

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Xây dựng ứng dụng game

Little Commander trên PYTHON

GVHD: Từ Lăng Phiêu
SV: Lâm Kha Tuấn - 3120410584
Phạm Trần Mỹ Duyên - 3120410099

TP. HỒ CHÍ MINH, THÁNG 5/2024

Mục lục

1	Phần giới thiệu	2
1.1	Các thư viện cơ bản	2
1.2	Giới thiệu dự án	2
2	CẤU TRÚC THƯ MỤC VÀ NGUYÊN LÝ	3
2.1	Cấu trúc thư mục	3
2.2	Nguyên lý hoạt động	4
3	HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG	10
3.1	Cài đặt	10
3.2	Hướng dẫn cách chơi	10
3.3	Giao diện của ứng dụng	10
4	TÀI LIỆU THAM KHẢO	13

1 Phần giới thiệu

1.1 Các thư viện cơ bản

Python 3.0 là một ngôn ngữ lập trình thông dịch, có cú pháp đơn giản và dễ đọc, một ngôn ngữ mã nguồn mở, cho phép mọi người tham gia phát triển và cải thiện ngôn ngữ thông qua các đóng góp vào mã nguồn.

Pygame



Pygame là một thư viện Python được sử dụng để phát triển trò chơi và ứng dụng đa phương tiện. Được tạo ra dựa trên SDL (Simple DirectMedia Layer), Pygame cung cấp một giao diện lập trình ứng dụng (API) đơn giản để tạo ra các ứng dụng đa phương tiện, đặc biệt là trò chơi. Một số ưu điểm của pygame:

- Pygame được thiết kế để dễ dàng học và sử dụng, đặc biệt là cho các người mới bắt đầu trong lập trình trò chơi.
- Pygame hoạt động trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux, giúp bạn phát triển trò chơi và ứng dụng đa nền tảng một cách dễ dàng.
- Cung cấp các công cụ và thư viện cho việc vẽ đồ họa, xử lý hình ảnh, văn bản, âm thanh và âm nhạc.

Socket trong Python là một thư viện mạnh mẽ và linh hoạt cho phép các chương trình giao tiếp qua mạng máy tính. Nó cung cấp các phương thức để tạo và sử dụng socket, qua đó các chương trình có thể gửi và nhận dữ liệu qua mạng. Thư viện này hỗ trợ cả giao thức TCP (Transmission Control Protocol) và UDP (User Datagram Protocol), cho phép lập trình viên xây dựng các ứng dụng mạng đa dạng như máy chủ web, ứng dụng chat, trò chơi trực tuyến, và nhiều ứng dụng khác.

1.2 Giới thiệu dự án

Hiện nay, ngành phát triển trò chơi điện tử đang phát triển nhanh, trở nên xu hướng trong xã hội. Các trò chơi điện tử đang đem lại một doanh thu lớn cho các công ty, cá nhân.

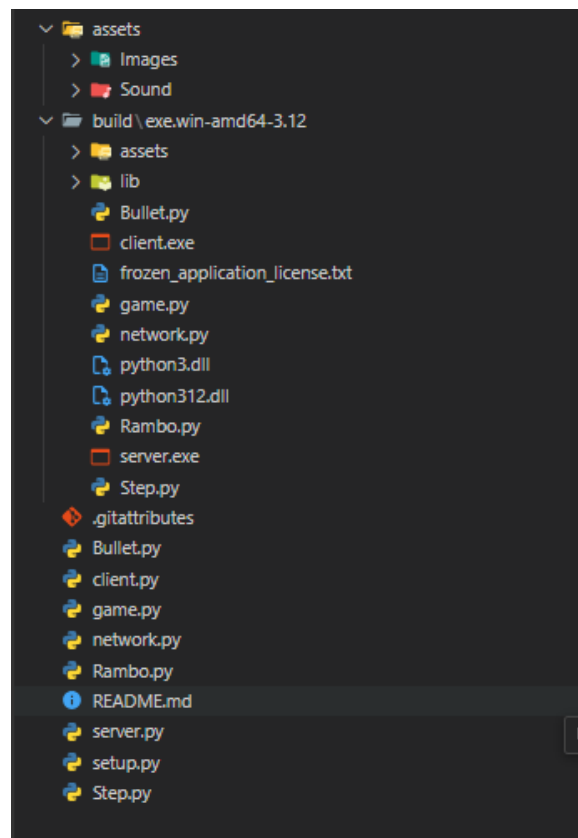
Lĩnh vực phát triển trò chơi điện tử đang tạo ra nhiều trò chơi hay, hấp dẫn. Chúng giúp cho người chơi giải trí sau những giờ làm việc căng thẳng, tạo động lực cho lần làm việc tiếp theo. Trong tình hình đó, nhóm em quyết định xây dựng trò chơi bắn súng 2 người chơi. Trò chơi được viết bằng ngôn ngữ Python và chủ yếu sử dụng thư viện Pygame. Trò chơi có cách chơi đơn giản và có thể kết nối chơi 2 người trực tiếp với nhau thông qua kết nối cùng 1 mạng LAN.

