

TRƯỜNG ĐẠI HỌC SÀI GÒN  
KHOA CÔNG NGHỆ THÔNG TIN



## PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

---

Ứng dụng game

# Rock Paper Scissors trên PYTHON

---

GVHD: Từ Lăng Phiêu

Nhóm 29

SV: Huỳnh Minh Quân - 312410433

Thiều Anh Sang - 3120410449

Huỳnh Đức Tâm - 3120410459

Nguyễn Bảo Tân - 3120410464

TP. HỒ CHÍ MINH, THÁNG 5/2024

# Mục lục

<b>1</b>	<b>Phần giới thiệu</b>	<b>2</b>
1.1	Giới thiệu game Rock Paper Scissors . . . . .	2
1.2	Lợi ích của lập trình game với Python . . . . .	2
<b>2</b>	<b>Cơ sở lý thuyết</b>	<b>4</b>
2.1	Phần mềm mã nguồn mở . . . . .	4
2.2	Python . . . . .	4
2.3	IDE Pycharm . . . . .	5
2.4	Thư viện hỗ trợ . . . . .	5
<b>3</b>	<b>Xây dựng game Rock Paper Scissors</b>	<b>7</b>
3.1	Mô tả giao diện ứng dụng game . . . . .	7
3.2	Code chức năng cho game . . . . .	8
3.3	Cài đặt và thiết lập môi trường . . . . .	16

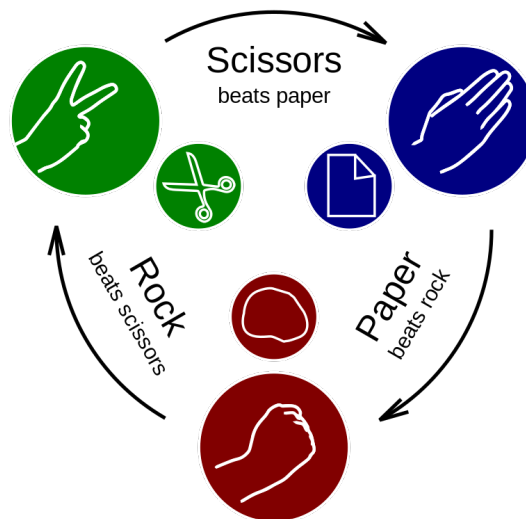
# 1 Phân giới thiệu

## 1.1 Giới thiệu game Rock Paper Scissors

"Rock Paper Scissors" là một ứng dụng game giải trí đơn giản nhưng vô cùng thú vị, dựa trên trò chơi truyền thống cùng tên hay được gọi là Oẳn Tù Tì, hoàn toàn chiến thắng dựa trên sự may mắn. Trong trò chơi này, người chơi chọn một trong ba lựa chọn: "Rock" (đá), "Paper" (giấy), hoặc "Scissors" (kéo), và cố gắng chiến thắng bằng cách chọn lựa tốt hơn so với lựa chọn của đối thủ.

**Luật chơi đá giấy kéo (Rock Paper Scissors) rất đơn giản:**

- Cần có 2 người tham gia trò chơi
- Mỗi người chơi bí mật chọn một trong ba cử chỉ: Đá, Giấy hoặc Kéo.
- Đá thắng Kéo, Kéo thắng Giấy, Giấy thắng Đá.
- Nếu hai người chơi chọn cùng một cử chỉ, sẽ là hòa và chơi lại.



Hình 1: Hình minh họa rock paper scissors

### Mục tiêu xây dựng game

Ứng dụng game được phát triển nhằm mục đích nghiên cứu về hệ điều hành mã nguồn mở và cách sử dụng socket trong ngôn ngữ python. Bằng việc xây dựng game Rock Paper Scissors - một trò chơi quen thuộc và đơn giản nhằm minh họa để hiểu rõ hơn môn học, mặt khác giúp người dùng dễ làm quen, tạo trải nghiệm thú vị khi tương tác với những người chơi khác.

## 1.2 Lợi ích của lập trình game với Python

Game được phát triển bằng Python mang lại nhiều lợi ích khác nhau, không chỉ cho những người tham gia phát triển mà còn cho người chơi. Dưới đây là một số lợi ích chính của việc phát triển game bằng Python:

- Dễ học và sử dụng: Python là một ngôn ngữ lập trình dễ học và sử dụng, đặc biệt là cho người mới bắt đầu. Cú pháp đơn giản và ngắn gọn của Python làm cho việc phát triển game trở nên dễ dàng hơn.



