÀI</b>	<b>3</b>
1.1	Mô tả đề tài . . . . .	3
1.2	Ý tưởng thực hiện . . . . .	3
<b>2</b>	<b>CƠ SỞ LÝ THUYẾT</b>	<b>4</b>
2.1	Thư viện Pygame . . . . .	4
2.1.1	Các thành phần chính của Pygame . . . . .	4
2.1.2	Cách cài đặt . . . . .	4
2.2	Thư viện Socket trong Python . . . . .	5
2.2.1	Khái niệm về socket: . . . . .	5
2.2.2	Ưu, nhược điểm của socket: . . . . .	5
2.3	Thư viện Time trong Python . . . . .	6
2.3.1	Các hàm cơ bản . . . . .	6
2.3.2	Ứng dụng của thư viện Time trong Pygame . . . . .	7
2.4	Thư viện Pickle trong Python . . . . .	8
2.4.1	Tuần tự hóa (Serialize) . . . . .	8
2.4.2	Giải tuần tự hóa (Deserialize) . . . . .	8
2.4.3	Sử dụng Pickle trong Pygame . . . . .	8
2.4.4	Ví dụ Kết Hợp Pygame và Pickle . . . . .	9
<b>3</b>	<b>THIẾT KẾ ỨNG DỤNG</b>	<b>10</b>
3.1	Giao diện chính . . . . .	10
3.2	Hình nền . . . . .	11
3.3	Hình ảnh các quân cờ . . . . .	12
<b>4</b>	<b>CÁC TÍNH NĂNG CỦA ỨNG DỤNG</b>	<b>13</b>
4.1	Màn hình chờ . . . . .	13
4.2	Màn hình chính . . . . .	13
4.3	Lựa chọn quân cờ . . . . .	14
4.4	Trò chơi kết thúc . . . . .	14
4.5	Lỗi kết nối . . . . .	15
4.6	Cấu trúc mã nguồn . . . . .	15
<b>5</b>	<b>CÁCH THỨC CÀI ĐẶT TRÒ CHƠI</b>	<b>16</b>



# 1 GIỚI THIỆU ĐỀ TÀI

## 1.1 Mô tả đề tài

Trong trò chơi này, người chơi sẽ ghép ngẫu nhiên người chơi khác để đấu cờ tướng với thời gian được quy định trong 1 trận. Người chơi sẽ tuân thủ các quy tắc của trận đấu cờ tướng như cờ đổ đi trước, quy luật của các quân cờ, tướng,...

## 1.2 Ý tưởng thực hiện

- **Nhập tên người chơi**

Người chơi vào ứng dụng, sau đó điền tên người chơi và bấm enter, tiếp đến sẽ hiển thị màn hình chờ.

- **Đợi đủ người chơi kết nối**

Người chơi đầu tiên kết nối vào ứng dụng sẽ phải đợi nếu chưa có người chơi thứ hai kết nối, còn đã có đủ 2 người chơi thì sẽ vào màn hình trò chơi chính.

- **Vào trận đấu**

Ứng dụng sẽ mặc định người đầu tiên kết nối là quân đỏ người còn lại là quân đen, hiển thị bàn cờ với tên người chơi và đối thủ.

Quân cờ đỏ sẽ đi trước và sẽ bắt đầu tính thời gian đi nước cờ.

- **Cách thức thực hiện nước đi**

Người sẽ chọn quân cờ, sau đó sẽ hiển thị các nước cờ có thể đi (các nước cờ của quân cờ được cờ tướng quốc tế quy định), tiếp đó bấm vào nước cờ muốn đi thì quân cờ sẽ di chuyển đến vị trí mà người chơi mong muốn.

- **Kiểm tra Tướng bị chiếu**

Mỗi nước đi của quân cờ luôn phải kiểm tra xem sau khi đi thì Tướng của phe đó có bị chiếu không. Nếu có, nước đi đó sẽ bị vô hiệu hóa và người chơi phải đi nước cờ khác.

- **Kiểm tra Tướng bị chiếu bí**

Kiểm tra tất cả các nước cờ của quân cờ sau khi đi mà Tướng vẫn còn bị chiếu thì đó là chiếu bí, sau đó trận đấu sẽ kết thúc và hiển thị người chiến thắng.