2 CẤU TRÚC THƯ MỤC VÀ NGUYÊN LÝ

2.1 Cấu trúc thư mục

Cấu trúc của thư mục bao gồm:

- Thư mục assets để chứa các tài nguyên để xây dựng game như hình ảnh, âm thanh, ...
- Thư mục build để người dùng có thể vào game bằng file exe.
- File server.py để tạo ra một kết nối socket giữa 2 người dùng trong một mạng LAN.
- File client.py để người dùng tiến hành đăng ký vào server để chơi.
- File network.py chứa các phương thức để nhận và gửi dữ liệu thông qua server.
- File game.py chứa các thông tin của 2 người chơi để gửi vào server.
- Các file Bullet, Step, Rambo dùng để khởi tạo nhân vật trong trò chơi.



Hình 1: Cấu trúc thư mục dự án

2.2 Nguyên lý hoạt động

Server Side

- Khởi tạo và cấu hình cho socket.

```
1 server = "localhost"
2 port = 5555
3
4 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5
6 try:
7     s.bind((server, port))
8 except socket.error as e:
9     str(e)
10
11 s.listen(2)
12 print("Waiting for a connection, Server Started")
13 drawWindow()
14
15 connected = set()
16 games = {}
17 idCount = 0
```

Code minh họa

- Xử lý khi client thực hiện kết nối, hoặc muốn gửi yêu cầu lên server

```
1 def threaded_client(conn, p, gameId):
2     global idCount
3     conn.send(str.encode(str(p)))
4     while True:
5         try:
6             data = conn.recv(4096).decode("UTF-8")
7             if gameId in games:
8                 game = games[gameId]
9                 if not data:
10                     break
11                 else:
12                     if data != "get":
13                         if data == "shoot":
14                             game.createBullet(p)
15                         elif data == "onStep":
16                             game.players[p].onStep = True
17                         elif data == "not_onStep":
18                             if p == 0:
19                                 game.players[0].onStep = False
20                                 game.players[0].fallDone = False
21                             if p == 1:
22                                 game.players[1].onStep = False
23                                 game.players[1].fallDone = False
```

```
24         elif data.startswith("bi_ban_boi_dan_"):
25             id_dan = int(data.replace("bi_ban_boi_dan_", ""))
26             player_biban = game.players[p]
27             for i in range(0, len(game.players)):
28                 for bullet in game.players[i].bullets:
29                     if bullet.id == id_dan:
30                         game.players[i].bullets.remove(bullet)
31                         break
32             player_biban.hp -= 20
33             player_biban.hurting = True
34             for i in range(0, len(game.players)):
35                 if game.players[i].hp <= 0:
36                     if i == 0:
37                         game.playerWin = 1
38                     elif i == 1:
39                         game.playerWin = 0
40             elif data == "reset":
41                 game.reset()
42             else:
43                 # Cp nht di chuyn ca ngi chi
44                 game.play(p, data)
45             game.update()
46             game.capNhatDan()
47             conn.sendall(pickle.dumps(game))
48         else:
49             break
50     except socket.error as e:
51         print(e)
52         break
53 print("Lost connection")
54 try:
55     del games[gameId]
56     print("Closing Game", gameId)
57 except:
58     pass
59 idCount -= 1
60 conn.close()
```