- Phong phú về thư viện và framework: Python có nhiều thư viện và framework mạnh mẽ như Pygame, Panda3D, Pyglet, và Kivy, giúp nhà phát triển tạo ra các trò chơi đa dạng từ game 2D đơn giản đến game 3D phức tạp.
- Đa nền tảng: Game được phát triển bằng Python có thể chạy trên nhiều nền tảng khác nhau bao gồm Windows, macOS và Linux, giúp mở rộng đối tượng người chơi tiềm năng.
- Cộng đồng lớn và hỗ trợ mạnh mẽ: Python có một cộng đồng phát triển lớn và mạnh mẽ, nơi bạn có thể tìm kiếm thông tin, hướng dẫn và hỗ trợ từ các nhà phát triển khác trên toàn thế giới.
- Tích hợp dễ dàng với công nghệ khác: Python có khả năng tích hợp tốt với các công nghệ khác như AI (Trí tuệ nhân tạo), Machine Learning (Học máy), và các API (Giao diện lập trình ứng dụng), mở ra nhiều cơ hội mở rộng cho trò chơi.
- Tăng cường kỹ năng lập trình: Phát triển game bằng Python không chỉ giúp bạn tạo ra sản phẩm giải trí mà còn giúp bạn cải thiện kỹ năng lập trình của mình, từ việc xử lý dữ liệu đến thiết kế giao diện người dùng.

## 2 Cơ sở lý thuyết

### 2.1 Phần mềm mã nguồn mở

Phần mềm mã nguồn mở là phần mềm với mã nguồn được công bố, với một giấy phép sử dụng đi kèm, cho phép bất cứ ai cũng có thể sử dụng, nghiên cứu, thay đổi cải tiến phần mềm và phân phối phần mềm ở dạng chưa thay đổi hoặc đã thay đổi

Một số lợi ích của việc phát triển phần mềm mã nguồn mở:

- Sử dụng miễn phí

Điều đầu tiên mà có lẽ ai khi biết về mã nguồn mở có thể sử dụng cho cả mục đích cá nhân lẫn thương mại mà không phải chi trả chi phí bản quyền. Điều này khiến mọi người có thể tùy biến được hầu hết các chức năng của mã nguồn mà không bị giới hạn hay trả thêm chi phí như những phần mềm mã nguồn độc quyền khác.

- Khả năng bảo mật cao

Mặc dù là mã nguồn miễn phí nhưng hầu hết các mã nguồn mở đều có khả năng bảo mật tuyệt vời. Lý do vì mã nguồn mở ngay từ đầu được xây dựng và đóng góp bởi cộng đồng, trong đó có vô số những nhà lập trình thiên tài. Nghĩa là khi có bất kỳ vấn đề nào xảy ra, chúng ngay lập tức được sửa đổi, khắc phục... điều này thể hiện rõ nhất tinh thần cộng đồng của mã nguồn mở.

- Can thiệp sâu vào vấn đề quản trị và điều chỉnh

Khi làm việc trên mã nguồn mở bạn có thể toàn quyền truy cập, quản trị cũng như điều chỉnh cấu trúc để mã nguồn có thể phù hợp với nhu cầu của bản thân. Điều này cũng kích thích một số nhà lập trình từ việc sử dụng mã nguồn mở này để tạo ra nhiều phần mềm hữu ích hơn cho cộng đồng.

- Tính ổn định của mã nguồn mở

Điều này được thể hiện căn bản nhất là vô số những website sử dụng mã nguồn mở đến hiện tại vẫn duy trì được khả năng ổn định trong vận hành liên tục mà không mắc phải bất kỳ vấn đề nào. Nguyên nhân chính có lẽ xuất phát từ việc mã nguồn mở được xây dựng dựa trên nguyên tắc tối ưu cho cộng đồng ai cũng có thể sử dụng nên về bản chất sẽ duy trì được tính ổn định trong vận hành hơn so với một số mã nguồn đóng.

### 2.2 Python

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

Những lợi ích của Python bao gồm:

- Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
- Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.
- Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.



- Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
- Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
- Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.
- Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.