## 2 CƠ SỞ LÝ THUYẾT

### 2.1 Thư viện Pygame

Pygame là một thư viện mã nguồn mở được sử dụng rộng rãi trong việc phát triển trò chơi và ứng dụng đa phương tiện bằng ngôn ngữ lập trình Python. Dựa trên nền tảng SDL (Simple DirectMedia Layer), Pygame cung cấp các công cụ và tính năng để xử lý âm thanh, đồ họa và sự kiện, giúp người dùng dễ dàng tạo ra các ứng dụng tương tác hấp dẫn.

#### 2.1.1 Các thành phần chính của Pygame

- **Màn hình (Display):** Màn hình là nơi hiển thị mọi thứ trong game. Bạn có thể sử dụng hàm `pygame.display.set-mode()` để tạo ra màn hình game với kích thước tùy chọn.
- **Hình ảnh (Images):** Pygame hỗ trợ nhiều định dạng hình ảnh khác nhau như PNG, JPG. Bạn có thể tải và hiển thị hình ảnh lên màn hình game bằng cách sử dụng hàm `pygame.image.load()` và `pygame.blit()`
- **Sự kiện (Events):** Pygame cung cấp hệ thống sự kiện mạnh mẽ giúp bạn dễ dàng xử lý các sự kiện từ bàn phím, chuột hoặc gamepad. Bạn có thể sử dụng hàm `pygame.event.get()` để lấy các sự kiện và xử lý chúng trong vòng lặp game.
- **Âm thanh (Sounds):** Pygame hỗ trợ phát âm thanh và nhạc trong game. Bạn có thể sử dụng hàm `pygame.mixer.Sound()` để tải và phát các tệp âm thanh.
- **Thời gian (Time):** Pygame cung cấp các hàm để quản lý thời gian và đồng bộ hóa game. Hàm `pygame.time.Clock()` giúp kiểm soát tốc độ khung hình và tạo độ trễ trong game.

#### 2.1.2 Cách cài đặt

Bạn có thể cài đặt Pygame thông qua pip, trình quản lý gói Python tiêu chuẩn. Mở terminal hoặc command prompt và chạy lệnh sau:

```
pip install pygame hoặc pip3 install pygame
```

Lệnh trên sẽ tải và cài đặt phiên bản mới nhất của Pygame từ PyPI (Python Package Index). Sau khi quá trình cài đặt hoàn tất, bạn có thể bắt đầu sử dụng Pygame để phát triển trò chơi và ứng dụng đa phương tiện bằng Python.



## 2.2 Thư viện Socket trong Python

### 2.2.1 Khái niệm về socket:

Socket là giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều (two-way communication) để kết nối 2 process trò chuyện với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket

- Tạo máy chủ:

---

```
1 import socket
2
3 # Định nghĩa host và port mà server sẽ chạy và lắng nghe
4 host = 'localhost'
5 port = 4000
6
7 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8 s.bind((host, port))
9 s.listen(1) # Chỉ chấp nhận 1 kết nối
10
11 print("Server listening on port", port)
12
13 c, addr = s.accept()
14 print("Connected from", str(addr))
15
16 # Gửi dữ liệu tới client
17 c.send(b"Hello, how are you")
18 c.send("Bye".encode())
19
20 c.close()
```

---

- Tạo client:

---

```
1 import socket
2
3 s = socket.socket()
4 s.connect(("localhost", 4000))
5
6 # Nhận dữ liệu từ server
7 data = s.recv(1024)
8 print("Received data:", data.decode())
9
10 s.close()
```

---

### 2.2.2 Ưu, nhược điểm của socket:

- Ưu điểm:

- Tăng tốc độ truyền tải thông tin giữa hai chiều: Socket cho phép giao

tiếp hai chiều giữa các ứng dụng, giúp truyền dữ liệu nhanh chóng và hiệu quả.

**-Dễ phát hiện và xử lý lỗi:** Khi có lỗi xảy ra, socket giúp dễ dàng phát hiện và xử lý. Điều này làm cho việc gỡ lỗi và duy trì ứng dụng dễ dàng hơn.

**-Cách sử dụng đơn giản:** Python cung cấp module socket để tạo và quản lý kết nối socket. Việc sử dụng socket không đòi hỏi cài đặt thêm các phần mềm bổ sung khác.

**-Không cần dùng nhiều phương pháp kết nối khác nhau:** Socket là một cách tiêu chuẩn để giao tiếp giữa các ứng dụng trên mạng, không cần phải sử dụng nhiều phương thức kết nối khác nhau.