Code minh họa

- Chấp nhận tạo kết nối mới, kiểm tra xem nếu đã đủ 2 người chơi thì sẽ tạo 1 game mới nếu không thì sẽ đưa người chơi vào trạng thái chờ.

```
1 is_running = True
2 is_shutdown = threading.Event()
3
4 def accept_connections(is_shutdown):
5     global idCount
6     s.settimeout(0.5)
7     while True:
8         if is_shutdown.is_set():
```



```
9         break
10     try:
11         conn, addr = s.accept()
12         print("Connected to:", addr)
13
14         idCount += 1
15         p = 0
16         gameId = (idCount - 1) // 2
17         if idCount % 2 == 1:
18             games[gameId] = Game(gameId)
19             print("Creating a new game with id... " + str(gameId))
20         else:
21             games[gameId].ready = True
22             p = 1
23         threading.Thread(target=threaded_client, args=(conn, p, gameId)).start()
24     except:
25         pass
26
27 accept_thread = threading.Thread(target=accept_connections, args=(is_shutdown,))
28 accept_thread.start()
29
30 while is_running:
31     for event in pygame.event.get():
32         if event.type == pygame.QUIT:
33             is_running = False
34             is_shutdown.set()
35
36 print("Ready to join")
37 accept_thread.join()
38 pygame.quit()
39 s.close()
40 sys.exit()
```

Code minh họa

- Network chứa các phương thức để giao tiếp, gửi và nhận dữ liệu giữa client và server

```
1 import pickle
2 import socket
3
4
5 class Network:
6     def __init__(self):
7         self.client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8         # self.client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
9         self.server = "localhost"
10        self.port = 3633
11        self.addr = (self.server, self.port)
12        self.post = self.connect()
13
14    def getPost(self):
```

```
15     return self.post
16
17 def connect(self):
18     try:
19         self.client.connect(self.addr)
20         print("Server Connected")
21         return self.client.recv(4094 * 2).decode()
22     except:
23         pass
24
25 def send(self, data):
26     try:
27         self.client.send(str.encode(data))
28         # print("DA GUI SEND")
29         pick = pickle.loads(self.client.recv(2048 * 2))
30         # print("DA NHAN SEND")
31         return pick
32     except socket.error as e:
33         print("NETWORK Error")
34         print(e)
```

Client Side

- Nhận ID người chơi từ server và liên tục nhận cập nhật game từ server. Đồng thời có sẽ gửi các yêu cầu, dữ liệu của người chơi lên server.

```
1 def main():
2     run = True
3     clock = pygame.time.Clock()
4     n = Network()
5
6     player = int(n.getPost())
7     print("You are player", player)
8
9     while run:
10        clock.tick(FPS)
11        try:
12            game = n.send("get")
13        except:
14            print("Couldn't get game")
15            break
16        if game.playerWin != -1:
17            font = pygame.font.SysFont("comicsans", 60)
18            win_text = font.render("You Win", 1, WHITE)
19            lose_text = font.render("You Lose", 1, WHITE)
20            if game.playerWin == player:
21                win.blit(win_text, (100, 350))
22            else:
23                win.blit(lose_text, (100, 350))
24            pygame.display.update()
25            pygame.time.delay(4000)
```



```
26         return
27
28     action = ''
29     keys = pygame.key.get_pressed()
30     if keys[pygame.K_LEFT]:
31         action = "left"
32     elif keys[pygame.K_RIGHT]:
33         action = "right"
34
35     for event in pygame.event.get():
36         if event.type == pygame.QUIT:
37             run = False
38             pygame.quit()
39         if event.type == pygame.KEYUP:
40             if event.key == pygame.K_LEFT:
41                 action = "un_left"
42             elif event.key == pygame.K_RIGHT:
43                 action = "un_right"
44             elif event.key == pygame.K_UP:
45                 action = "un_jump"
46         if event.type == pygame.KEYDOWN:
47             if event.key == pygame.K_UP:
48                 if keys[pygame.K_LEFT]:
49                     action = "left_jump"
50                     jump_sound.play()
51                     print("nhay sang trai")
52
53                 elif keys[pygame.K_RIGHT]:
54                     action = "right_jump"
55                     jump_sound.play()
56                     print("nhay sang phai")
57             else:
58                 action = "jump"
59                 jump_sound.play()
60             if event.key == pygame.K_s:
61                 action = "shoot"
62                 current_player = game.players[player]
63                 if len(current_player.bullets) < 2:
64                     shoot_sound.play()
65                 else:
66                     reload_amor.play()
67             if action != "" and game.connected():
68                 n.send(action)
69             redrawWindow(win, game, player, n)
70
71
72 def menu_screen():
73     run = True
74     clock = pygame.time.Clock()
75     while run:
76         clock.tick(FPS)
77         win.blit(background, (0, 0))
```



```
78
79     font = pygame.font.SysFont("comicsans", 60)
80     text = font.render("Click to Play!", 1, (255, 0, 0))
81     win.blit(text, (100, 200))
82     pygame.display.update()
83
84     for event in pygame.event.get():
85         if event.type == pygame.QUIT:
86             pygame.quit()
87             run = False
88         if event.type == pygame.MOUSEBUTTONDOWN:
89             run = False
90
91     main()
92
93
94 while True:
95     menu_screen()
```