## 2.3 IDE Pycharm

PyCharm là một Môi trường phát triển tích hợp (IDE) được phát triển bởi JetBrains, chuyên dụng cho lập trình Python. Đây là công cụ được đánh giá cao và sử dụng rộng rãi bởi các lập trình viên Python ở mọi trình độ, từ mới bắt đầu đến chuyên nghiệp. Điểm nổi bật của PyCharm:

- Hỗ trợ mã hóa thông minh: PyCharm cung cấp nhiều tính năng hỗ trợ mã hóa thông minh như tự động hoàn thành code, kiểm tra lỗi cú pháp, refactor code, v.v., giúp lập trình viên viết code nhanh chóng, chính xác và hiệu quả hơn.
- Điều hướng dự án dễ dàng: PyCharm giúp bạn dễ dàng điều hướng trong dự án Python của mình với các tính năng như tìm kiếm nhanh, xem cấu trúc thư mục, chuyển đổi nhanh giữa các tệp tin, v.v.
- Giải thích và gỡ lỗi hiệu quả: PyCharm cung cấp các công cụ mạnh mẽ để giải thích và gỡ lỗi code Python, giúp bạn nhanh chóng xác định và sửa lỗi trong chương trình của mình.
- Hỗ trợ phát triển web: PyCharm hỗ trợ phát triển web Python với các framework phổ biến như Django và Flask.
- Tích hợp nhiều công cụ: PyCharm tích hợp nhiều công cụ khác nhau như Git, SVN, Docker, v.v., giúp bạn quản lý dự án và phát triển hiệu quả hơn.
- Giao diện tùy chỉnh: PyCharm có giao diện người dùng trực quan và có thể tùy chỉnh cao, giúp bạn tạo môi trường làm việc phù hợp với sở thích của mình.

## 2.4 Thư viện hỗ trợ

### Thư viện Pygame

Pygame là một thư viện mã nguồn mở cho phép lập trình game và đa phương tiện trên Python. Thư viện này cung cấp các công cụ cho việc tạo các ứng dụng đa phương tiện, chẳng hạn như các trò chơi, âm nhạc, đồ họa và các ứng dụng tương tự.

Ưu điểm của Pygame:

- Dễ học: Pygame có cú pháp đơn giản và dễ hiểu, cho phép người dùng mới bắt đầu nhanh chóng học được cách sử dụng thư viện.
- Đa nền tảng: Pygame có thể hoạt động trên nhiều nền tảng khác nhau, bao gồm Windows, Mac OS X, Linux và nhiều hệ điều hành khác.



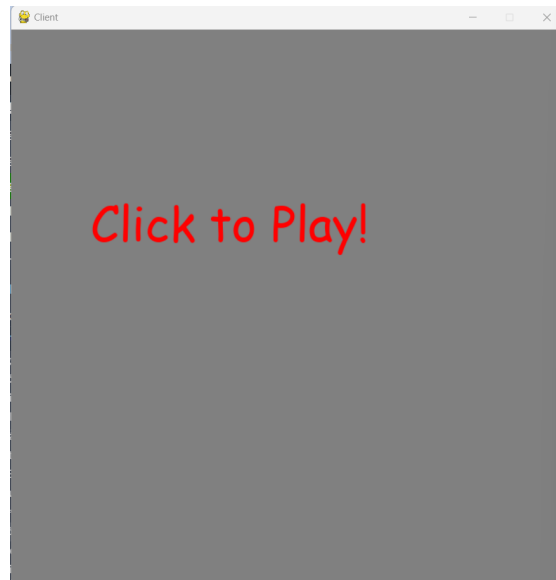
- Đa dạng: Pygame có nhiều tính năng và hỗ trợ đầy đủ cho các yêu cầu phức tạp trong lập trình game và đa phương tiện.
- Cộng đồng lớn: Pygame có một cộng đồng lớn với nhiều tài liệu hướng dẫn và các ví dụ, giúp người dùng tìm hiểu và giải quyết các vấn đề.