- **Nhược điểm:**

**-Phức tạp trong việc xử lý đồng thời (concurrency):** Khi bạn cần xử lý nhiều kết nối cùng lúc, việc quản lý đồng thời trở nên phức tạp. Bạn phải tự viết mã để xử lý nhiều kết nối song song hoặc sử dụng thư viện bên ngoài để giảm thiểu công việc này.

**-Không an toàn trong môi trường không tin cậy:** Socket không cung cấp tính bảo mật cao. Nếu không được cấu hình đúng cách, socket có thể dễ dàng bị tấn công bởi các hacker hoặc mã độc.

**-Khó khăn trong việc kiểm tra và gỡ lỗi:** Khi có lỗi xảy ra trong kết nối socket, việc gỡ lỗi và xác định vấn đề có thể khá phức tạp. Điều này đặc biệt đúng khi bạn phải xử lý nhiều kết nối cùng lúc.

## 2.3 Thư viện Time trong Python

Thư viện Time trong Python cung cấp nhiều hàm hữu ích để làm việc với thời gian. Dưới đây là một số hàm quan trọng:

### 2.3.1 Các hàm cơ bản

- **time.time():** Trả về thời gian hiện tại tính bằng giây kể từ thời điểm epoch (01/01/1970).

```
1 import time
2 current_time = time.time()
3 print("Thời gian hiện tại:", current_time)
```

- **time.sleep():** Tạm dừng chương trình trong một khoảng thời gian tính bằng giây.

```
1 import time
2 print("Bat au dem nguoc:")
3 time.sleep(5)
```



```
4 print("Het thoi gian!")
```

---

- **time.ctime():** Chuyển đổi thời gian tính bằng giây kể từ epoch thành chuỗi thời gian dễ đọc.
- 

```
1 import time
2 current_time = time.time()
3 print("Thoi gian hien tai:", time.ctime(current_time))
```

---

### 2.3.2 Ứng dụng của thư viện Time trong Pygame

Trong Pygame, thư viện Time thường được sử dụng để quản lý tốc độ khung hình và tạo độ trễ trong game. Điều này giúp game chạy mượt mà và ổn định hơn.

- **Quản lý tốc độ khung hình:**
- 

```
1 import pygame
2 import time
3
4 pygame.init()
5 screen = pygame.display.set_mode((800, 600))
6 pygame.display.set_caption("Game with Time")
7
8 clock = pygame.time.Clock()
9 running = True
10 while running:
11     for event in pygame.event.get():
12         if event.type == pygame.QUIT:
13             running = False
14
15     screen.fill((0, 0, 0))
16     pygame.draw.circle(screen, (255, 0, 0), (400, 300), 20)
17     pygame.display.flip()
18     # Giới hạn tốc độ khung hình ở 60 fps
19     clock.tick(60)
20
21 pygame.quit()
```

---

- **Tạo độ trễ trong game:**
- 

```
1 import pygame
2 import time
3
4 pygame.init()
5 screen = pygame.display.set_mode((800, 600))
6 pygame.display.setCaption("Game with Delay")
7
8 running = True
9 while running:
```



```
10     for event in pygame.event.get():
11         if event.type == pygame.QUIT:
12             running = False
13
14     screen.fill((0, 0, 0))
15     pygame.draw.circle(screen, (255, 0, 0), (400, 300), 20)
16     pygame.display.flip()
17
18
19     time.sleep(0.5)
20
21 pygame.quit()
```

---

## 2.4 Thư viện Pickle trong Python

Pickle là một thư viện chuẩn của Python dùng để tuần tự hóa (serialize) và giải tuần tự hóa (deserialize) các đối tượng Python. Pickle giúp bạn lưu trữ các đối tượng thành tập tin và sau đó tải lại chúng vào bộ nhớ.

### 2.4.1 Tuần tự hóa (Serialize)

Để lưu trữ một đối tượng Python vào tệp, bạn có thể sử dụng hàm `pickle.dump()`:

```
1 import pickle
2
3 data = {'key': 'value'}
4 with open('data.pkl', 'wb') as file:
5     pickle.dump(data, file)
```

---

### 2.4.2 Giải tuần tự hóa (Deserialize)

Để tải một đối tượng Python từ tệp, bạn có thể sử dụng hàm `pickle.load()`:

```
1 with open('data.pkl', 'rb') as file:
2     loaded_data = pickle.load(file)
3     print(loaded_data)
```

---

### 2.4.3 Sử dụng Pickle trong Pygame

Bạn có thể sử dụng Pickle để lưu trữ trạng thái của game, điểm số, hoặc các dữ liệu khác mà bạn muốn lưu trữ giữa các phiên làm việc.