Hình : Code File client.py

3 HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG

3.1 Cài đặt


Thực hiện các bước sau:

- Clone ứng dụng game từ đường link [Little Commander](https://github.com/khatuanlam/Little_Commander) bằng cú pháp :

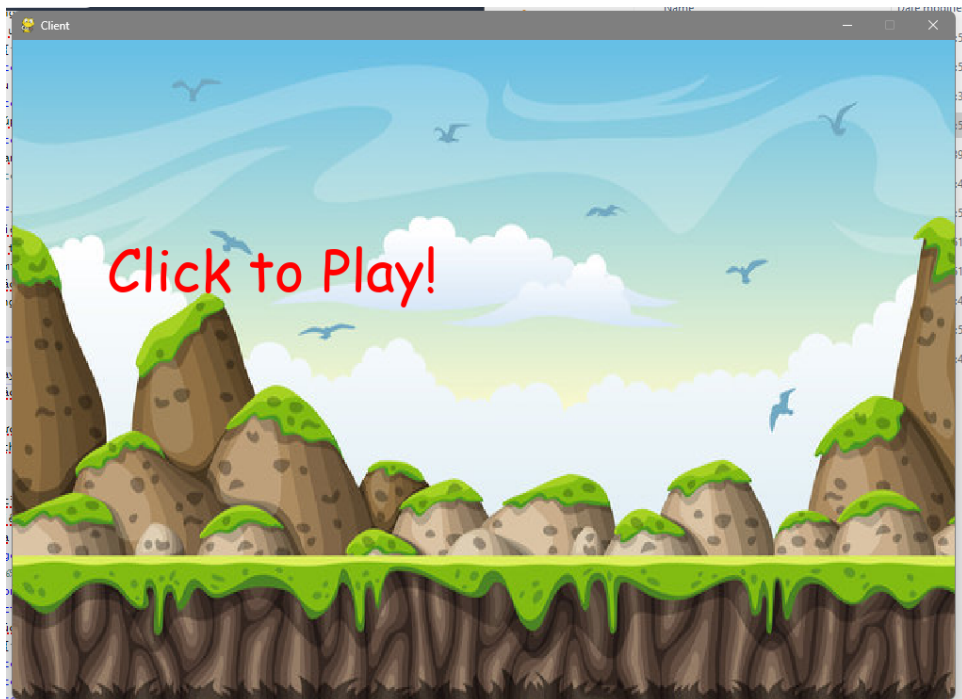
```
1 git clone https://github.com/khatuanlam/Little_Commander.git
```

- Run file server.exe trong thư mục Build.
- Sau đó run file client.exe để tiến hành vào game.