Nhược điểm của Pygame:

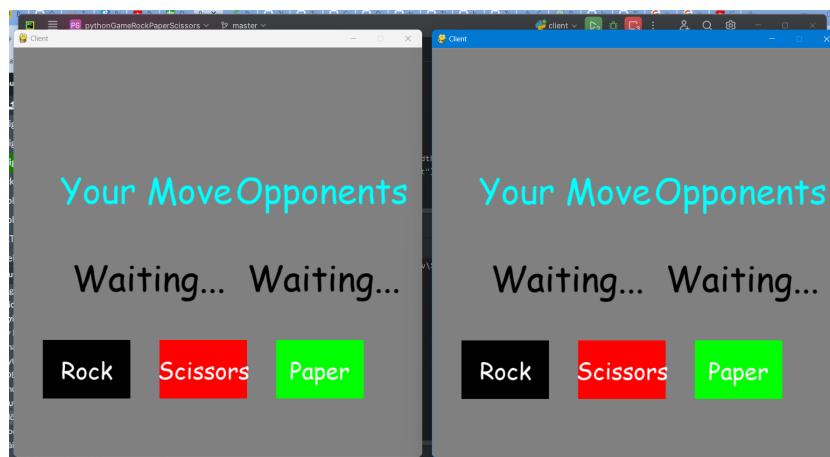
- Khả năng tối ưu hóa không tốt: Pygame không được tối ưu hóa tốt cho các ứng dụng có đồ họa phức tạp, do đó, nếu không được xử lý tốt, có thể dẫn đến giảm hiệu suất.
- Hạn chế đồ họa: Pygame không hỗ trợ các tính năng đồ họa cao cấp như DirectX hay OpenGL, vì vậy không thể sử dụng để tạo ra các trò chơi và ứng dụng đồ họa cao cấp.
- Không được bảo trì thường xuyên: Pygame không được bảo trì và cập nhật thường xuyên, do đó, có thể không hoạt động tốt với các phiên bản Python mới nhất.
- Không hỗ trợ 3D: Pygame không hỗ trợ lập trình 3D, do đó không phù hợp để tạo ra các trò chơi và ứng dụng có đồ họa 3D phức tạp.

### 3 Xây dựng game Rock Paper Scissors

#### 3.1 Mô tả giao diện ứng dụng game

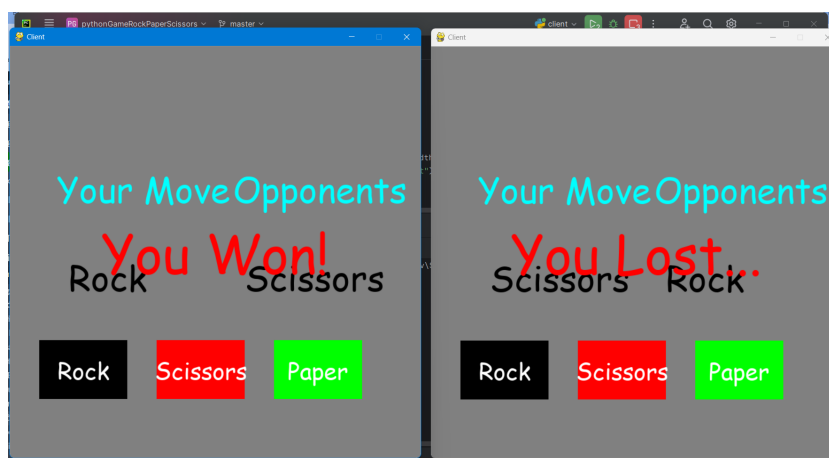


Hình 2: Giao diện chương trình khi khởi chạy



Hình 3: Giao diện người chơi





Hình 4: Kết quả sau khi chơi

## 3.2 Code chức năng cho game

### Socket trong python

Socket là một giao diện lập trình ứng dụng (API) mạng được sử dụng để truyền và nhận dữ liệu giữa hai chương trình chạy trên các máy tính khác nhau. Socket được ví như một đầu nối ảo cho phép hai chương trình giao tiếp với nhau thông qua mạng.