- Lưu Trạng Thái Game:

```
1 game_state = {
2     'score': 100,
3     'level': 2,
```

---

```
4     'player_pos': (50, 100)
5 }
6 with open('game_state.pkl', 'wb') as file:
7     pickle.dump(game_state, file)
```

#### • Tải Trạng Thái Game:

```
1 with open('game_state.pkl', 'rb') as file:
2     game_state = pickle.load(file)
3     print(game_state)
```

### 2.4.4 Ví dụ Kết Hợp Pygame và Pickle

Dưới đây là một ví dụ đơn giản kết hợp Pygame và Pickle để lưu và tải trạng thái game:

```
1 import pygame
2 import pickle
3 import time
4
5 pygame.init()
6 screen = pygame.display.set_mode((800, 600))
7 pygame.display.set_caption("Game with Pickle")
8
9 # Trạng thái game ban đầu
10 game_state = {
11     'score': 0,
12     'player_pos': [400, 300]
13 }
14
15 # Tải trạng thái game nếu có
16 try:
17     with open('game_state.pkl', 'rb') as file:
18         game_state = pickle.load(file)
19 except FileNotFoundError:
20     pass
21
22 player_pos = game_state['player_pos']
23 score = game_state['score']
24
25 running = True
26 while running:
27     for event in pygame.event.get():
28         if event.type == pygame.QUIT:
29             running = False
30
31     keys = pygame.key.get_pressed()
32     if keys[pygame.K_LEFT]:
33         player_pos[0] -= 5
34     if keys[pygame.K_RIGHT]:
```

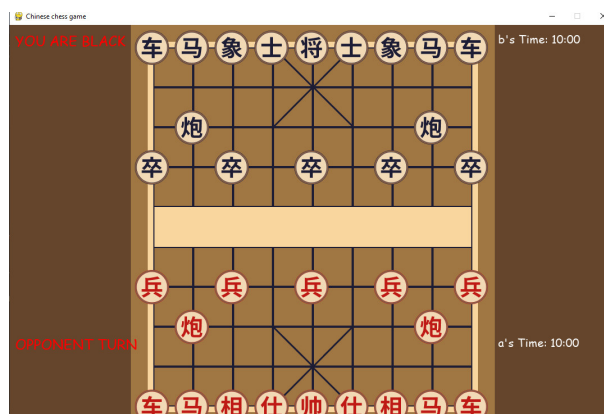
```
35     player_pos[0] += 5
36     if keys[pygame.K_UP]:
37         player_pos[1] -= 5
38     if keys[pygame.K_DOWN]:
39         player_pos[1] += 5
40
41     screen.fill((0, 0, 0))
42     pygame.draw.circle(screen, (255, 0, 0), player_pos, 20)
43     pygame.display.flip()
44     time.sleep(0.01)
45
46     # Lưu trạng thái game khi thoát
47     game_state['player_pos'] = player_pos
48     game_state['score'] = score
49     with open('game_state.pkl', 'wb') as file:
50         pickle.dump(game_state, file)
51
52     pygame.quit()
```

Trong ví dụ trên, trạng thái game bao gồm vị trí của người chơi và điểm số sẽ được lưu lại vào tệp khi game thoát và sẽ được tải lại khi game khởi động lại. Thư viện `time` được sử dụng để làm chậm vòng lặp game một chút, giúp game chạy mượt mà hơn.

## 3 THIẾT KẾ ỨNG DỤNG

### 3.1 Giao diện chính

Giao diện chính của trò chơi có thể được thiết kế với một lưới ô vuông để hiển thị bản đồ và các đối tượng trong trò chơi. Giao diện có thể sử dụng giao diện đồ họa 2D để hiển thị các đối tượng trong môi trường trò chơi.