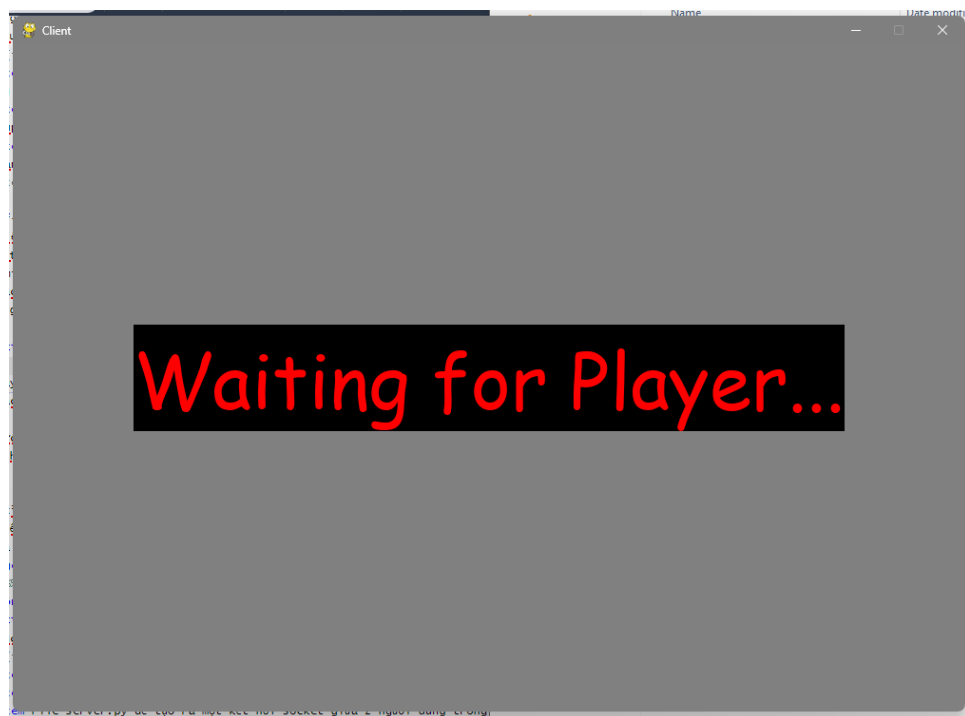
3.2 Hướng dẫn cách chơi

- Dùng các phím  để di chuyển nhân vật.
- Phím S để thực hiện bắn súng.
- Phím Space để nhảy.
- Đến khi đối thủ hết HP thì bạn sẽ chiến thắng.

3.3 Giao diện của ứng dụng



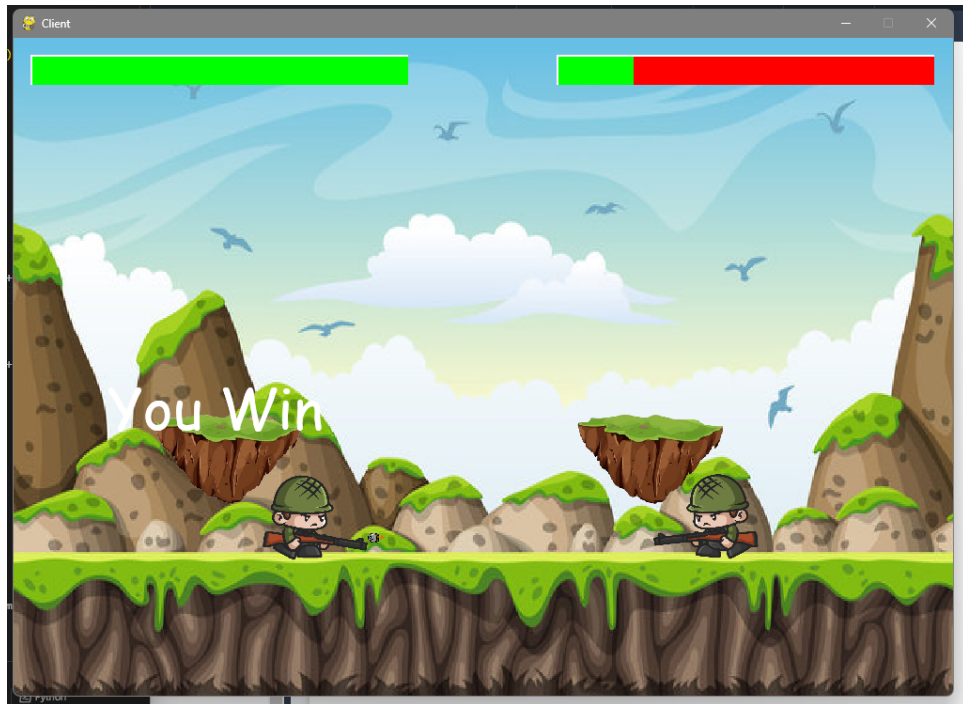
Giao diện khi bắt đầu



Giao diện khi chờ người chơi



Giao diện khi game bắt đầu



Giao diện khi chiến thắng



4 TÀI LIỆU THAM KHẢO

Tài liệu

- [1] Advanced Guide to Python 3 Programming “[Advanced Guide to Python 3 Programming](#)”, *John Hunt*.
- [2] Socket Game Source Code “[Online Rock-Paper-Scissors Game](#)”,
- [3] Python Socket Document “[Python Socket](#)”,
- [4] Python Game using Pygame “[Buiding Game Tutorial](#)”,