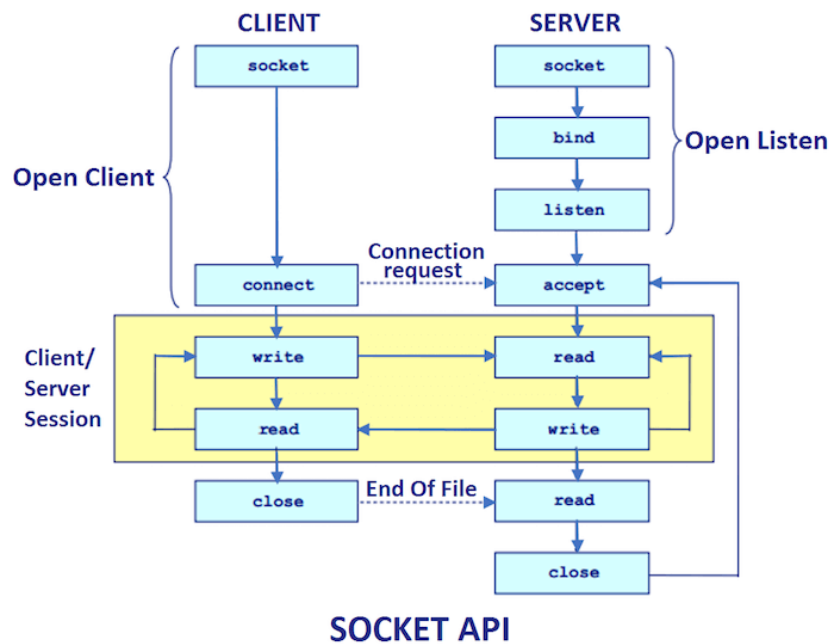
Mô-đun socket trong Python: Python cung cấp mô-đun socket để hỗ trợ lập trình socket. Mô-đun này cung cấp các hàm và lớp để tạo socket, kết nối với máy chủ, gửi và nhận dữ liệu,...

Có hai loại socket chính:

- Socket Stream (TCP): Loại socket này cung cấp kết nối đáng tin cậy giữa hai chương trình. Dữ liệu được gửi theo thứ tự và không bị mất.
- Socket Datagram (UDP): Loại socket này cung cấp kết nối không đáng tin cậy giữa hai chương trình. Dữ liệu có thể bị mất hoặc đến không đúng thứ tự.

### Các bước sử dụng socket trong Python:

```
1 import socket %Nhập mô-đun socket
2
3 %Tạo socket
4 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5
6 %Ket noi voi may chu
7 sock.connect(('localhost', 8000))
8
9 %Gui du lieu
10 sock.sendall("Hello, world!")
11
12 %Nhan du lieu
13 data = sock.recv(1024)
14 print(data.decode('utf-8'))
15
```



Hình 5: Enter Caption

```
16 %Dong socket
17 sock.close()
```

### Ứng dụng của socket:

Socket được sử dụng trong nhiều ứng dụng mạng khác nhau như:

- Web: Các trình duyệt web sử dụng socket để kết nối với các máy chủ web và tải xuống trang web.
- Email: Các máy khách email sử dụng socket để gửi và nhận email.
- Chat: Các ứng dụng chat sử dụng socket để cho phép người dùng trò chuyện với nhau.
- Chơi game: Các trò chơi trực tuyến sử dụng socket để cho phép người chơi chơi game với nhau.

### server.py

```
1 import socket
2 from _thread import *
3 import pickle
4 from game import Game
5
6 server = "192.168.1.4"
7 port = 5555
8
```



```
9  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10
11  try:
12      s.bind((server, port))
13  except socket.error as e:
14      str(e)
15
16  s.listen(2)
17  print("Waiting for a connection, Server Started")
18
19  connected = set()
20  games = {}
21  idCount = 0
22
23
24  def threaded_client(conn, p, gameId):
25      global idCount
26      conn.send(str.encode(str(p)))
27
28      reply = ""
29      while True:
30          try:
31              data = conn.recv(4096).decode()
32
33              if gameId in games:
34                  game = games[gameId]
35
36                  if not data:
37                      break
38                  else:
39                      if data == "reset":
40                          game.resetWent()
41                      elif data != "get":
42                          game.play(p, data)
43
44                      conn.sendall(pickle.dumps(game))
45                  else:
46                      break
47          except:
48              break
49
50  print("Lost connection")
51  try:
52      del games[gameId]
53      print("Closing Game", gameId)
54  except:
55      pass
56  idCount -= 1
57  conn.close()
58
59
60
```



```
61 while True:
62     conn, addr = s.accept()
63     print("Connected to:", addr)
64
65     idCount += 1
66     p = 0
67     gameId = (idCount - 1)//2
68     if idCount % 2 == 1:
69         games[gameId] = Game(gameId)
70         print("Creating a new game...")
71     else:
72         games[gameId].ready = True
73         p = 1
74
75
76     start_new_thread(threaded_client, (conn, p, gameId))
```