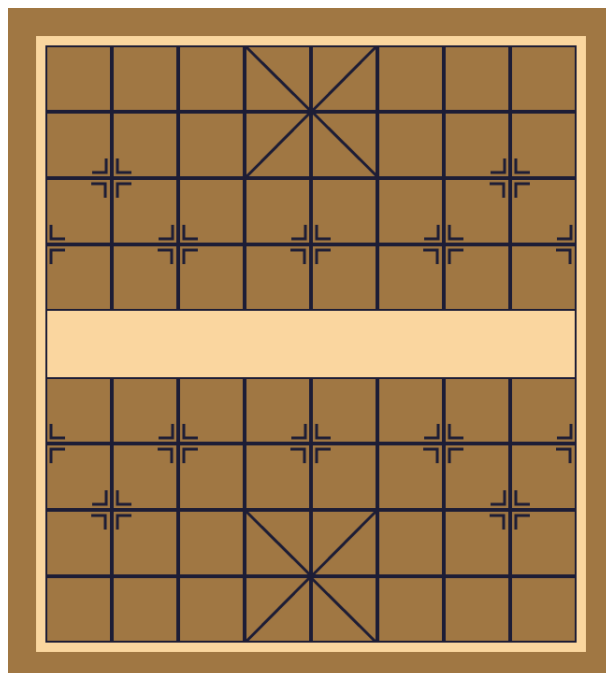


Hình 1: Hình nền của trò chơi

### 3.2 Hình nền



Hình 2: Hình nền của trò chơi



Hình 3: Bàn cờ của trò chơi

### 3.3 Hình ảnh các quân cờ



Hình 4: Quân cờ đỏ



Hình 5: Quân cờ đen

## 4 CÁC TÍNH NĂNG CỦA ỨNG DỤNG

### 4.1 Màn hình chờ

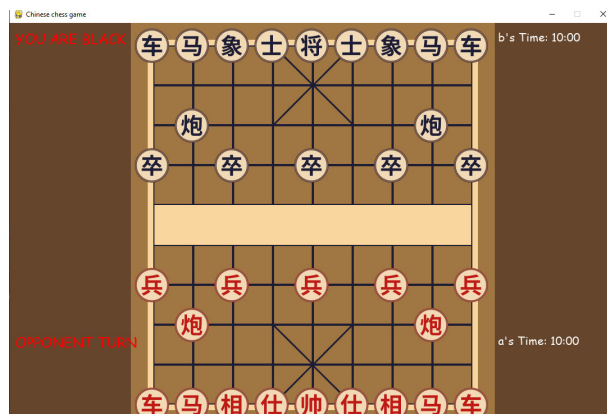
Khi người chơi đã kết nối vào trò chơi nhưng chưa đủ số lượng người tham gia trò chơi thì sẽ được vào màn hình chờ.



Hình 6: Màn hình chờ của trò chơi

### 4.2 Màn hình chính

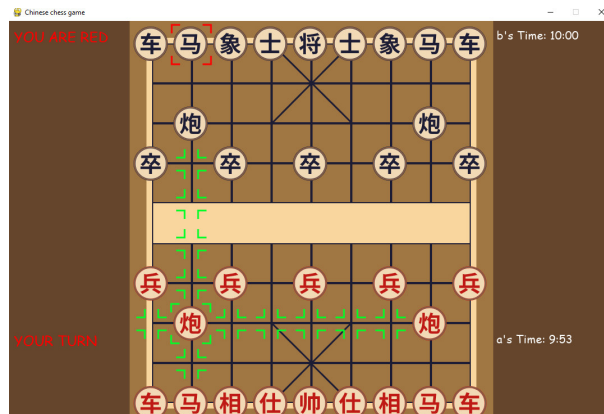
Khi đã đủ người chơi kết nối vào trò chơi thì trò chơi sẽ được bắt đầu.



Hình 7: Màn hình chính của trò chơi

### 4.3 Lựa chọn quân cờ

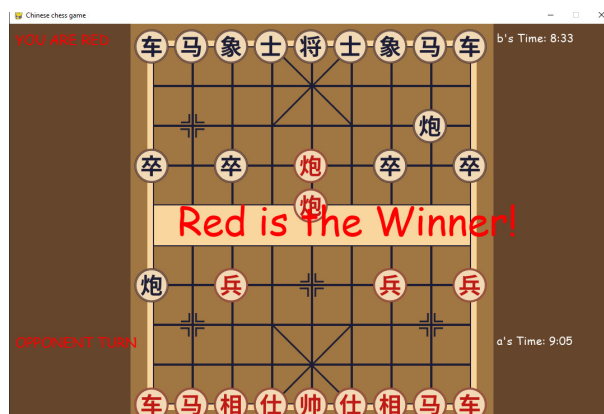
Khi người chơi nhấn chọn quân cờ sẽ hiện ra các đường đi của quân cờ có thể đi và hiện ra đối thủ mà quân cờ được chọn có thể ăn.