### network.py

```
1 import socket
2 import pickle
3
4
5 class Network:
6     def __init__(self):
7         self.client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8         self.server = "192.168.1.4"
9         self.port = 5555
10        self.addr = (self.server, self.port)
11        self.p = self.connect()
12
13    def getP(self):
14        return self.p
15
16    def connect(self):
17        try:
18            self.client.connect(self.addr)
19            return self.client.recv(2048).decode()
20        except:
21            pass
22
23    def send(self, data):
24        try:
25            self.client.send(str.encode(data))
26            return pickle.loads(self.client.recv(2048*2))
27        except socket.error as e:
28            print(e)
```

### game.py



```
1 class Game:
2     def __init__(self, id):
3         self.p1Went = False
4         self.p2Went = False
5         self.ready = False
6         self.id = id
7         self.moves = [None, None]
8         self.wins = [0,0]
9         self.ties = 0
10
11     def get_player_move(self, p):
12         """
13         :param p: [0,1]
14         :return: Move
15         """
16         return self.moves[p]
17
18     def play(self, player, move):
19         self.moves[player] = move
20         if player == 0:
21             self.p1Went = True
22         else:
23             self.p2Went = True
24
25     def connected(self):
26         return self.ready
27
28     def bothWent(self):
29         return self.p1Went and self.p2Went
30
31     def winner(self):
32
33         p1 = self.moves[0].upper()[0]
34         p2 = self.moves[1].upper()[0]
35
36         winner = -1
37         if p1 == "R" and p2 == "S":
38             winner = 0
39         elif p1 == "S" and p2 == "R":
40             winner = 1
41         elif p1 == "P" and p2 == "R":
42             winner = 0
43         elif p1 == "R" and p2 == "P":
44             winner = 1
45         elif p1 == "S" and p2 == "P":
46             winner = 0
47         elif p1 == "P" and p2 == "S":
48             winner = 1
49
50         return winner
51
```



```
52 def resetWent(self):  
53     self.p1Went = False  
54     self.p2Went = False
```

## client.py

```
1  import pygame  
2  from network import Network  
3  import pickle  
4  pygame.font.init()  
5  
6  width = 700  
7  height = 700  
8  win = pygame.display.set_mode((width, height))  
9  pygame.display.set_caption("Client")  
10  
11  
12 class Button:  
13     def __init__(self, text, x, y, color):  
14         self.text = text  
15         self.x = x  
16         self.y = y  
17         self.color = color  
18         self.width = 150  
19         self.height = 100  
20  
21     def draw(self, win):  
22         pygame.draw.rect(win, self.color, (self.x, self.y, self.width,  
23             self.height))  
24         font = pygame.font.SysFont("comicsans", 40)  
25         text = font.render(self.text, 1, (255,255,255))  
26         win.blit(text, (self.x + round(self.width/2) -  
27             round(text.get_width()/2), self.y + round(self.height/2) -  
28             round(text.get_height()/2)))  
29  
30     def click(self, pos):  
31         x1 = pos[0]  
32         y1 = pos[1]  
33         if self.x <= x1 <= self.x + self.width and self.y <= y1 <= self.y +  
34             self.height:  
35             return True  
36         else:  
37             return False  
38  
39  
40 def redrawWindow(win, game, p):  
41     win.fill((128,128,128))  
42  
43     if not(game.connected()):  
44         font = pygame.font.SysFont("comicsans", 80)  
45         text = font.render("Waiting for Player...", 1, (255,0,0), True)
```

```
42         win.blit(text, (width/2 - text.get_width()/2, height/2 -
43                        text.get_height()/2))
44     else:
45         font = pygame.font.SysFont("comicsans", 60)
46         text = font.render("Your Move", 1, (0, 255, 255))
47         win.blit(text, (80, 200))
48
49         text = font.render("Opponents", 1, (0, 255, 255))
50         win.blit(text, (380, 200))
51
52         move1 = game.get_player_move(0)
53         move2 = game.get_player_move(1)
54         if game.bothWent():
55             text1 = font.render(move1, 1, (0, 0, 0))
56             text2 = font.render(move2, 1, (0, 0, 0))
57         else:
58             if game.p1Went and p == 0:
59                 text1 = font.render(move1, 1, (0, 0, 0))
60             elif game.p1Went:
61                 text1 = font.render("Locked In", 1, (0, 0, 0))
62             else:
63                 text1 = font.render("Waiting...", 1, (0, 0, 0))
64
65             if game.p2Went and p == 1:
66                 text2 = font.render(move2, 1, (0, 0, 0))
67             elif game.p2Went:
68                 text2 = font.render("Locked In", 1, (0, 0, 0))
69             else:
70                 text2 = font.render("Waiting...", 1, (0, 0, 0))
71
72         if p == 1:
73             win.blit(text2, (100, 350))
74             win.blit(text1, (400, 350))
75         else:
76             win.blit(text1, (100, 350))
77             win.blit(text2, (400, 350))
78
79         for btn in btns:
80             btn.draw(win)
81
82     pygame.display.update()
83
84     btns = [Button("Rock", 50, 500, (0, 0, 0)), Button("Scissors", 250, 500,
85               (255, 0, 0)), Button("Paper", 450, 500, (0, 255, 0))]
86
87     def main():
88         run = True
89         clock = pygame.time.Clock()
90         n = Network()
91         player = int(n.getP())
92         print("You are player", player)
```

```
192     while run:
193         clock.tick(60)
194         try:
195             game = n.send("get")
196         except:
197             run = False
198             print("Couldn't get game")
199             break
200
201     if game.bothWent():
202         redrawWindow(win, game, player)
203         pygame.time.delay(500)
204         try:
205             game = n.send("reset")
206         except:
207             run = False
208             print("Couldn't get game")
209             break
210
211     font = pygame.font.SysFont("comicsans", 90)
212     if (game.winner() == 1 and player == 1) or (game.winner() == 0 and
213         player == 0):
214         text = font.render("You Won!", 1, (255,0,0))
215     elif game.winner() == -1:
216         text = font.render("Tie Game!", 1, (255,0,0))
217     else:
218         text = font.render("You Lost...", 1, (255, 0, 0))
219
220     win.blit(text, (width/2 - text.get_width()/2, height/2 -
221         text.get_height()/2))
222     pygame.display.update()
223     pygame.time.delay(2000)
224
225     for event in pygame.event.get():
226         if event.type == pygame.QUIT:
227             run = False
228             pygame.quit()
229
230         if event.type == pygame.MOUSEBUTTONDOWN:
231             pos = pygame.mouse.get_pos()
232             for btn in btns:
233                 if btn.click(pos) and game.connected():
234                     if player == 0:
235                         if not game.p1Went:
236                             n.send(btn.text)
237                     else:
238                         if not game.p2Went:
239                             n.send(btn.text)
240
241     redrawWindow(win, game, player)
242
243 def menu_screen():
```





```
142     run = True
143     clock = pygame.time.Clock()
144
145     while run:
146         clock.tick(60)
147         win.fill((128, 128, 128))
148         font = pygame.font.SysFont("comicsans", 60)
149         text = font.render("Click to Play!", 1, (255,0,0))
150         win.blit(text, (100,200))
151         pygame.display.update()
152
153         for event in pygame.event.get():
154             if event.type == pygame.QUIT:
155                 pygame.quit()
156                 run = False
157             if event.type == pygame.MOUSEBUTTONDOWN:
158                 run = False
159
160     main()
161
162     while True:
163         menu_screen()
```

### 3.3 Cài đặt và thiết lập môi trường

- Python version: python 3.9.0
- OS: Ubuntu 22.04, Windows 11 22H2, macOS 14.4.1

#### Tải ứng dụng

```
1 git clone https://github.com/thieuanhsang/pythonGameRockPaperScissors.git
```

#### Download package pygame

```
1 pip install pygame
```

#### Chỉnh lại cấu hình server trong file server.py và network.py

```
1 ipconfig
```



```
IPv4 Address. . . . . : 192.168.1.4
```

Hình 6: Xem ip của host

```
1 server = "192.168.1.4"
2 port = 5555
```



## Tài liệu

- [1] Python documentation “**link: <https://docs.python.org/3/>**”, *Python3*, lần truy cập cuối: 8/05/2024.
- [2] Pygame documentation “**link: <https://www.pygame.org/docs/>**”, *PyGame*, lần truy cập cuối: 8/05/2024.
- [3] Python Online Game “**link: <https://www.youtube.com/playlist?list=PLzMxCBgfZo4-kR7Rh-7JCVDN8lm3Utumvq>**”, *Tech With Tim*, lần truy cập cuối: 8/05/2024.