Hình 8: Quân cờ được chọn

### 4.4 Trò chơi kết thúc

Khi người chơi đã chiếu bí đối thủ thì màn hình sẽ hiện ra người chơi màu nào chiến thắng và kết thúc trò chơi.



Hình 9: người chơi chiến thắng

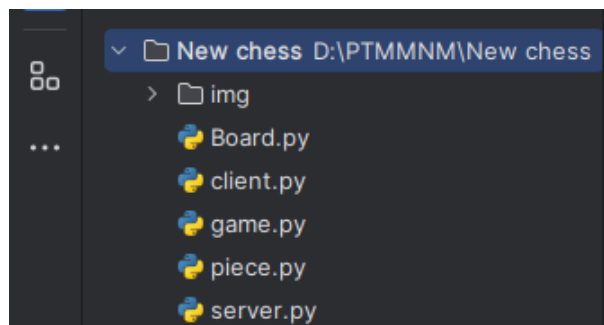
#### 4.5 Lỗi kết nối

Thông báo cho người chơi server đã bị lỗi và người chơi cần kết nối lại để chơi.



Hình 10: Lỗi kết nối

#### 4.6 Cấu trúc mã nguồn



Hình 11: Cấu trúc mã nguồn

Thư mục img dùng để chứa hình ảnh của trò chơi

File Board.py dùng để lưu thông tin bàn cờ và di chuyển các nước đi của quân cờ

File client.py dùng để kết nối với máy chủ

File piece.py dùng để lưu thông tin quân cờ và các nước đi của quân cờ

File server.py dùng để lắng nghe các kết nối từ client, nhận tin nhắn từ client (nếu có), và gửi trả lời về lại client.



## 5 CÁCH THỨC CÀI ĐẶT TRÒ CHƠI

### Cách thức cài đặt

Để cài đặt trò chơi, bạn cần thực hiện các bước sau:

1. **Cài đặt Python:** Đầu tiên, bạn cần cài đặt Python trên máy tính của mình. Python là ngôn ngữ lập trình chính mà bạn sẽ sử dụng để phát triển trò chơi. Bạn có thể tải Python từ trang web chính thức của Python (<https://www.python.org/downloads/>). Sau đó, làm theo hướng dẫn trên màn hình để hoàn thành quá trình cài đặt.
2. **Cài đặt thư viện:** Sau khi cài đặt Python, bạn cần cài đặt thư viện và công cụ cần thiết để phát triển trò chơi.
  - **Pygame:** Pygame là một thư viện phổ biến để phát triển trò chơi trong Python. Bạn có thể cài đặt Pygame bằng cách sử dụng pip, một trình quản lý gói cho Python:

```
1 pip install pygame
```
  - **Các thư viện khác:** Tùy thuộc vào yêu cầu của trò chơi, bạn có thể cài đặt các thư viện khác như Pyglet, Panda3D, hoặc Arcade.
3. **Bắt đầu phát triển trò chơi:** Sau khi đã cài đặt Python và các thư viện cần thiết, bạn đã sẵn sàng bắt đầu phát triển trò chơi của mình. Hãy tạo một tập tin Python mới và bắt đầu viết mã nguồn cho trò chơi.

Để trò chơi hoạt động, bạn cần thực hiện các bước sau:

### Chạy server

Đầu tiên, bạn cần chạy server bằng cách thực hiện lệnh sau trong terminal:

```
1 python Server.py
```

Bước này sẽ khởi động server để trò chơi có thể chạy và giao tiếp với client.

### Chạy client

Đầu tiên, bạn cần chạy client bằng cách thực hiện lệnh sau trong terminal:

```
1 python clietnt.py
```

Khi người chơi vào phòng thành công, khi đó client và server kết nối thành công và bắt đầu trò chơi



## Tài liệu

- [1] <https://viblo.asia/p/lap-trinh-socket-bang-python-jvEla084Zkw>
- [2] <https://zigavn.com/co-tuong/huong-dan-luat-choi>
- [3] ChatGPT “link: <https://chat.openai.com/>”